Rethinking Self-supervised Correspondence Learning: A Video Frame-level Similarity Perspective

Jiarui Xu Xiaolong Wang
UC San Diego

Abstract

Learning a good representation for space-time correspondence is the key for various computer vision tasks, including tracking object bounding boxes and performing video object pixel segmentation. To learn generalizable representation for correspondence in large-scale, a variety of self-supervised pretext tasks are proposed to explicitly perform object-level or patch-level similarity learning. Instead of following the previous literature, we propose to learn correspondence using Video Frame-level Similarity (VFS) learning, i.e, simply learning from comparing video frames. Our work is inspired by the recent success in image-level contrastive learning and similarity learning for visual recognition. Our hypothesis is that if the representation is good for recognition, it requires the convolutional features to find correspondence between similar objects or parts. Our experiments show surprising results that VFS surpasses state-of-the-art self-supervised approaches for both OTB visual object tracking and DAVIS video object segmentation. We perform detailed analysis on what matters in VFS and reveals new properties on image and frame level similarity learning. Project page is available at https://jerryxu.net/VFS.

1. Introduction

Learning visual correspondence across space and time is one of the most fundamental problems in computer vision. It is widely applied in 3D reconstruction, scene understanding, and modeling object dynamics. The research of learning correspondence in videos can be cast into two categories: the first one is learning object-level correspondence for visual object tracking [59, 57, 66], relocalizing the object with bounding boxes along the video; the other one is learning fine-grained correspondence, which is commonly applied in optical flow estimation [30, 20] and video object segmentation [9, 77]. While both lines of research have been extensively explored, most approaches acquire training supervision from simulations or limited human annotations, which increases the difficulty for generalization across different data and tasks.

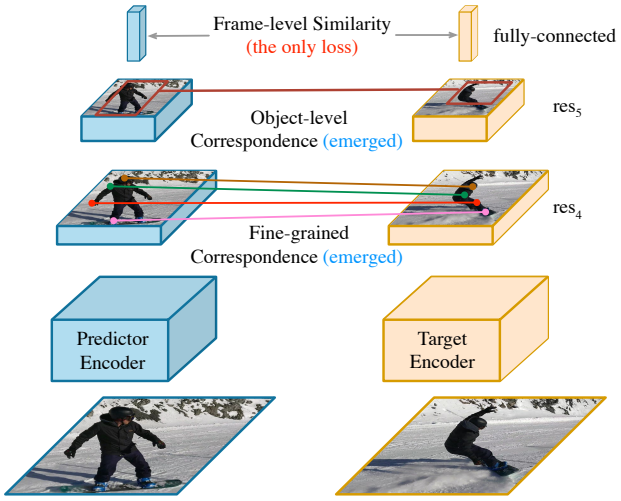

Figure 1. Video Frame-level Similarity (VFS) learning. It compares the fully-connected layer embeddings of frames from the same video for learning. By minimizing the frame-level feature distance, the fine-grained and object-level correspondence can automatically emerge in res_4 and res_5 blocks in the ResNet architecture, without using any explicit tracking-based pretext task.

One way to tackle this problem is to learn representations for correspondence using free temporal supervision signals in videos. Recently, a lot of efforts have been made in self-supervised learning of space-time visual correspondence [67, 71, 33], where different pretext tasks are designed to learn to track pixels or objects in videos. For example, Wang et al. [71] propose to use the cycle-consistency of time (i.e., forward-backward object tracking) as a supervisory signal for learning. Building on this, Jabri et al. [33] combine the cycle-consistency of time with patch-level similarity learning and achieve a significant improvement in learning correspondence. Given this encouraging result, we take a step back and ask the questions: Do we really need to design self-supervised object (or patch) tracking task explicitly to learn correspondence? Can image-level similarity learning alone learn the correspondence?

Recent development in image-level similarity learning

(e.g., contrastive learning) has shown the self-supervised representation can be applied to different downstream semantic recognition tasks, and even surpassing the ImageNet pre-training networks [26, 12, 13, 10, 50, 74, 28, 63, 64]. Our hypothesis is that if the higher-level fully-connected layer feature encodes the object structure and semantic information, it needs to be supported by the ability of finding correspondence between similar object instances and between object parts [80]. This forces the convolutional representation to implicitly learn visual correspondence.

With this hypothesis, we propose to perform Video Frame-level Similarity (VFS) learning for space-time correspondence without any explicit tracking-based pretext task. As illustrated in Figure 1, we forward one pair or multiple pairs of frames from the same video into a siamese network, and compute the similarity between the frame-level features (fully-connected layer embeddings) for learning the network representation. We examine the learning with negative pairs as [12, 26] and without negative pairs as [25, 14] under our VFS framework. We build our model based on the ResNet architecture [27]. During inference, we use the res_4 features for fine-grained correspondence task (e.g., DAVIS object segmentation [56]) and the res_5 features for object-level correspondence task (e.g., OTB object tracking [73]). Surprisingly, we find VFS can surpass state-of-the-art self-supervised correspondence learning approaches [33, 42]. Based on our experiments, we observe the following key elements for VFS:

(i) Training with large frame gaps and multiple frame pairs improves correspondence. When sampling a pair of frames from the same video for similarity learning, we observe that increasing the time differences between the two frames can improve fine-grained correspondence noticeably on DAVIS ($\sim 3\%$), and object-level correspondence significantly on OTB ($> 10\%$). Training with multiple frame pairs at the same time achieves further improvement.

(ii) Training with color augmentation is harmful for fine-grained correspondence, but beneficial for object-level correspondence. With color augmentation, the performance on the DAVIS dataset is decreased ($\sim 3\%$) while it significantly improved performance on the OTB dataset ($\sim 10\%$), which indicates the feature learns better object invariance.

(iii) Training without negative pairs improves both fine-grained and object-level correspondences. Recent literature has shown that similarity learning for visual representation is achievable even without negative pairs [25, 14]. While the results are surprising, it was still unclear how it can be beneficial for performance. In this paper, we show that VFS without negative training pairs can improve representations for different levels of correspondence.

(iv) Training with deeper networks gives significant improvements. While deeper networks generally improves recognition performance, it is not the case when training with self-supervised tracking pretext tasks for correspondence. We observe very small improvement or even worse correspondence results when using ResNet-50 compared to ResNet-18 with previous approaches [71, 33, 42]. When learning correspondence with VFS implicitly, we achieve much better performance when using a deeper model.

Given our detailed analysis and state-of-the-art performance, we hope VFS can serve as a strong baseline for self-supervised correspondence learning. The study of intermediate representations for correspondence also provides a better understanding on what image-level self-supervised similarity learning has learned. Finally, VFS also reveals the new property of similarity learning without negative pairs: it improves both object-level and fine-grained correspondence.

2. Related Work

Temporal Correspondence. Learning correspondence from video frames is a long stand problem in computer vision. We can classify the temporal correspondence into two categories. The first one is the *fine-grained* correspondence which has been widely studied in optical flow and motion estimation [47, 30, 49, 7, 6, 61, 45]. For example, Brox and Malik [6] proposed a region hierarchy matching approach to perform dense and long-range flow estimation. Recently, deep learning based approaches have been applied to estimate optical flows by training on synthetic datasets [8, 18, 58, 32, 62]. While largely improving the efficiency, training on synthetic data largely restricts the network's generalization ability to real world scenes. The second one is finding *object-level* correspondence which is meant to offer reliable and long-range visual object tracking [59, 57, 76, 72, 53, 36]. While tracking by training a detector to perform per-frame recognition offers promising results [57, 2, 37, 69, 4, 41], there is recent rise back to the classic tracking-by-matching methods [15, 76, 29, 44] using deep features [5, 66]. For example, Bertinetto et al. [5] propose a fully-convolutional siamese network and adopt similarity learning for tracking. However, these approaches are still heavily relying on human annotations for training. In this paper, we propose a self-supervised similarity learning approach which learns both fine-grained and object-level correspondence.

