Aligning Videos in Space and Time

Senthil Purushwalkam*[1], Tian Ye*[1], Saurabh Gupta[2], and Abhinav Gupta[1,3]

[1] Carnegie Mellon University
[2] University of Illinois at Urbana-Champaign
[3] Facebook AI Research

Abstract. In this paper, we focus on the task of extracting visual correspondences across videos. Given a query video clip from an action class, we aim to align it with training videos in space and time. Obtaining training data for such a fine-grained alignment task is challenging and often ambiguous. Hence, we propose a novel alignment procedure that learns such correspondence in space and time via cross video cycle-consistency. During training, given a pair of videos, we compute cycles that connect patches in a given frame in the first video by matching through frames in the second video. Cycles that connect overlapping patches together are encouraged to score higher than cycles that connect non-overlapping patches. Our experiments on the Penn Action and Pouring datasets demonstrate that the proposed method can successfully learn to correspond semantically similar patches across videos, and learns representations that are sensitive to object and action states.

Keywords: understanding via association, video alignment, visual correspondences

1 Introduction

> Ask not "what is this?", ask "what is this like".
>
> *Moshe Bar*

What does it mean to understand a video? The most popular answer right now is labeling videos with categories such as "opening bottle". However, action categories hardly tell us anything about the process – it doesn't tell us where is the bottle or when it was opened, let alone the different other states it can exist in, and what parts are involved in what transitions. Dense semantic labeling is a non-starter because exhaustive and accurate labels for objects, their states and actions are not easy to gather.

In this paper, we investigate the alternative of *understanding via association*, *i.e.* video understanding by extracting visual correspondences between training and test videos. Focusing on 'what is a given video like', rather than 'what class it belongs to', side-steps the problem of hand-defining a huge taxonomy and dense labeling. Inspired by this, in this paper, we focus on the task of creating associations or visual correspondences across training and test videos. More

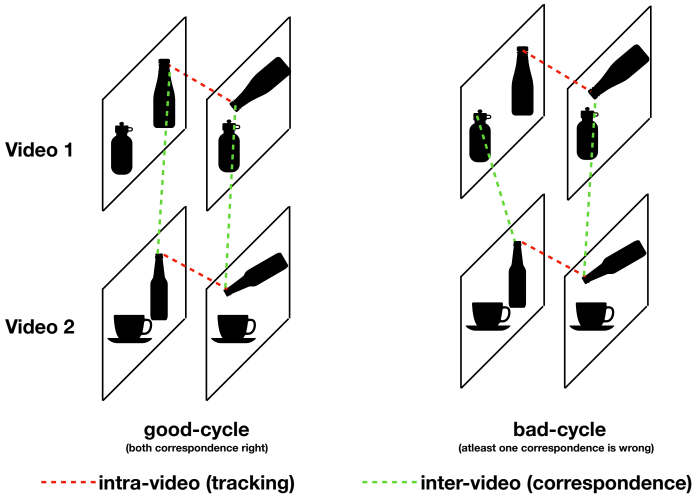
Video 1
Video 2
good-cycle
(both correspondence right)
bad-cycle
(atleast one correspondence is wrong)
intra-video (tracking)
inter-video (correspondence)

Fig. 1: Learning Correspondence via Cycle Supervision. Features that allow sequences of matches (cycles) that begin and end at the same patch are desired.

specifically, we try to align videos in both space and time. This poses two core and inter-related questions: (a) what is the granularity of visual correspondence? (b) what is the right distance metric or features to extract this correspondence?

Let us focus on the first issue: the granularity, *i.e.* the level at which we should establish correspondence: pixel-level, patch-level or frame-level. The trade-off here is between discriminability and the amount of data required for good correspondences. While full frames are more discriminative (and easy to match), they are also quite specific. For example, finding a frame that depicts the same relation between the bottle and the cup as shown in Figure 1 would require large amounts of training data before a good full-frame correspondence can be found. Consequently, past work with hand-crafted descriptors focused on establishing visual correspondence by matching interest points [30,47] and image patches [42]. However, given lack of dense supervision, recent work that tries to revisit these ideas through learning [9] seeks to correspond whole frames, through temporal consistency of frames. While this works well for full frame correspondence, it doesn't produce patch-level correspondences which is both richer, and more widely applicable. This motivates our pursuit for a method to obtain dense patch-level correspondences across videos.

The second issue at hand is of how to learn a distance metric (or equivalently an appropriate feature space) for extracting visual correspondences. Classical work focused on using manually-defined features [30,47] with a variety of distance metrics. However, given the widespread effectiveness of supervised end-to-end learning for computer vision tasks [25] (including visual correspondence [36]), it is natural to ask how to leverage learning for this task, *i.e.* what is the right objective function and supervision for learning features for obtaining correspondences? The conventional approach would be to reuse generic features from a standard task such as image classification or action recognition. As our experiments will demonstrate, neither features learned for ImageNet classification, nor ones trained for action recognition generate good correspondences due to their inability to encode object states. At the same time, direct manual annotation for visual correspondence across videos is challenging and infeasible to scale. This necessitates design of a self-supervised approach.

Interestingly, some recent efforts pursue this direction, and exploit consistency in correspondences as supervision to learn frame-level correspondence [9], or intra-video correspondence (tracking) [52]. Our proposed method extends these methods to learn patch-level correspondences across videos via *cross video cycle-consistency*. During training, given a pair of videos, we compute matches for a patch forward in time in the first video, then match to a patch in the second video, match this patch backward in time in the second video and finally match back to a patch in the first video. This sequence of patches is referred to as a 'cycle'. Cycles that start and end at overlapping patches are encouraged to score higher than cycles that connect non-overlapping patches (see Figure 1). This allows our approach to generate finer level correspondence across videos (as SIFT Flow [29] does for images), while also harnessing the capabilities of the modern end-to-end learning approaches. Our experiments show that features learned using our approach are more effective at corresponding objects in the same state across videos, than features trained for ImageNet classification, or for action classification.

2 Related Work

Our work learns space-time visual correspondence by use of cycle consistency. In this section, we present a survey of related literature on video understanding (datasets, tasks and techniques), correspondence techniques in videos, and use of self-supervision and cycle consistency for learning features and correspondences.
Video Datasets and Tasks. A number of past efforts have been devoted to collecting new video understanding datasets, and extending static image tasks to videos. Leading efforts in recent times include datasets like Kinetics [22], AvA [16], Charades [40], EPIC Kitchen [5], VLOG [11], MultiTHUMOS [56]. While some of these datasets focus on action classification, a number of them investigate new tasks, such as temporal action localization [56], detection of subjects, verbs and objects [16], classification in first-person videos [5], and analysis of crowd-sourced videos [15, 40]. These works extend video understanding by scaling it up.
Architectures for Action Classification. Researchers have also pursued design of expressive neural network architectures for the task of action classification [4, 41, 45, 46, 48, 54]. Some works investigate architectures to encourage the modelling of time flow [33, 38], or long-range temporal dependencies [10, 50, 53], or object tracking [13]. While these models often capture useful intuitions, their focus is still on optimizing models for the task of action classification. Hence, even though the model has the right inductive biases, learning is bottle-necked

by the low-entropy output space that of action class labels.

Beyond Action Recognition. Many efforts have also pursued the task of detailed video understanding in recent times. For example, video prediction tasks [7,26] have the promise to go beyond action classification, as they force the model to predict much more than what can be effectively annotated. Wang *et al.* [49] model actions as operators that transform states of objects, and Nagarajan *et al.* [34] learn about how humans interact with different objects. In contrast, we take a non-parametric approach, and understand videos by understanding what they are like, and corresponding them with other videos in space and time.

Cycle Consistency and Correspondence. Forward-backward consistency and cycle consistency have been used in computer vision for establishing correspondence in an unsupervised manner [21,39]. Zhou *et al.* [61] use cycle-consistency to establish dense correspondence between 3D shapes, Godard *et al.* [14], use cycle consistency for learning to predict depth, Zhu *et al.* [62] use cycle consistency to learn how to generate images, and Wang *et al.* [52] use cycle consistency to learn features for correspondence over time in videos. Work from Wang *et al.* [52] is a primary motivation for our work, and we investigate use of cycle consistency to learn *cross-video* correspondences. To our knowledge, ours is the first work to investigate spatio-temporal alignment across videos with cycle consistency.

Spatial Correspondence. Finding correspondences across video frames is a fundamental problem and has been actively studied for decades. Optical flow [3] seeks to establish correspondences at the pixel-level. While numerous effective approaches have been proposed [31,32,43,44], optical flow estimation is still challenging over long time periods, and fails across videos. This issue is partially alleviated by performing correspondence at a patch level. SIFT Flow [29], a seminal work in this domain, uses SIFT descriptors [30] to match patches across scene. SIFT Flow can be used to transfer labels from training data to test samples in many applications [12,28,37,57]. However, patch correspondence approaches [17,23,60], rely on the local appearance of the patches for matching. We use a similar method to obtain spatio-temporal correspondences across videos, but account for the object states and not just the local appearance.

Cross-video Spatio-Temporal Alignment. Past works have studied spatio-temporal alignment in videos. Sermanet *et al.* [38] learn time sensitive features in a supervised manner by collecting time aligned data for an action. Alayrac *et al.* [2] learn features sensitive to object states by classifying object bounding box into before or after action. Dwibedi *et al.* [9] focus on learning temporal correspondence by enforcing consistency in nearest neighbors at frame-level. This focus on frame-level modeling ignores spatial alignment. In contrast, we focus on corresponding image patches across videos in time and *space*. This leads to learning of state-sensitive *object* representations (as opposed to scene representations). We are not aware of any past work that tackles the problem of establishing spatio-temporal correspondences across videos.

Self-supervision. A number of past works employ self-supervised learning to alleviate the need for semantic supervision from humans to acquire generic image representations. Past works have employed images [8,58], videos [33,35,38,51,52],

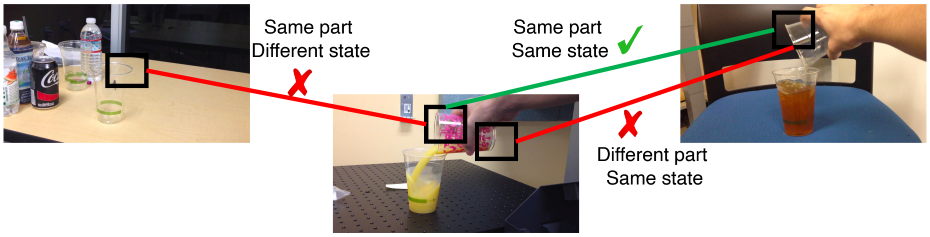

Fig. 2: What is a good correspondence? A good correspondence is a match where patches correspond to the same semantic part, and are in the same state with respect to the depicted action.

and also motor actions [1,20]. Our alignment of videos in space and time, can also be seen as a way to learn representations in a self-supervised manner. However, we learn features that are sensitive to object state, as opposed to generic image features learned by these past methods.