Self-supervised Learning from Videos. Self-supervised learning offers a way to learn generalizable visual representations with different pretext tasks [19, 17, 55, 51, 81, 21]. Beyond static images, temporal information from videos also offers rich supervision for representation learning [23, 1, 34, 48, 60, 70, 43, 54, 22]. For example, Wang et al. [70] use off-the-shelf tracker to track objects in videos to provide supervisory signals. The learned representation has shown to be useful for multiple downstream recognition

and geometry estimation tasks. Instead of learning a general representation, there is a line of recent research on self-supervised learning specifically for finding correspondence [67, 71, 68, 40, 39, 42, 33]. For example, Vondrick et al. [67] propose to propagate the current frame color to predict the future frame color as a pretext task to learn fine-grained correspondence between the current and future frame. Other tasks including tracking objects [71, 68] and patches [33] are also designed to explicitly find different levels of correspondences. But is explicit tracking task the only way to learn correspondence? Our work introduces an alternative perspective to learn correspondence implicitly via image-level similarity learning.

Image-level Similarity Learning. The image-level similarity learning provides a way for visual representation learning. It uses a siamese network to enforce two different views or augmentations of the same image to have similar features. The recent proposed self-supervised contrastive learning is a one version of similarity learning [74, 52, 28, 79, 3, 63, 26, 13, 50, 12, 10, 22]. The idea is to learn representations via attracting the similar (positive) image pairs and repulsing a large number of dissimilar (negative) image pairs. For example, He et al. [26] propose to use a momentum network to encode the large number of negatives for efficient contrastive learning. They have shown the learned representation achieves state-of-the-art performance when transferred to multiple downstream recognition tasks. Recently, it has been shown that the negative pairs are not necessary to learn a good visual representation [24, 14]. While it provides a better understanding and another manner for similarity learning, it is unclear if learning without negatives gives better representations. In this paper, we extend image-level similarity learning to video frames. Instead of performing recognition tasks, our aim is to learn visual correspondence. We also reveals that learning without negatives is beneficial for finding space-time correspondence.

3. Method

We propose Video Frame-level Similarity (VFS) learning for different levels of space-time correspondence. In this section, we will first introduce two common practices for image-level similarity learning for visual representations. Then we will unify these two paradigms under VFS.

3.1. Background: Image-level Similarity Learning

The image-level similarity learning [74, 26, 24, 12, 14] provides a way to learn visual representations in a self-supervised manner. It learns the representation by minimizing the distance between two different augmented views of the same image in feature space. We classify the similarity learning into two types based on whether it is using the negative examples during training.

Similarity Learning with Negatives. Contrastive learning [74, 26, 12] is similarity learning with negative sample pairs. In image-level contrastive learning, the positive pairs are different augmented views of the same image, and the negative pairs are from different images. The training objective is to push the representations of the positive (similar) pairs to be close to each other, while keep the representations of the negative (dissimilar) pairs to be far.

Formally, we denote x and x' as two different augmented views of an input image. The contrastive learning utilizes a siamese network architecture with a predictor encoder $\mathcal{P}$ and a target encoder $\mathcal{T}$. We use these two networks to extract features for both inputs respectively, and normalize the outputs with l_2-normalization. The output embeddings for x and x' can be represented as $p \triangleq \mathcal{P}(x)/\|\mathcal{P}(x)\|_2$ and $z \triangleq \mathcal{T}(x')/\|\mathcal{T}(x')\|_2$. Let $\mathcal{U} = \{u_1, u_2, \dots, u_K\}$ be the negative bank that stores the features of negative samples. The optimization objective is minimizing InfoNCE loss [52], defined as

$$\mathcal{L}_{p,z,\mathcal{U}} = -\log \frac{\exp(p{\cdot}z/\tau)}{\exp(p{\cdot}z/\tau) + \sum_{k=1}^{K} \exp(p{\cdot}u_k/\tau)} \quad (1)$$

where τ is a temperature hyper-parameter [74].

There are multiple ways to construct the negative sample bank including directly storing all the features [74] or sampling the negatives online [12]. In this paper, we adopt a recent practice proposed by He et al. [26], which uses a momentum updated queue as the negative bank. To achieve this, the target encoder network $\mathcal{T}$ is updated as a moving average of the predictor encoder $\mathcal{P}$ as,

$$\xi \leftarrow m\xi + (1-m)\theta, \; m \in [0, 1), \quad (2)$$

where θ, ξ are parameters of $\mathcal{P}$ and $\mathcal{T}$. Thus the encoders $\mathcal{P}$ and $\mathcal{T}$ share the same network architecture in [26].

Similarity Learning without Negatives. Recently, researchers discover that similarity learning without negative sample pairs can achieve comparable performance as contrastive learning in visual representation learning [24, 14]. Without the negative samples, the optimization objective can be simplified as the minimizing the cosine feature distance between two views

$$\mathcal{L}_{p,z} = \|p - z\|_2^2 = 2 - 2{\cdot}p{\cdot}z. \quad (3)$$

However, optimizing this objecitve will easily lead to degenerated solution and yields a collapsed representation [24, 14]. To resolve this issue, Chen et al. [14] propose two techniques: (i) Share most of the parameters between the predictor encoder $\mathcal{P}$ and the target encoder $\mathcal{T}$, except adding one additional MLP for the predictor encoder; (ii) Stop the gradients back-propagated from the loss to the target network. In this paper, we adopt this approach for similarity learning without negative data. Please refer to [14] for more details.

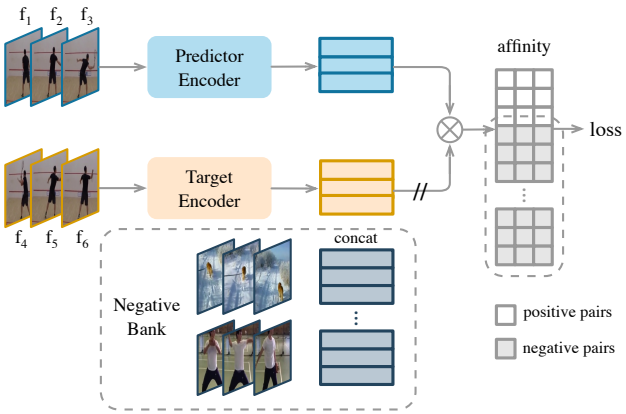

Figure 2. **Video Frame-level Similarity Pipeline**. The affinity matrix is the pairwise feature similarity between predictor features and target features. The dash rectangle areas indicate the *with negative pairs* case. The features of negative samples are stored in the negative bank and concatenate with target features. The encoder is trained to maximize the affinity of positive pairs and minimize the affinity of negative ones.

3.2. Video Frame-level Similarity Learning

Building on image-level similarity learning, we propose to perform similarity learning between video frames, i.e., Video Frame-level Similarity learning (VFS). Our approach considers frames at different timestamps as different views for similarity learning. In a video clip with length L, there are $\frac{L^2}{2}$ possible positive sample pairs for learning. In VFS, each video frame is pulled towards a global video feature, resulting in a representation invariant to natural object deformation and viewpoint changes over time. In our experiments, we show that this learning objective can enforce the emergence of visual correspondence from the convolutional layers. Our hypothesis is that, like applying other augmentations in image-level similarity learning, sampling different temporal views also help the representation to learn object structure and even semantic information. This implicitly requires the convolutional features to learn about object and fine-grained correspondence. We will introduce our approach in details as follows.

Learning Objectives. Given a video with L frames $\{f_1, f_2, \dots, f_L\}$, we sample two random frames f_i, f_j and apply data augmentation on them. We then forward both frames to the predictor encoder $\mathcal{P}$ and target encoder $\mathcal{T}$ to extract their features as $p_i \triangleq \mathcal{P}(f_i)/\|\mathcal{P}(f_i)\|_2$, $z_j \triangleq \mathcal{T}(f_j)/\|\mathcal{T}(f_j)\|_2$. We provide two options for VFS learning with and without the negative pairs. Following Eq. 1, the objective for VFS *with negative pairs* can be represented as,

$$\mathcal{L}_{p_i,z_j,\mathcal{U}} = -\log \frac{\exp(p_i \cdot z_j/\tau)}{\exp(p_i \cdot z_j/\tau) + \sum_{k=1}^{K} \exp(p_i \cdot u_k/\tau)}. \quad (4)$$

method	I_i	$\mathbb{E}(I_i - I_{i-1})$
Continuous Sampling	$I_1 + (i-1)\delta$	δ
Distant Sampling	$\frac{L}{n}(i-1) + \text{unif}(0, \frac{L}{n})$	$\frac{L}{n}$

Table 1. **Video Frame Sampling**. I_i is the i-th sampled frame index. δ is the frame interval of *continuous sampling*. n is the number of segments (frames) of *distant sampling*.

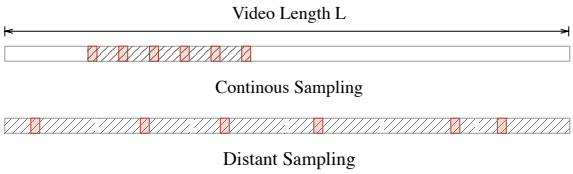

Figure 3. **Video Frame Sampling**. Dash areas are selected segments for sampling. Solid blocks indicate sampled frames. *Continuous sampling* yields temporally continuous frames, while *distant sampling* samples frames with the larger displacement as well as randomness.

On the other hand, VFS can also be trained *without negative pairs* by following Eq. 3, and the objective can be represented as,

$$\mathcal{L}_{p_i,z_j} = \|p_i - z_j\|_2^2 = 2 - 2 \cdot p_i \cdot z_j. \quad (5)$$

While we unify learning with and without negatives under VFS, we adopt the implementation details in [13, 14] to adjust the architectures and training schemes accordingly as introduced in Section 3.1.

We illustrate our learning pipeline as Figure 2. Beyond sampling one pair of data, learning from videos allows to sample multi-paired positive samples. To be more specific, we can sample n frames from the videos, and divide them into two splits for predictor and target encoders. We then compute the feature similarity between $\frac{n}{2}$ predictor features and $\frac{n}{2}$ target features which yields a affinity matrix is of shape $\frac{n}{2} \times \frac{n}{2}$. When training without negatives, we can apply Eq. 5 to maximize each element in the affinity matrix, i.e., minimize the distance between each video frame pair. When training with K negative examples, the shape of the affinity matrix becomes $\frac{n}{2} \times (\frac{n}{2} + K)$. We apply Eq. 4 for learning, where the negative bank $\mathcal{U} = \{u_1, u_2, \dots, u_K\}$ are sampled from other videos. We illustrate the sampling method for the n frames as follows.

Temporal Sampling. Given a video with L frames, sampling n frames yields a set of indices $\{I_1, I_2, \dots, I_n\}$, where $I_i \in [1, L]$. In this paper, we investigate two strategies for temporal sampling: (i) *Continuous sampling*, which first selects a starting frame index and then continuously samples with a fixed frame interval δ; (ii) *Distant sampling*, which splits the video into n disjoint segments, then randomly selects 1 frame from each segment as $I_i = \frac{L}{n}(i-1) + \text{unif}(0, \frac{L}{n})$. We illustrate the two sampling strategies in both Table 1 and Figure 3. The continuous

sampling strategy is widely applied in architectures that exploits 3D Convolution [65, 11] for locally consistent feature map [75]. On the other hand, the distant sampling provides a larger coverage of the entire video and a more aggressive augmentation effect. We will ablate how the sampling strategy affects correspondence learning in our experiments.

Data Augmentation. Besides using temporal signals to provide different views of training data, we also adopt the common data augmentation practices in image-level similarity learning [74, 13, 12]. We apply the spatial augmentation (e.g., random cropping and flipping) and color augmentation (e.g., grayscale and color jitter) in our experiments. Specifically, we observe the color augmentation plays an important role in correspondence learning and we will report our findings in the experiment section.

4. Experiments

We perform experiments on representation learned by VFS for fine-grained correspondence tasks [56, 35, 82] and object-level correspondence task [73]. We will first introduce our training details and evaluation metrics, then we will perform extensive ablations on different elements for VFS and provides a better understanding on how VFS learns correspondence. Based on these observations, we finally report VFS surpasses all previous self-supervised learning approaches on both correspondence tasks.

4.1. Self-Supervised Pre-Training

Architectures. We use standard ResNet [27] as the backbone network. We introduce the architecture for training without negative pairs (following [14]) and the architecture for training with negative pairs (following [13]) as below.

- *Architecture without negative pairs.* The predictor encoder consists of a backbone network and a projector followed by a predictor. The target encoder is composed of the backbone and the projector. The parameters of the backbone and the projector are shared between the two encoders. The projector is a 3-layer MLP and the predictor is a 2-layer MLP as [14]. All batch normalization layers in the backbone, the projector and the predictor are synchronized across devices (SyncBN) as in [12, 24, 14].
- *Architecture with negative pairs.* The predictor encoder and target encoder share the same architecture, including a backbone followed by a 2-layer projector MLP. The parameters of the target encoder are updated with momentum $m = 0.999$ with Eq. 2. There is no predictor head. Shuffle BN [13] is used instead of SyncBN. We set the temperature $\tau = 0.2$ and the negative bank size $K = 65536$.

Pre-training. We adopt the Kinetics[38] dataset for self-supervised training. It consists of ∼240k training videos. The batch size is 256. The learning rate is initialized to 0.05, and decays in the cosine schedule [12, 46, 13]. We use SGD optimizer with momentum 0.9 and weight decay 0.0001. We found that training for 100 epochs is sufficient for ResNet-18 models, and ResNet-50 models need 500 epochs to converge (roughly the same number of iterations as 100 epochs training [14] on ImageNet).

4.2. Evaluation

We evaluate the pre-trained representation on both fine-grained and object-level correspondence downstream tasks.

Fine-grained Correspondence. To evaluate the quality of fine-grained correspondence, we follow the same testing protocol and downstream tasks in [33, 71]. Without any fine-tuning, we directly use unsupervised pre-trained model as the feature extractor. The fine-grained similarity is measured on the res_4 feature map, with its stride reduced to 1 during inference. As in [71, 33, 42], the recurrent inference strategy is applied: The first frame ground truth labels as well the prediction results in the latest 20 frames are propagated to the current frame. We evaluate the fine-grained correspondence over three downstream tasks and datasets: video object segmentation in DAVIS-2017 [56], human pose tracking in JHMDB [35] and human part tracking in VIP [82]. We perform most of our ablations with DAVIS-2017 and report the comparisons with state-of-the-art approaches in all datasets.