3 Alignment via Cross-Video Cycle Consistency

Our goal is to learn how to spatio-temporally align two videos. We tackle this problem by extracting patch level visual correspondence across two videos. But what defines a good correspondence? A good spatio-temporal correspondence is one where two patches from different videos are linked when they depict the same objects (or their parts) and are in similar states. For example, two patches depicting rim of the cups are in correspondence as shown in Figure 2 because the patches correspond to same part and the cups are in same state (tilted for pouring). On the other hand, the other two correspondences are bad because either the patches correspond to different object parts or the states of object do not match.

While it is easy to learn features that can correspond the same objects in various states over time by learning to track [51,52], it is far more challenging to learn features that correspond different objects in the same state. We specifically tackle this problem in our proposed approach. One of the biggest challenge here is the supervision. It is difficult to obtain supervision for such a dense correspondence task, thus we pursue a weakly-supervised approach. Our central idea is to employ *cross-video cycle-consistency*. Specifically, we create cycles in videos of the same action class, that track patches within a video, match it to a patch in another video, track this patch back in time, and then match back to the original video. Figure 3 illustrates the idea. Cycles that can track back to the same patch are encouraged (green cycle), while cycles that get back to a different patch in the first video are discouraged (red cycles). Enforcing this objective on a large collection of foreground patches would lead to choosing semantically aligned tracks. However, note that this could lead to some trivial cycles involving very short (or single frame) tracks in the second video. It is important to disregard

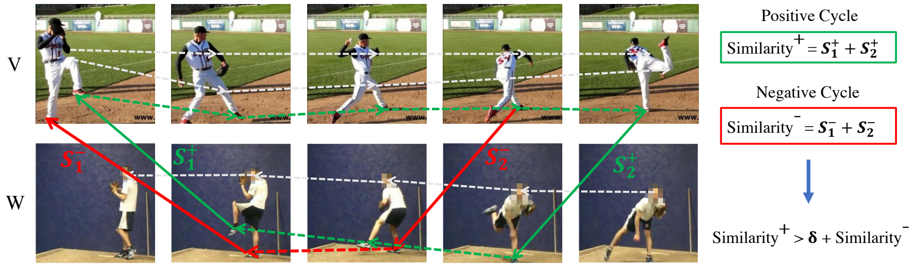
V
W
S_1^-
S_1^+
S_2^-
S_2^+
Positive Cycle
Similarity$^+$ = $S_1^+ + S_2^+$
Negative Cycle
Similarity$^-$ = $S_1^- + S_2^-$
Similarity$^+$ > δ + Similarity$^-$

Fig. 3: Overview: Given tracks in two video of the same class (shown by white dotted lines), we learn an embedding to correspond patches across videos. This is done by computing cycles (pair of cross-video edges) that correctly track a patch back to itself. We compute the best cycle that corresponds a patch to itself (shown in green) and encourage it to have a higher similarity than the best cycle that corresponds a patch to a different patch (shown in red) via a margin loss.

such solutions in order to focus on cycles where object states vary (we disregard cycles that involve tracks of length 3 or less). We now formally describe the training objective.

3.1 Formulation

Let's assume we have a tracker $\mathcal{T}$, that given a video V, produces a set of tracks on the video. We will use $V^i_{m:n}$ to denote the sequence of patches in track i starting from frame m and ending at frame n. The image patch for track i in frame m is denoted as V^i_m (see Figure 4). In this work, for obtaining tracks, we use the tracker proposed in [52] which is trained in an unsupervised manner. f_θ, realized via convolutional neural networks, denotes the desired feature embedding that establishes visual correspondence across *different* videos.

Consider the cycle shown in Figure 4: $V^i_m \to V^i_n \to W^j_q \to W^j_p \to V^k_m$. This cycle has following jumps: forward-tracking in V, matching V to W, backward-tracking in W and matching back from W to V. We represent this cycle as $\{V^i_{m:n}, W^j_{p:q}, V^k_m\}$. The score of this cycle can be expressed as the sum of patch similarities of the matches involved. However, note that the first and third matches in a cycle are extracted using off-the-shelf tracker, therefore do not depend on f_θ and can be assumed to have a constant score. Therefore, the final score of a cycle can be computed using cosine similarity s as:

$$S(\{V^i_{m:n}, W^j_{p:q}, V^k_m\}) = \underbrace{s(f_\theta(V^i_n), f_\theta(W^j_q))}_{\substack{\text{Jump from video } V \text{ (frame } n,\\ \text{patch } i\text{) to video } W \text{ (frame } q,\\ \text{patch } j\text{)}}} + \underbrace{s(f_\theta(W^j_p), f_\theta(V^k_m))}_{\substack{\text{Jump from video } W \text{ (frame } p,\\ \text{patch } j\text{) to video } V \text{ (frame } m,\\ \text{patch } k\text{)}}} \tag{1}$$