Object-level Correspondence. We evaluate object-level correspondence with visual object tracking in the OTB-100 [73] dataset. We adopt the SiamFC [5] tracking algorithm with our representation from the res_5 block. We follow the same evaluation protocol and hyperparamters in VINCE [22] and SeCo [78]: Given a pre-trained ResNet, the strides in res_4 and res_5 layers are removed, and the dilation rate of 3x3 convolution blocks in res_4 and res_5 layers are set to 2 and 4 respectively. This commonly used modification makes the res_5 block resolution compatible with the original setting in SiamFC. The network modification does not affect the pre-trained weights. We perform most of our analysis and ablation for VFS using the frozen representation after pre-training. We compare to the state-of-the-art results using fine-tuning in the end following [22, 78].

4.3. Results and Ablative Analysis

We first perform analysis on different elements of VFS. We use ResNet-18 as the default backbone. The experiments are under the *without negative pairs* setting unless specified otherwise.

Augmentation. Out first discovery is that color augmentation plays an important role in VFS and affect fine-grained and object-level correspondence in an opposite way. We report our results on augmentations in Table 2. With color

different frame	color aug	spatial aug	DAVIS $\mathcal{J}\&\mathcal{F}_m$	DAVIS $\mathcal{J}_m$	DAVIS $\mathcal{F}_m$	OTB Precision	OTB Success
			38.6	37.3	39.9	5.4	4.7
	✓		50.9	49.3	52.6	0.4	0.3
		✓	62.2	60.8	63.6	30.6	26.1
	✓	✓	61.2	59.3	63.1	53.0	39.3
✓			63.4	61.1	65.7	37.4	28.8
✓	✓		58.4	56.4	60.4	46.2	34.6
✓		✓	65.0	62.6	67.4	48.1	37.9
✓	✓	✓	61.9	59.5	64.3	57.3	43.0

Table 2. **Ablation on different augmentations**. Color augmentations are random color jitter, grayscale conversion, gaussian blur. Spatial augmentations are random resized crop and horizontal flip. "different frame" indicates whether the inputs for predictor and target encoder are different.

frame interval	DAVIS $\mathcal{J}\&\mathcal{F}_m$	DAVIS $\mathcal{J}_m$	DAVIS $\mathcal{F}_m$	OTB Precision	OTB Success
0	62.2	60.8	63.6	30.6	26.1
2	63.5	61.7	65.4	38.1	31.2
4	62.9	61.4	64.3	35.5	29.5
8	63.5	61.7	65.3	38.5	31.8
16	63.9	61.9	65.8	44.1	33.9
32	64.5	62.4	66.7	46.9	36.1
D	65.0	62.6	67.4	48.1	37.9

Table 3. **Ablation on frame interval**. 0 means sample two identical frames. D stands for the *distant sampling*. Others use *continuous sampling* with fixed frame sampling interval.

augmentation, the performance on DAVIS decrease by $> 3\%$ and the OTB precision improves over 10%. We conjecture that while color augmentation helps learning invariance to object appearance changes, the per-pixel distortion confuses the lower-level convolution features to find fine-grained correspondence. From Table 2, we also observe sampling different frames in a video (with distant sampling) indeed help learning both object-level and fine-grained correspondence, and adding spatial augmentations (random crop and flip) can further improve the results. In the following experiments, we adopt different frame inputs and spatial augmentation by default. We will report results with and without color augmentations in the different ablations.

Temporal Sampling. Does sampling with larger frame interval improve correspondence learning? To answer this question, we study the temporal sampling strategy and report our results in Table 3. We perform the study without using the color augmentation. Recall we have two temporal sampling strategies in VFS (Section 3.2). With *continuous sampling*, we elaborate the frame interval δ from 0 to 32. We observe as δ increases, both fine-grained and object-level correspondence improves consistently. With *distant sampling* using $n = 2$ frame inputs (labeled as "D" in Table 3), we achieve the best results. We conjecture increasing the frame intervals offers better augmentation effect which leads to better correspondence. We will adopt the distant sampling in the following subsections.

color aug	frame num	DAVIS $\mathcal{J}\&\mathcal{F}_m$	DAVIS $\mathcal{J}_m$	DAVIS $\mathcal{F}_m$	OTB Precision	OTB Success
	n=2	65.0	62.6	67.4	48.1	37.9
	n=4	65.8	63.2	68.4	51.5	38.4
	n=8	66.7	64.0	69.4	53.0	39.6
✓	n=2	61.9	59.5	64.3	57.3	43.0
✓	n=4	62.9	60.5	65.3	58.4	43.8
✓	n=8	62.5	59.8	65.1	59.0	43.8

Table 4. **Ablation on multiple frames**. "frame num" is the number of sampled frame.

negative pairs	color aug	DAVIS $\mathcal{J}\&\mathcal{F}_m$	DAVIS $\mathcal{J}_m$	DAVIS $\mathcal{F}_m$	OTB Precision	OTB Success	ImageNet Acc@1
		65.0	62.6	67.4	48.1	37.9	22.0
✓		64.7	62.2	67.3	39.0	31.7	24.2
	✓	61.9	59.5	64.3	57.3	43.0	31.8
✓	✓	61.5	59.3	63.7	53.7	40.8	33.8

Table 5. **Ablation on negative pairs**. For *with negative pairs* setting, the learning objective is Eq 4.

negative pairs	color aug	Backbone	DAVIS $\mathcal{J}\&\mathcal{F}_m$	DAVIS $\mathcal{J}_m$	DAVIS $\mathcal{F}_m$	OTB Precision	OTB Success
		ResNet-18	65.0	62.6	67.4	48.1	37.9
		ResNet-50	68.9	66.5	71.3	47.4	34.6
✓		ResNet-18	64.7	62.2	67.3	39.0	31.7
✓		ResNet-50	68.3	65.8	70.8	46.4	34.4
	✓	ResNet-18	61.9	59.5	64.3	57.3	43.0
	✓	ResNet-50	67.1	64.6	69.6	59.5	43.4
✓	✓	ResNet-18	61.5	59.3	63.7	53.7	40.8
✓	✓	ResNet-50	67.2	64.7	69.7	56.5	40.7

Table 6. **Ablation on deeper models**. ResNet-18 and ResNet-50 are compared.

Multiple Frames. We investigate the effect of training with multiple pairs of frames from a video. With distant sampling, we ablate the number of segments $n = 2, 4, 8$, which indicates using $1, 2, 4$ pairs of frames from one video during training. We report our results in Table 4. With or without using color augmentation, we observe training with more pairs of frames generally improves the correspondence representation. Note the first row in Table 4 corresponds to the last row in Table 3.

Negative Pairs. So far we conduct our experiments under the *without negative pairs* setting. But will adding the negative pairs (i.e., contrastive learning) help correspondence learning? To our surprise, adding negative pairs in training hurts learning both fine-grained and object-level correspondence. We report the comparisons in Table 5. While the fine-grained correspondence result on DAVIS drops slightly with negative pairs, the performance of object tracking in OTB degenerates significantly. When training without color augmentations, the object tracking precision drops $\sim 9\%$.