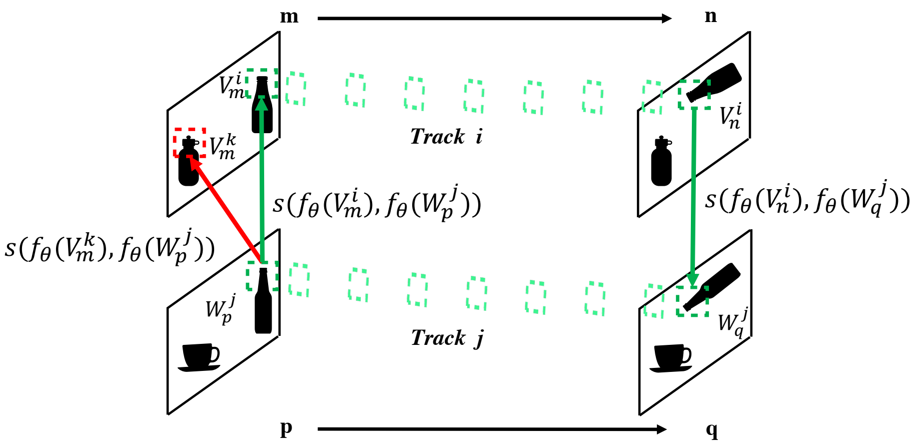
m
n
p
q
V_m^i
V_m^k
V_n^i
W_p^j
W_q^j
Track i
Track j
$s(f_\theta(V_m^i), f_\theta(W_p^j))$
$s(f_\theta(V_n^i), f_\theta(W_q^j))$
$s(f_\theta(V_m^k), f_\theta(W_p^j))$

Fig. 4: Formulation: The score of a cycle is sum of the scores of two jumps as per f_θ.

Given a starting patch V_m^i and an ending patch V_m^k, there can be numerous cycles depending on the length n considered in video V, the segment (p, q) of video W considered and the track j chosen in video W. When the patches V_m^i and V_m^k are highly overlapping, we expect the best cycle to have a high score. On the other hand, when these patches do not overlap, we want all the cycles to score low. We formulate this objective to optimize f_θ as a margin loss. First, for the pair of patches V_m^i, V_m^k, we compute the score of the best cycle as:

$$\kappa(V_m^i, V_m^k) = \max_{n,p,q,j} S(\{V_{m:n}^i, W_{p:q}^j, V_m^k\}) \tag{2}$$

The margin loss can then be formulated as:

$$\begin{aligned} &\max\left[0, -\kappa(V_m^i, V_m^{i_+}) + \kappa(V_m^i, V_m^{i_-}) + \delta\right] \\ &\forall i_+, i_- : \text{IoU}(V_m^i, V_m^{i_+}) \geq 0.5 \text{ and } \text{IoU}(V_m^i, V_m^{i_-}) < 0.5 \end{aligned} \tag{3}$$

where, δ is the fixed margin. This can be optimized using stochastic gradient descent, to learn function f_θ.

We found that using a *soft version of the max function* (Γ as defined below) instead of the *max function* in Eq. 2 was important for training. Soft version of max function, Γ is defined as follows:

$$\Gamma(\mathbf{x}) = \sum_c \mathbf{x}_c \frac{e^{\mathbf{x}_c}}{\sum_{c'} e^{\mathbf{x}_{c'}}} \tag{4}$$

Here c represents a cycle and $\mathbf{x}_c$ represents the score of that cycle. This prevents the model from getting stuck in the local minima of greedily boosting the single best cycle. The soft version of max also allows computation of gradients *w.r.t* all patches that participate in score computation, thereby updating the representations of a larger number of samples.

3.2 Using Features for Spatio-Temporal Alignment

The representation f_θ trained using our approach can be used to extract cross-video correspondences at the level of patches, tracks, frames and videos:

Patch Correspondence. f_θ can be used to correspond image patches. As f_θ learns features sensitive to state of the object, it allows us to correspond and retrieve objects that are in the same state. See Section 4 for results.

Track Correspondence. Cycles in our formulation correspond tracks with one another. Given a set of tracks in videos V and W, we correspond each track i in video V, to the track in W that maximizes the score in Eq. 1:

$$\arg\max_{j} \left(\max_{n,p,q} S\left(\{V^i_{m:n}, W^j_{p:q}, V^i_m\}\right) \right). \tag{5}$$

Temporal Alignment. We compute the similarity between a given pair of frames (V_m and W_p) in the two videos V and W by computing the total similarity between corresponding patches in the two frames:

$$T(V_m, W_p) = \sum_i \max_j s\left(f_\theta(V^i_m), f_\theta(W^j_p)\right). \tag{6}$$

These frame-level similarities can be used to obtain sub-video alignments. For example, if one wants to align K frames in video 1 to K frames in video 2 we can pick temporally-consistent top-K correspondences.

Video Retrieval. f_θ provides a natural metric for retrieving videos. Given a query video V and a set of videos $\mathcal{W}$, we retrieve the most similar video to V, by maximizing the total frame-level temporal alignment score:

$$W = \arg\max_{W \in \mathcal{W}} \sum_m \max_p T(V_m, W_p). \tag{7}$$

4 Experiments

Our goal is to demonstrate that we can align videos in space and time by leveraging f_θ learned using cross-video cycle-consistency supervision. Quantitatively measuring performance of dense spatio-temporal alignment is challenging due to the lack of ground-truth data. Therefore, in order to demonstrate the effectiveness of our approach, our experiments involve factored quantitative evaluations, and qualitative visualizations. More specifically, we study performance of our model at track correspondence, and temporal alignment.

Datasets: We perform alignment experiments on the Penn Action Dataset [59] and the Pouring Dataset [38].

Baselines: We compare our learned features to three alternate popular feature learning paradigms that focus on:

- semantics (image classification, object detection),
- local patch appearance (object trackers),
- motion and therefore object transformations (action classification models).