What is the reason causing this performance drop when training with negatives? Our hypothesis is that training with negative pairs may sacrifice the performance on mod-

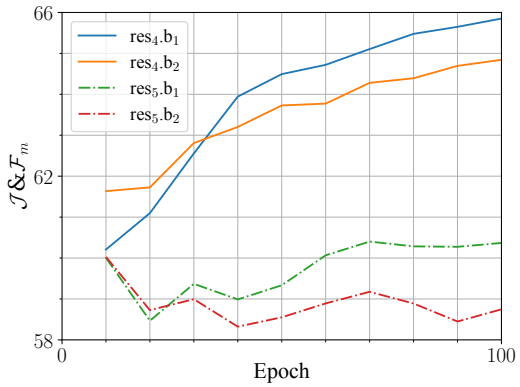

Figure 4. $\mathcal{J}\&\mathcal{F}_m$ on DAVIS at different epochs with **ResNet-18**.

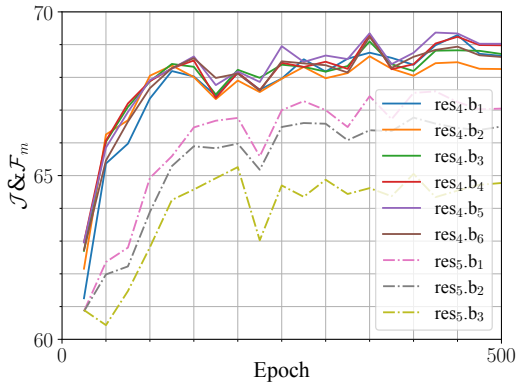

Figure 5. $\mathcal{J}\&\mathcal{F}_m$ on DAVIS at different epochs with **ResNet-50**.

eling intra-instance invariance for learning better features for cross instance discrimination. To prove this hypothesis, we perform linear classification on top of the frozen features on the ImageNet-1k dataset [16] and report the results on the right column of Table 5. We observe that the model trained with negatives indeed leads to better semantic classification with around 2% improvement, which supports our hypothesis. Note that these results are not directly comparable to state-of-the-art results on ImageNet-1k [12, 13, 14], since we train our model (ResNet-18 backbone) with video frames (with large domain gap) and optimize for correspondence learning instead of semantic classification. With these observations, we conjecture training without negatives can learn better correspondence representation, which has not been shown in previous literature to our knowledge.

Deeper Models. We so far mainly perform analysis with ResNet-18. We observe the performance does not change or even degrades in previous self-supervised learning approaches [33, 42, 39, 71] with ResNet-50. So can VFS be scaled up to deeper models? To answer this, we train VFS with ResNet-50 backbone and report the results in Table 6. We observe deeper networks can significantly improve both fine-grained and object-level correspondence with VFS. Under without negative pairs setting, ResNet-50 backbone improves DAVIS $\mathcal{J}\&\mathcal{F}_m$ by 3.9%, OTB precision by 2.2% over ResNet-18. More interestingly, while the ResNet-18 performance on DAVIS decrease by 3.1% with color augmentation, the gap has been closed to 1.8% under the ResNet-50 setting. We conjecture stacking more convolution blocks helps adapt to appearance distortion. More ResNet-50 comparison could be found in Table 7 and 8.

Different Blocks of Layers. We plot the VFS model performance on DAVIS throughout different epochs of similarity learning with ResNet-18 (Figure 4) and ResNet-50 (Figure 5). We investigate how each feature block in res_4 and res_5 layers performs. For example, res_4,b_1 indicates the feature from the first block in res_4. In Figure 4, all blocks begin with $\sim 60\%$ on DAVIS in early stages of training. However, the gap between res_4 and res_5 becomes larger as the model is trained longer. The blocks in res_5 ends up with similarly scores as beginning, while the results of res_4 continuously improve. This supports our previous results that res_4 learns fine-grained correspondence over time and res_5 focuses more on object-level features. Similarly for ResNet-50 in Figure 5, res_4 also outperforms res_5 by a noticeable margin for finding fine-grained correspondence.

4.4. Comparison with State-Of-The-Art

We compare fine-grained correspondence results of VFS against previous self-supervised methods in Table 7. The results are all reported with the last block in res_4 across all methods. Our method achieves state-of-the-art performance using ResNet-50. With ResNet-50, we observes UVC [42] does not benefit from using a deeper networks and the performance of CRW [33] becomes significantly worse. Learning with VFS, the deeper network with ResNet-50 improves 2.2% on DAVIS and 3.3% on VIP over ResNet-18, which is significant. We observe consistent results across the JHMDB [35] human pose and VIP [82] human part tracking tasks. With ResNet-18, VFS achieves comparable performance with CRW [33]. As shown in Figure 4 and 5, the last block in res_4 may not achieve the optimal performance, thus we also report the result of the best block with gray color for reference.

The comparison on learning object-level correspondence is reported in Table 8. The result of CRW [33] significantly underperforms other pre-training methods. In CRW, the network only receives patch inputs (64×64 patch cropped from 256×256 input). One possible reason for the inferior performance is that the object in the video does not fit in the small patch, thus the network may not be able to learn object-level correspondence. We report the fine-tuning performance of our best setting, namely distant sampling with $n = 8$, with color augmentation and without negative pairs. VFS with ResNet-50 backbone brings 5% precision gain over ResNet-18, yields 73.9% precision and 52.5% success

Method	Backbone	Stride	Dataset	DAVIS $\mathcal{J}\&\mathcal{F}_m$	DAVIS $\mathcal{J}_m$	DAVIS $\mathcal{F}_m$	VIP mIoU	JHMDB PCK@0.1	JHMDB PCK@0.2
Supervised [27]	ResNet-18	32	ImageNet	62.9	60.6	65.2	31.9	53.8	74.6
SimSiam [14]	ResNet-18	32	ImageNet	62.0	60.0	64.0	30.3	55.2	74.0
MoCo [26]	ResNet-18	32	ImageNet	60.8	58.6	63.1	29.2	55.0	72.7
VINCE [22]	ResNet-18	32	Kinetics	60.4	57.9	62.8	30.5	55.6	73.4
CorrFlow [40]*	ResNet-18‡	4	OxUvA	50.3	48.4	52.2	-	58.5	78.8
MAST [39]*	ResNet-18‡	4	OxUvA	63.7	61.2	66.3	-	-	-
MAST [39]*	ResNet-18‡	4	YT-VOS	65.5	63.3	67.6	-	-	-
Vid. Color. [67]	ResNet-18†	8	Kinetics	34.0	34.6	32.7	-	45.2	69.6
TimeCycle [71]	ResNet-18†	8	VLOG	39.2	40.1	38.3	28.9	57.3	78.1
UVC [42]	ResNet-18†	8	Kinetics	57.8	56.3	59.2	34.1	58.6	79.6
UVC+track [42]*	ResNet-18†	8	Kinetics	59.5	57.7	61.3	-	-	-
CRW [33]	ResNet-18†	8	Kinetics	**67.6**	**64.8**	**70.2**	38.6	59.3	**80.3**
VFS	ResNet-18	32	Kinetics	66.7	64.0	69.4	**39.9**	**60.5**	79.5
VFS (best block)	ResNet-18	32	Kinetics	67.9	65.0	70.8	-	-	-
Supervised [27]	ResNet-50	32	ImageNet	66.0	63.7	68.4	39.5	59.2	78.3
SimSiam [14]	ResNet-50	32	ImageNet	66.3	64.5	68.2	35.0	58.4	77.5
MoCo [26]	ResNet-50	32	ImageNet	65.4	63.2	67.6	36.1	60.4	79.3
VINCE [22]	ResNet-50	32	Kinetics	65.6	63.4	67.8	36.0	58.2	76.3
TimeCycle [71]	ResNet-50†	8	VLOG	40.7	41.9	39.4	28.9	57.7	78.5
UVC [42]	ResNet-50†	8	Kinetics	56.3	54.5	58.1	34.2	56.0	76.6
CRW [33]	ResNet-50†	8	Kinetics	29.7	27.6	31.9	10.4	21.3	32.2
VFS	ResNet-50	32	Kinetics	**68.9**	**66.5**	**71.3**	**43.2**	**60.9**	**80.7**
VFS (best block)	ResNet-50	32	Kinetics	69.4	66.7	72.0	-	-	-

Table 7. **Comparison with state-of-the-art on fine-grained correspondence**. $*$ indicates localization is involved during label propagation. † denotes strides of last two layers are removed. ‡ denotes max pooling of stem layer is also removed. Stride is the output stride (downsample ratio) of ResNet.