For models that capture semantics, we compare to ImageNet-trained ResNet-18 model `layer4` features (earlier layers do not improve results significantly), and a Mask-RCNN [18] object detection model trained on the MS-COCO [27] dataset. These models capture rich object-level semantics. For models that capture local patch appearance, we compare to features obtained via learning to track from Wang *et al.* [52]. Lastly, for models that focus on motion, we compare to features obtained via training for action classification on Kinetics [22] (ResNet-3D-18), and for frame-level action classification on Penn Action Dataset. Note, these together represent existing feature learning paradigms. Comparisons to these help us understand the extent to which our learned representations capture object state. Lastly, we also compare to recent paper from Dwibedi *et al.* [9] which only performs temporal alignment. To demonstrate the need for also modeling spatial alignment, we a consider a spatial downstream task of detecting the contact point between the thumb and a cup in the Pouring Dataset (since models from [9] are only available for the Pouring Dataset).

4.1 Experimental Settings

Tracks: We use an off-the-shelf tracker [52] to obtain tracks on videos for training and testing. Since we wish to focus on the foreground of videos for alignment, the pre-processing requires extracting tracks of foreground patches. To show robustness to patch extraction mechanism, we experiment with the following patch generation schemes (use of more sophisticated schemes is future work). For the Penn Action dataset, we track patches sampled on human detections from a Mask-RCNN detector [18]. For the Pouring dataset, we perform foreground estimation by clustering optical flow. As an ablation, we also experiment with ground-truth tracks of human keypoints in Penn Action dataset.

Training Details. We use a ResNet-18 [19] pre-trained on the ImageNet dataset [6] as our backbone model, and extract features from the last convolutional layer using RoI pooling. These features are further processed using 2 fully connected layers (and ReLU non-linearities) to obtain a 256-dimensional embedding for the input patch. We optimize the model using the Adam optimizer [24], with a learning rate of 0.0001, and a weight decay of 0.00001. We train the model for 30000 iterations on the Penn Action dataset and 500 iterations on the Pouring Dataset with each batch consisting of 8 pairs of videos. For computational efficiency, we divide each video into 8 temporal chunks. During training, we randomly sample one frame from each chunk to construct a sequence of 8 frames.

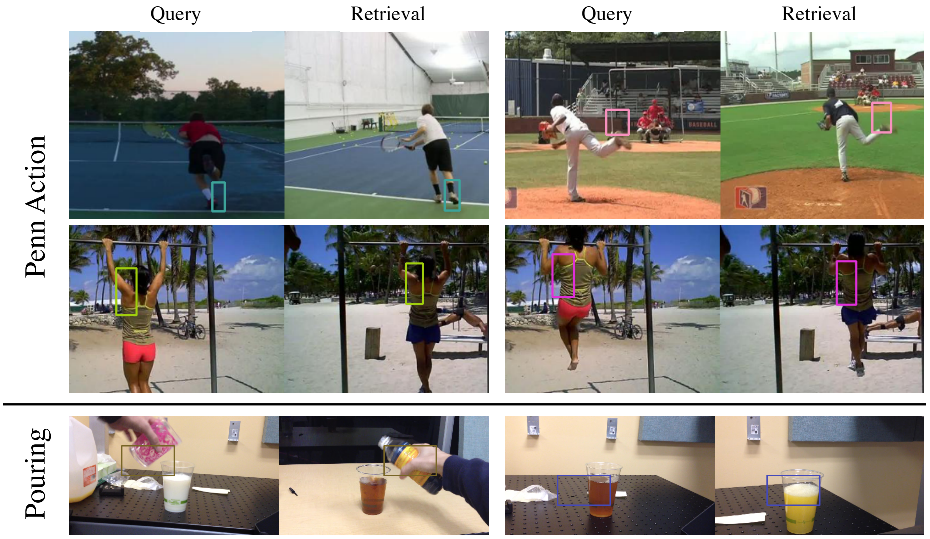

Fig. 5: Nearest neighbor patch correspondence. For random patches in query videos (left), we show the nearest neighbor patch across all frames (right) in a video retrieved using our method. We observe that our learned feature space is sensitive to the state of the object. Example in row 2 further highlights this point where our features match similar appearing patches differently based on the state of the person in the query. Row 3 shows an example from the Pouring dataset.

4.2 Qualitative Results

First we show some qualitative results of correspondences that can be extracted by our approach. Figure 5 shows some examples. We show the query frame on the left, and the corresponding nearest neighbor patch across all frames on the right. We observe that our model matches based on both the appearance and the state of the object. Next, we show that our approach can temporally align videos. Figure 6 visualizes temporal alignment on the pouring task.

Finally, we qualitatively compare the correspondence using our features compared to ImageNet and action classification features. Figure 7 shows the spatio-temporal alignment on Penn-Action dataset. Given a query video, we retrieve the most similar video based on spatio-temporal alignment. We use human keypoints to form tracks. The spatial alignment is shown by shape and color of keypoints, and the temporal alignment is shown in vertical (frames on top and bottom are temporally aligned). As compared to baseline methods, our approach is able to retrieve a more similar video, better align the frames in time, and more accurately correspond tracks with one other.

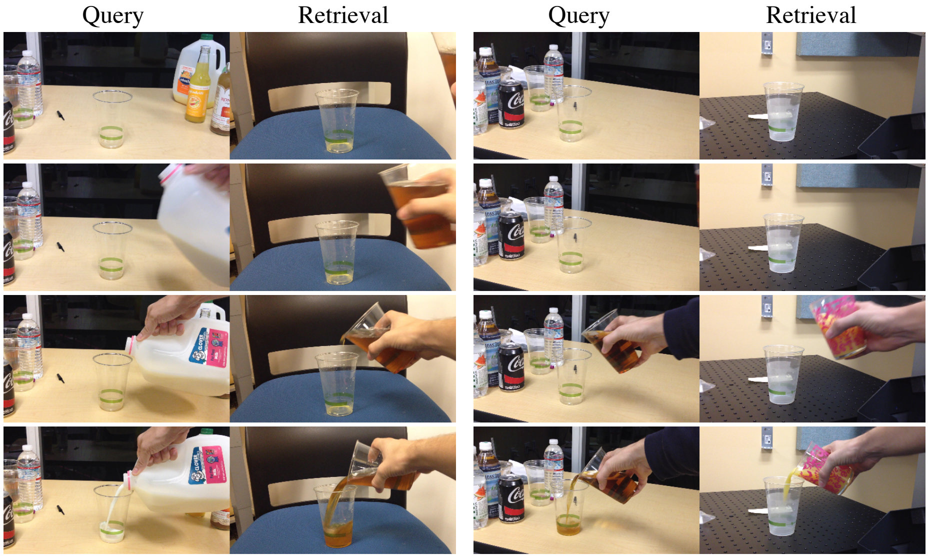

Fig. 6: Qualitative Results on Pouring Dataset: We show qualitative examples of retrieval and temporal alignment (query on left, retrieval on right) from the Pouring Dataset, based on the similarity metric learned by our model.

4.3 Quantitative Evaluation

Evaluating Temporal Alignment. Given a query video, we first obtain the closest video and then do temporal alignment as described in Section 3.2. For a given pair of frames V_m and W_p, we densely sample foreground patches and compute an average similarity using f_θ as the feature extractor. We can then temporally align the frames of videos V and W using the similarity measure in Eq. 6. Starting with 8 frames each, we align 4 frames from the query video to 4 frames in the retrieved video.

We evaluate the quality of the temporal alignment, by comparing the pose configuration of the human in the aligned frames (*i.e.* is the human in the same state in query and retrieved video). More specifically, we use the ground truth keypoint annotations to estimate and compare the angle between the surrounding limbs at left and right knee, left and right elbow, left and right hip and the neck. We report the average absolute angle difference over all joints (lower is better) in Table 1. We observe that features learned using our proposed cross-video cycle consistency leads to better temporal alignment than features from ImageNet classification, Mask-RCNN [18], frame and video classification, and intra-video correspondence [52].

Evaluating Spatial Alignment with Patches. Our proposed model can also perform spatial alignment. Given temporally aligned video frames, we use the similarity function s with the learned features f_θ to correspond image patches in temporally aligned video frames. We measure the quality of alignment by

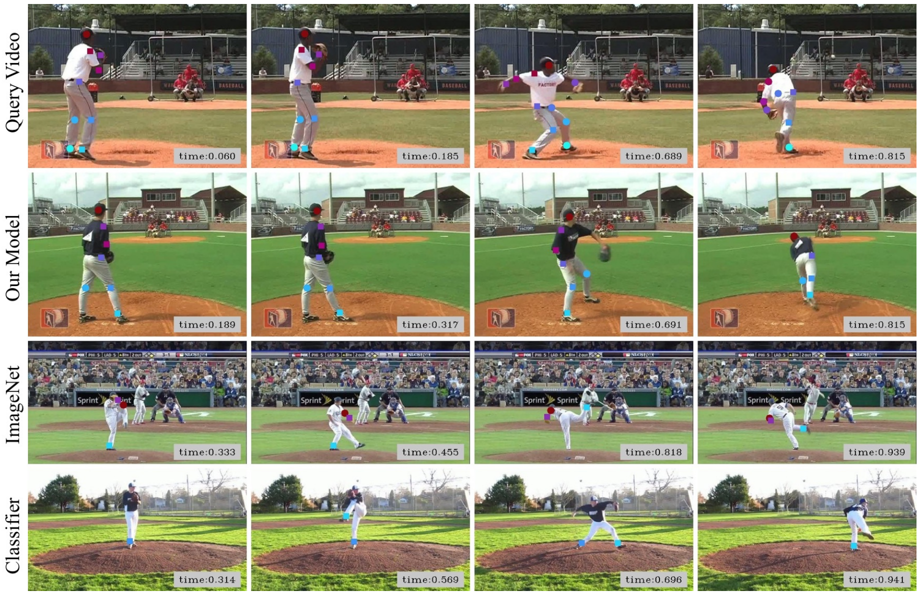

Fig. 7: We show qualitative examples of retrieval and spatio-temporal alignment on the Penn Action Dataset to compare different feature spaces. The top row shows snapshots from the query video, the second row shows video retrieved from our model (trained on tracks from [52]), the third row shows retrievals using ImageNet features, and the fourth row shows retrievals using features obtained by finetuning on the dataset using the class labels. Each columns shows temporally aligned frames, while coloured markers show spatial alignment. For all methods, we use keypoint tracks at inference time in order to showcase spatial alignment.

counting how many of the corresponding keypoints lie in aligned patches. We report the average accuracy using various feature extractors in Table 2.

Evaluating Keypoint Tracks Correspondence. Given a track in query video V, a spatially aligned track in reference video W can be identified, by using the same similarity function s with the learned features f_θ. We evaluate this by aligning keypoint tracks provided in the Penn Action dataset. Given a track of a keypoint in video V, we measure the accuracy which the aligned track corresponds to the same keypoint in video W. We report this accuracy in Table 3. Note that this alignment uses keypoint tracks only for performing inference and quantitative evaluations. Model was trained using tracks from Wang *et al.* [52] on foreground patches as before.

4.4 Ablations

Additionally, we also compare to 3 variants of our model, to understand the effectiveness of the different parts of our model. We discuss spatial alignment results (as measured by accuracy at keypoint track correspondence).

Table 1: Temporal Alignment on Penn Action Dataset [59]: We measure temporal alignment by measuring alignment in keypoint configuration at point of temporal alignment.