Method	Backbone	Dataset	OTB Precision	OTB Success
Supervised [27]	ResNet-18	ImageNet	61.4	43.0
SimSiam [14]	ResNet-18	ImageNet	58.8	42.9
MoCo [26]	ResNet-18	ImageNet	62.0	47.0
VINCE [22]	ResNet-18	Kinetics	62.9	46.5
CRW [33]	ResNet-18†	Kinetics	52.6	40.1
VFS	ResNet-18	Kinetics	**68.9**	**52.2**
Supervised [27]	ResNet-50	ImageNet	65.8	45.5
SimSiam [14]	ResNet-50	ImageNet	61.0	43.2
MoCo [26]	ResNet-50	ImageNet	63.7	46.5
VINCE [22]	ResNet-50	Kinetics	66.0	47.6
SeCo [78]	ResNet-50	Kinetics	71.9	51.8
CRW [33]	ResNet-50†	Kinetics	4.9	4.8
VFS	ResNet-50	Kinetics	**73.9**	**52.5**

Table 8. **Comparison with state-of-the-art on object-level correspondence**. † denotes strides of last two layers are removed.

score. Our simple VFS surpasses ImageNet supervised pretraining as well as previous self-supervised state-of-the-art SeCo [78], which involves joint training of 3 pretext tasks.

5. Discussion and Conclusion

We propose a simple yet effective approach for self-supervised correspondence learning. We demonstrate that both fine-grained and object-level correspondence can emerge in different layers of the ConvNets with video frame-level similarity learning. In addition to the state-of-the-art performance, we provide the following insights.

Is designing a tracking-based pretext task a necessity for self-supervised correspondence learning? It might not be necessary. While tracking-based pretext tasks still have potentials, it is limited by small backbone models and is now surpassed by our simple frame-level similarity learning. To make the tracking-based pretext tasks useful, we need to first make its learning scalable and generalizable in model size and network architectures.

Does color augmentation improve the correspondence? Yes and no. We show that color augmentation is beneficial for correspondence in object-level but jeopardizes the fine-grained correspondence. While color augmentation brings object appearance invariance, it also confuses the lower-layer convolution features.

How to sample video frames? Sample multiple frames and sample with a large gap. The large temporal gap provides more aggressive temporal transform, which boosts correspondence significantly. Comparing multiple pairs of frame further improves the results.

Is negative pairs helpful? No. We observe inferior performance when training with negative samples, specifically for object-level correspondence. We also shed light on the reason why *without negative pairs* is more helpful, which has not been studied before.

In conclusion, we hope VFS can serve as a strong baseline for self-supervised correspondence learning. We provide a new perspective on studying similarity learning for visual representations beyond recognition tasks.

Acknowledgements. This work was supported, in part, by grants from DARPA LwLL, NSF 1730158 CI-New: Cognitive Hardware and Software Ecosystem Community Infrastructure (CHASE-CI), NSF ACI-1541349 CC*DNI Pacific Research Platform, and gifts from Qualcomm and TuSimple.

References

[1] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *ICCV*, 2015. 2

[2] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, 2008. 2

[3] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019. 3

[4] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 941–951, 2019. 2

[5] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016. 2, 5

[6] Thomas Brox, Christoph Bregler, and Jitendra Malik. Large displacement optical flow. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 41–48. IEEE, 2009. 2

[7] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004. 2

[8] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, pages 611–625. Springer, 2012. 2

[9] S. Caelles, K.K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *CVPR*, 2017. 1

[10] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems*, pages 9912–9924, 2020. 2, 3

[11] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 5

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2, 3, 5, 7

[13] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2, 3, 4, 5, 7

[14] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020. 2, 3, 4, 5, 7, 8

[15] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. In *CVPR*, 2000. 2

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7

[17] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015. 2

[18] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 2

[19] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1734–1747, 2015. 2

[20] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick Van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. *arXiv*, 2015. 1

[21] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 2

[22] Daniel Gordon, Kiana Ehsani, Dieter Fox, and Ali Farhadi. Watching the world go by: Representation learning from unlabeled videos, 2020. 2, 3, 5, 8, 12

[23] Ross Goroshin, Joan Bruna, Jonathan Tompson, David Eigen, and Yann LeCun. Unsupervised learning of spatiotemporally coherent metrics. *ICCV*, 2015. 2

[24] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, pages 21271–21284, 2020. 3, 5

[25] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 2

[26] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 2, 3, 8

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 5, 8

[28] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192. PMLR, 2020. 2, 3

[29] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):583–596, 2014. 2

[30] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 1981. 1, 2

[31] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 12

[32] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017. 2

[33] Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. In *Advances in Neural Information Processing Systems*, pages 19545–19560, 2020. 1, 2, 3, 5, 7, 8, 12, 13

[34] Dinesh Jayaraman and Kristen Grauman. Learning image representations tied to ego-motion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1413–1421, 2015. 2

[35] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 3192–3199, 2013. 5, 7, 12, 15

[36] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Forward-backward error: Automatic detection of tracking failures. In *ICPR*, 2010. 2

[37] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *TPAMI*, 2012. 2

[38] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 5

[39] Zihang Lai, Erika Lu, and Weidi Xie. Mast: A memory-augmented self-supervised tracker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6479–6488, 2020. 3, 7, 8

[40] Zihang Lai and Weidi Xie. Self-supervised learning for video correspondence flow. In *BMVC*, 2019. 3, 8

[41] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018. 2

[42] Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. In *Advances in Neural Information Processing Systems*, pages 317–327, 2019. 2, 3, 5, 7, 8

[43] Yin Li, Manohar Paluri, James M. Rehg, and Piotr Dollár. Unsupervised learning of edges. In *CVPR*, 2016. 2

[44] Yang Li and Jianke Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *ECCV*, 2014. 2

[45] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *TPAMI*, 2011. 2

[46] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *ICLR*, 2017. 5

[47] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *DARPA Image Understanding Workshop*. Vancouver, British Columbia, 1981. 2

[48] Michaël Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv*, 2015. 2

[49] Etienne Mémin and Patrick Pérez. Dense estimation and object-based segmentation of the optical flow with robust techniques. *IEEE Transactions on Image Processing*, 1998. 2

[50] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020. 2, 3

[51] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. 2

[52] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3

[53] Pan Pan, Fatih Porikli, and Dan Schonfeld. Recurrent tracking using multifold consistency. In *Proceedings of the Eleventh IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2009. 2

[54] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *CVPR*, 2017. 2

[55] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 2

[56] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 2, 5, 12, 14

[57] Deva Ramanan, David A Forsyth, and Andrew Zisserman. Strike a pose: Tracking people by finding stylized poses. In *CVPR*, 2005. 1, 2