Method	Temporal Alignment Error ↓
ImageNet features	0.509
Features from Mask-RCNN [18]	0.504
Features from cycle-consistency based tracker [52]	0.501
Features from Kinetics [22] action classification model	0.492
Features from action classification	0.521
Our features (using tracks from [52] to train)	**0.448**

Table 2: Spatial Alignment on Penn Action Dataset [59]: We measure spatial alignment by measuring how accurately we can match keypoint by corresponding random patches between query and reference videos.

Method	Spatial Alignment Accuracy ↑
ImageNet features	0.153
Features from Mask-RCNN [18]	0.202
Features from cycle-consistency based tracker [52]	0.060
Features from Kinetics [22] action classification model	0.150
Features from action classification	0.157
Our features (using tracks from [52] to train)	**0.284**

Impact of quality of tracks used during training. We experiment with using tracks derived from ground truth key-point labels during training. We find that this leads to better features, and achieves a keypoint track correspondence accuracy of 0.650 *vs.* 0.551 when using tracks from Wang *et al.* [52]. The next ablations also uses ground-truth tracks for training.

Not searching for temporal alignment during training. Our formulation searches over temporal alignment at training time. This is done by searching for frames to jump between the two videos (max over n, p and q in Eq. 2). In this ablation, we learn features without searching for this temporal alignment, *i.e.* simply assume that the frames are aligned. The resulting features are worse at spatial alignment (keypoint track correspondence accuracy of 0.584 *vs.* 0.650).

Importance of reference video retrieval. As a first step for spatio-temporal alignment, we retrieve the best video to align. In order to ablate the performance of this retrieval task, we measure the average keypoint track correspondence accuracy by aligning all the queries to all reference videos. We observe that the accuracy drops by 15% indicating that the retrieval step is effective at choosing relevant videos.

4.5 Comparison on Pouring Dataset

We now show the necessity of learning spatial alignment by considering a spatial downstream task of predicting contact locations. We annotate the Pouring

Table 3: Track Correspondence on Penn Action Dataset [59]: We measure spatial alignment by measuring how accurately we can match keypoint tracks across videos. We compare our learned cross-video features with those obtained by pre-training on ImageNet and for action classification on the Penn Action dataset.

Method	**Track Correspondence Accuracy ↑**
ImageNet features	0.252
Features from action classification	0.110
Our features (using tracks from [52] to train)	**0.551**

Dataset [38] with locations of the contact point between the human thumb and the cup. We train a linear 1 × 1 convolution layer on the spatial features in various models to predict the probability of the contact point. We compare features from our model that are sensitive to locations of objects, *vs.* features from Dwibedi *et al.* [9] that only focus on learning good temporal alignment. We split the data into 210 training and 116 test images. We train a linear classifier on top of different features. Table shows the Percentage of Correct Keypoint (PCK) [55] metric for the localization of this contact point within a 16px × 16px neighborhood of the ground truth. We see that our features perform better than both ImageNet features, and features from [9]. Thus, features that are sensitive to object locations are essential for obtaining a rich understanding of videos.

5 Discussion

Method	**Accuracy ↑**
ImageNet features	27.1%
TCC [9]	32.7%
Ours	**38.6%**

In this work, we address the problem of video understanding in the paradigm of "understanding via associations". More specifically, we address the problem of finding dense spatial and temporal correspondences between two videos. We propose a weakly supervised cycle-consistency loss based approach to learn meaningful representations that can be used to obtain patch, track and frame level correspondences. In our experimental evaluation, we show that the features learned are more effective at encoding the states of the patches or objects involved in the videos compared to existing work. We demonstrate the efficacy of the spatio-temporal alignment through exhaustive qualitative and quantitative experiments conducted on multiple datasets.

References

1. Agrawal, P., Carreira, J., Malik, J.: Learning to see by moving. In: ICCV (2015) 5
2. Alayrac, J.B., Sivic, J., Laptev, I., Lacoste-Julien, S.: Joint discovery of object states and manipulation actions. In: ICCV (2017) 4
3. BK, P.H., Schunck, B.G.: Determining optical flow. Artificial intelligence **17**(1–3) (1981) 4
4. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR (2017) 3
5. Damen, D., Doughty, H., Maria Farinella, G., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al.: Scaling egocentric vision: The epic-kitchens dataset. In: ECCV (2018) 3
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR (2009) 9
7. Denton, E., Fergus, R.: Stochastic video generation with a learned prior. In: ICML (2018) 4
8. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: ICCV (2015) 4
9. Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A.: Temporal cycle-consistency learning. In: CVPR (2019) 2, 3, 4, 9, 14
10. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: ICCV (2019) 3
11. Fouhey, D.F., Kuo, W., Efros, A.A., Malik, J.: From lifestyle vlogs to everyday interactions. In: CVPR (2018) 3
12. Garro, V., Fusiello, A., Savarese, S.: Label transfer exploiting three-dimensional structure for semantic segmentation. In: Proceedings of the 6th International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications (2013) 4
13. Girdhar, R., Carreira, J., Doersch, C., Zisserman, A.: Video action transformer network. In: CVPR (2019) 3
14. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: CVPR (2017) 4
15. Goyal, R., Kahou, S.E., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al.: The "something something" video database for learning and evaluating visual common sense. In: ICCV. vol. 1 (2017) 3
16. Gu, C., Sun, C., Ross, D.A., Vondrick, C., Pantofaru, C., Li, Y., Vijayanarasimhan, S., Toderici, G., Ricco, S., Sukthankar, R., et al.: Ava: A video dataset of spatio-temporally localized atomic visual actions. In: CVPR (2018) 3
17. Ham, B., Cho, M., Schmid, C., Ponce, J.: Proposal flow. In: CVPR (2016) 4
18. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: ICCV (2017) 9, 11, 13
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 9
20. Jayaraman, D., Grauman, K.: Learning image representations tied to ego-motion. In: ICCV (2015) 5
21. Kalal, Z., Mikolajczyk, K., Matas, J.: Forward-backward error: Automatic detection of tracking failures. In: ICPR (2010) 4
22. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017) 3, 9, 13