[58] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4161–4170, 2017. 2

[59] Ishwar K Sethi and Ramesh Jain. Finding trajectories of feature points in a monocular image sequence. *IEEE Transactions on pattern analysis and machine intelligence*, (1):56–73, 1987. 1, 2

[60] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using LSTMs. *arXiv*, 2015. 2

[61] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010. 2

[62] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018. 2

[63] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 2, 3

[64] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020. 2

[65] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 5

[66] Jack Valmadre, Luca Bertinetto, Joao Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2805–2813, 2017. 1, 2

[67] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 391–408, 2018. 1, 3, 8

[68] Ning Wang, Yibing Song, Chao Ma, Wengang Zhou, Wei Liu, and Houqiang Li. Unsupervised deep tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1308–1317, 2019. 3

[69] Naiyan Wang and Dit-Yan Yeung. Learning a deep compact image representation for visual tracking. In *NIPS*, 2013. 2

[70] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2015. 2

[71] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2566–2576, 2019. 1, 2, 3, 5, 7, 8

[72] Hao Wu, Aswin C Sankaranarayanan, and Rama Chellappa. In situ evaluation of tracking algorithms using time reversed chains. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 2

[73] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *TPAMI*, 2015. 2, 5, 12, 15

[74] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 2, 3, 5

[75] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 305–321, 2018. 5

[76] Changjiang Yang, Ramani Duraiswami, and Larry Davis. Efficient mean-shift tracking via a new similarity measure. In *CVPR*, 2005. 2

[77] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. 2018. 1

[78] Ting Yao, Yiheng Zhang, Zhaofan Qiu, Yingwei Pan, and Tao Mei. Seco: Exploring sequence supervision for unsupervised representation learning. In *35th AAAI Conference on Artificial Intelligence*, 2021. 5, 8, 12

[79] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6210–6219, 2019. 3

[80] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 2

[81] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 2

[82] Qixian Zhou, Xiaodan Liang, Ke Gong, and Liang Lin. Adaptive temporal encoding network for video instance-level human parsing. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1527–1535, 2018. 5, 7, 12, 14

A. Implementation Details

Fine-grained Correspondence We apply recurrent inference strategy for fine-grained correspondence. To be more specific, we calculate the similarity between the current frame with the first frame ground truth labels as well the prediction results in the preceding m frames. Then the labels of top-k most similarly pixels are selected and propagated to the current frame. We only compute the similarity between features that are at most r pixels away from each other, i.e. *local* attention. The detailed hyperparamter setting for each dataset are listed in Table A.1

	DAVIS	VIP	JHMDB
top-k	10	10	10
preceding frame m	20	8	4
propagation radius r	12	20	20

Table A.1. Fine-grained Correspondence Inference Hyperparameter.

Object-level Correspondence For the fair comparison, we use fine-tuning setting when comparing with previous approaches [22, 78]. Specifically, an additional 1×1 convolution is placed on top of the backbone to transform the frozen representation. Note that only this 1×1 convolution is learnable during fine-tuning. So such protocol could be considered as the linear evaluation. We fine-tune the the 1×1 convolution layer on the GOT-10K [31] dataset, which consists of $\sim$ 10,000 video clips and 1.4 million frames. Adam optimizer is adopted during fine-tuning. The learning rate is initialized to 0.001 and decays by 0.9 every epoch. There is no weight decay. The network is fine-tuned for 50 epochs. The batch size is 8 for all experiments. The inference hyperparamters are the same for without fine-tuning and with fine-tuning setting.

B. Visualization

Without fine-tuning on any additional dataset, the fine-grained correspondence are directly evaluated on the res_4 features of pre-trained ResNet. We visualize our correspondence on 3 downstream tasks and datasets in Figure B.2,B.4,B.3, i.e. video object segmentation on DAVIS-2017 [56], human pose tracking on JHMDB [35], and human part tracking on VIP [82]. For DAVIS and VIP, there are usually more than one instances/parts. Our approach could output tight boundaries around the multiple target areas. For example, in the last row of Figure B.3, the human parts could still be segmented when more people appears in the video. In human pose tracking, even though each joint is propagated individually, we could still estimate the pose accurately. We also compare our VFS with state-of-the-art method [33] in Figure B.1. As last three rows illustrated, our VFS has less false positive object segmentation than [33]. It indicates that our VFS is more robust to distinguish similar pixels. Note that the inference hyperparamters for both methods are the same, the only difference is the pre-trained representation weight.

We use fine-tuned res_5 features for object-level correspondence on OTB-100 visual object tracking [73]. The results are visualized in Figure B.5. Our VFS could robustly track the target object even under difficult scenarios. For example, in the first row, there are multiple similar basketball players, and tracking target undergoes complicate object interaction as well as occlusion. Similarly for the deer in the third row, where the tracking target overtakes other similar deers. For the jumping person in the last row, the video suffers motion blur and large camera displacement.

We provide more visualization in our project page .