23. Kim, J., Liu, C., Sha, F., Grauman, K.: Deformable spatial pyramid matching for fast dense correspondences. In: CVPR (2013) 4
24. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 9
25. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012) 2
26. Lee, A.X., Zhang, R., Ebert, F., Abbeel, P., Finn, C., Levine, S.: Stochastic adversarial video prediction (2018) 4
27. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: ECCV (2014) 9
28. Liu, C., Yuen, J., Torralba, A.: Nonparametric scene parsing: Label transfer via dense scene alignment. In: CVPR (2009) 4
29. Liu, C., Yuen, J., Torralba, A.: Sift flow: Dense correspondence across scenes and its applications. TPAMI **33**(5) (2010) 3, 4
30. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60**(2) (2004) 2, 4
31. Lucas, B.D., Kanade, T., et al.: An iterative image registration technique with an application to stereo vision (1981) 4
32. Mémin, E., Pérez, P.: Dense estimation and object-based segmentation of the optical flow with robust techniques. IEEE Transactions on Image Processing **7**(5) (1998) 4
33. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and learn: unsupervised learning using temporal order verification. In: ECCV. Springer (2016) 3, 4
34. Nagarajan, T., Feichtenhofer, C., Grauman, K.: Grounded human-object interaction hotspots from video. In: ICCV (2019) 4
35. Pathak, D., Girshick, R., Dollár, P., Darrell, T., Hariharan, B.: Learning features by watching objects move. In: CVPR (2017) 4
36. Rocco, I., Arandjelović, R., Sivic, J.: End-to-end weakly-supervised semantic alignment. In: CVPR (2018) 2
37. Rubinstein, M., Joulin, A., Kopf, J., Liu, C.: Unsupervised joint object discovery and segmentation in internet images. In: CVPR (2013) 4
38. Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., Levine, S., Brain, G.: Time-contrastive networks: Self-supervised learning from video. In: ICRA. Pouring dataset licensed under (CC BY 4.0). (2018) 3, 4, 8, 14
39. Sethi, I.K., Jain, R.: Finding trajectories of feature points in a monocular image sequence. TPAMI (1) (1987) 4
40. Sigurdsson, G.A., Varol, G., Wang, X., Farhadi, A., Laptev, I., Gupta, A.: Hollywood in homes: Crowdsourcing data collection for activity understanding. In: ECCV (2016) 3
41. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NIPS (2014) 3
42. Singh, S., Gupta, A., Efros, A.A.: Unsupervised discovery of mid-level discriminative patches. In: ECCV. Springer (2012) 2
43. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: CVPR (2010) 4
44. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: CVPR (2018) 4
45. Tran, D., Bourdev, L.D., Fergus, R., Torresani, L., Paluri, M.: C3D: generic features for video analysis. CoRR, abs/1412.0767 **2**(7) (2014) 3
46. Varol, G., Laptev, I., Schmid, C.: Long-term temporal convolutions for action recognition. TPAMI **40**(6) (2017) 3

47. Wang, H., Ullah, M.M., Klaser, A., Laptev, I., Schmid, C.: Evaluation of local spatio-temporal features for action recognition. In: BMVC (2009) 2
48. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: ECCV. Springer (2016) 3
49. Wang, X., Farhadi, A., Gupta, A.: Actions ~ transformations. In: CVPR (2016) 4
50. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: CVPR (2018) 3
51. Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. In: ICCV (2015) 4, 5
52. Wang, X., Jabri, A., Efros, A.A.: Learning correspondence from the cycle-consistency of time. In: CVPR (2019) 3, 4, 5, 6, 9, 11, 12, 13, 14
53. Wu, C.Y., Feichtenhofer, C., Fan, H., He, K., Krahenbuhl, P., Girshick, R.: Long-term feature banks for detailed video understanding. In: CVPR (2019) 3
54. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: ECCV (2018) 3
55. Yang, Y., Ramanan, D.: Articulated human detection with flexible mixtures of parts. TPAMI (2012) 14
56. Yeung, S., Russakovsky, O., Jin, N., Andriluka, M., Mori, G., Fei-Fei, L.: Every moment counts: Dense detailed labeling of actions in complex videos. IJCV **126**(2-4) (2018) 3
57. Zhang, H., Xiao, J., Quan, L.: Supervised label transfer for semantic segmentation of street scenes. In: ECCV. Springer (2010) 4
58. Zhang, R., Isola, P., Efros, A.A.: Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In: CVPR (2017) 4
59. Zhang, W., Zhu, M., Derpanis, K.G.: From actemes to action: A strongly-supervised representation for detailed action understanding. In: ICCV (2013) 8, 13, 14
60. Zhou, T., Jae Lee, Y., Yu, S.X., Efros, A.A.: Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In: CVPR (2015) 4
61. Zhou, T., Krahenbuhl, P., Aubry, M., Huang, Q., Efros, A.A.: Learning dense correspondence via 3d-guided cycle consistency. In: CVPR (2016) 4
62. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV (2017) 4