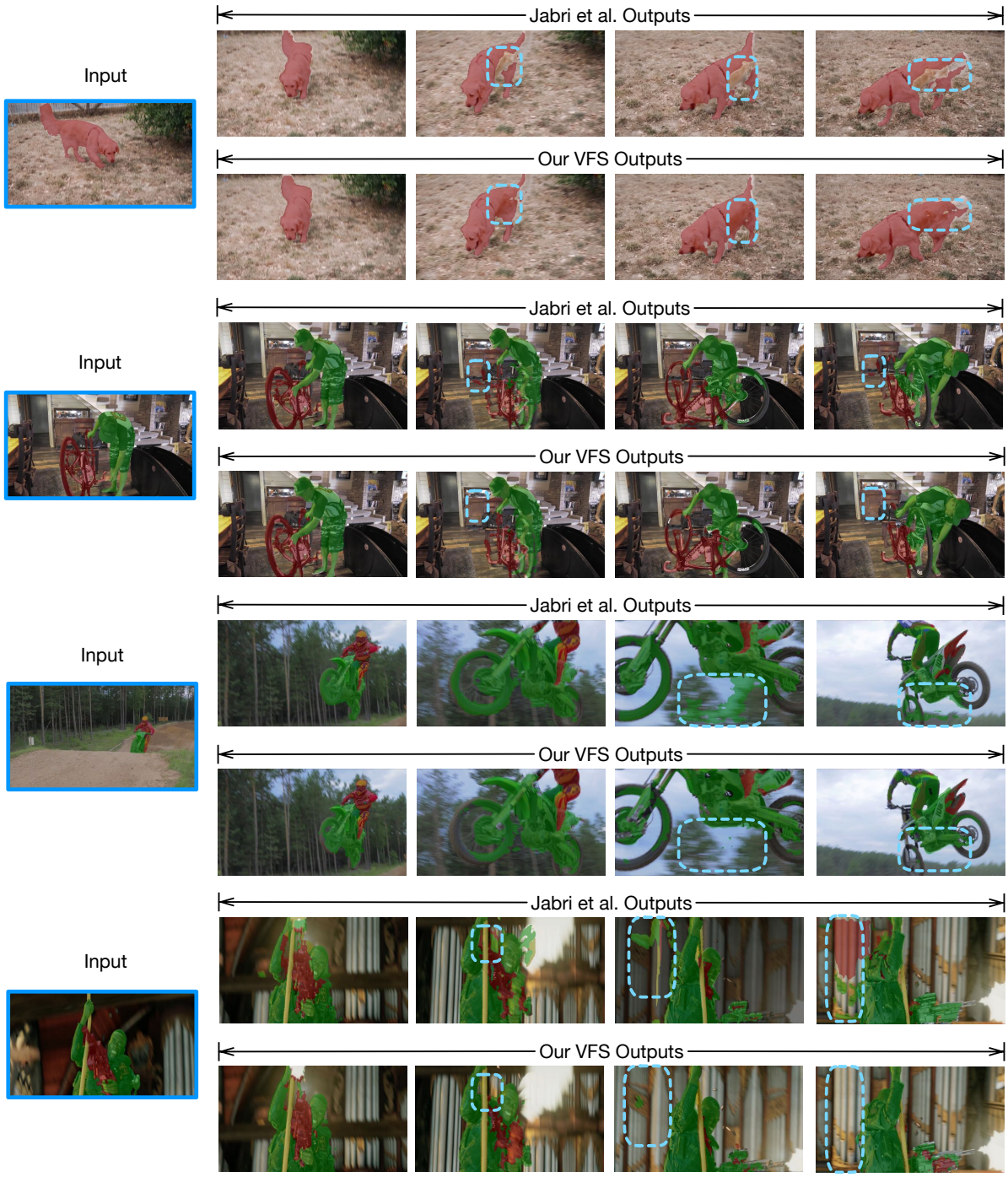

Figure B.1. **Compare Fine-grained Correspondence on DAVIS**. Comparing with previous state-of-the-art Jabri et al. [33], our VFS could generate results of higher quality and with less false positives. Blue dash areas indicate failure cases in [33], where our approach could output plausible results. More comparison are provided in the project page.

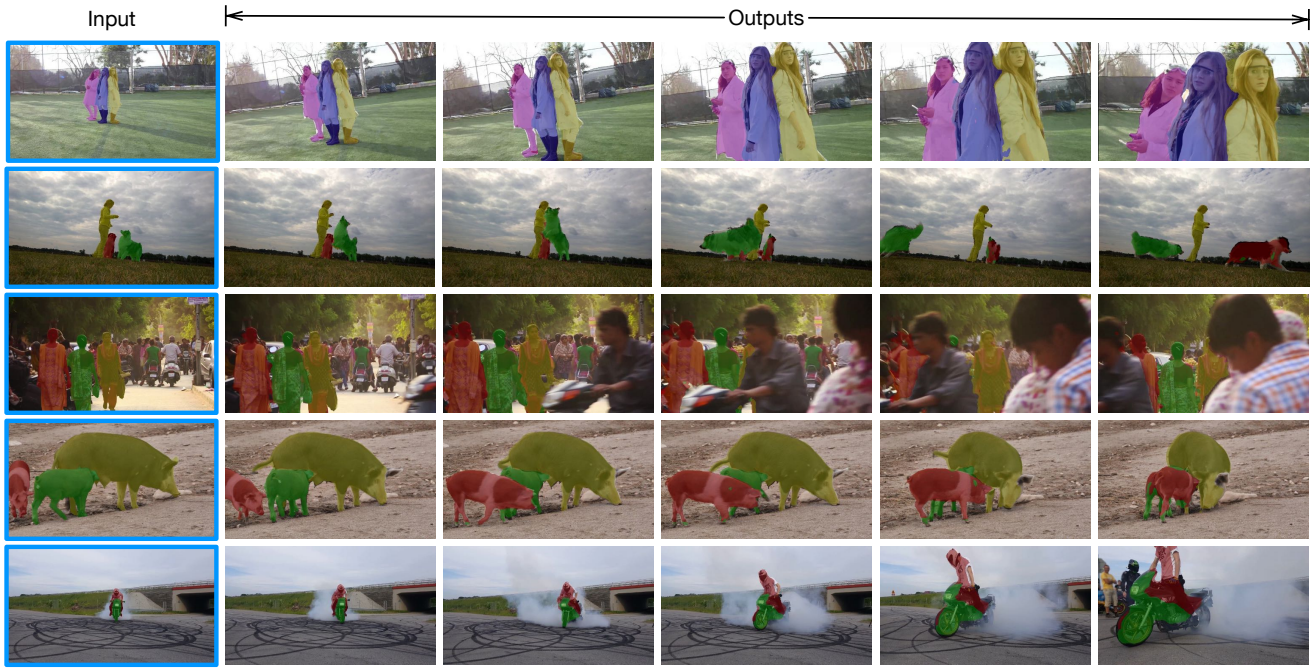

Figure B.2. Qualitative Results for video object segmentation on DAVIS-2017 [56].

Figure B.3. Qualitative Results for human part tracking on VIP [82].

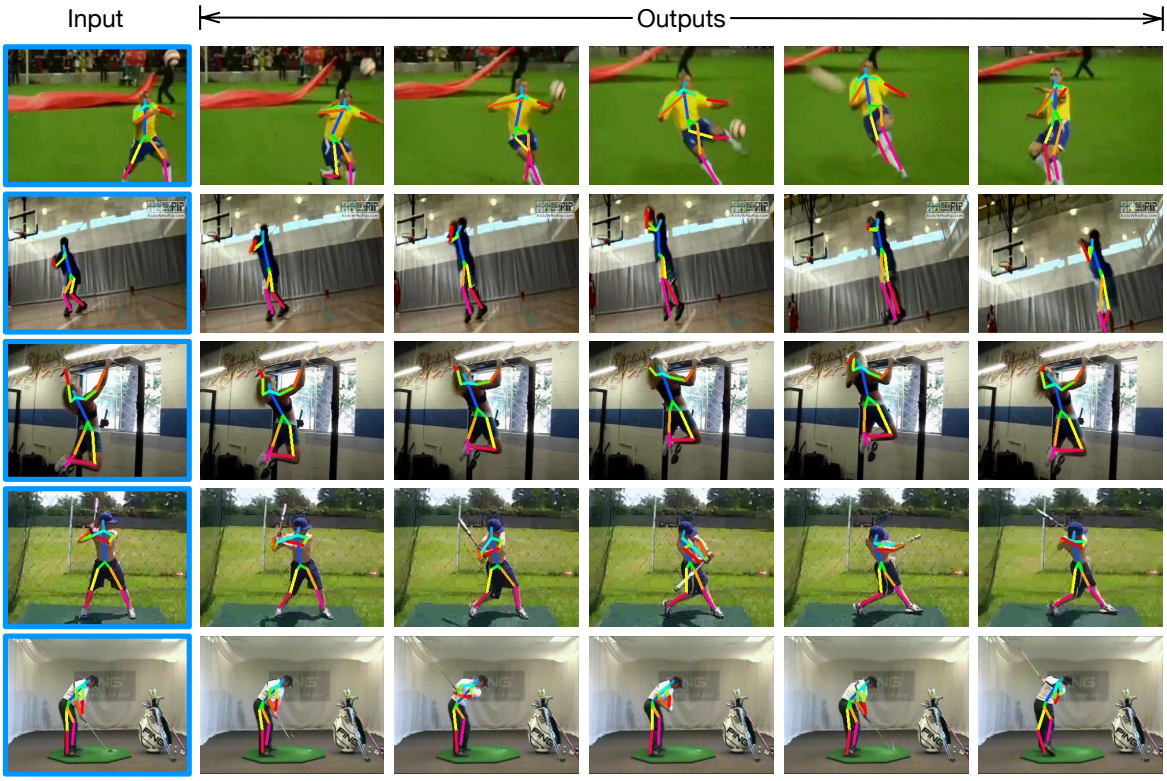

Figure B.4. Qualitative Results for human pose tracking on JHMDB [35].

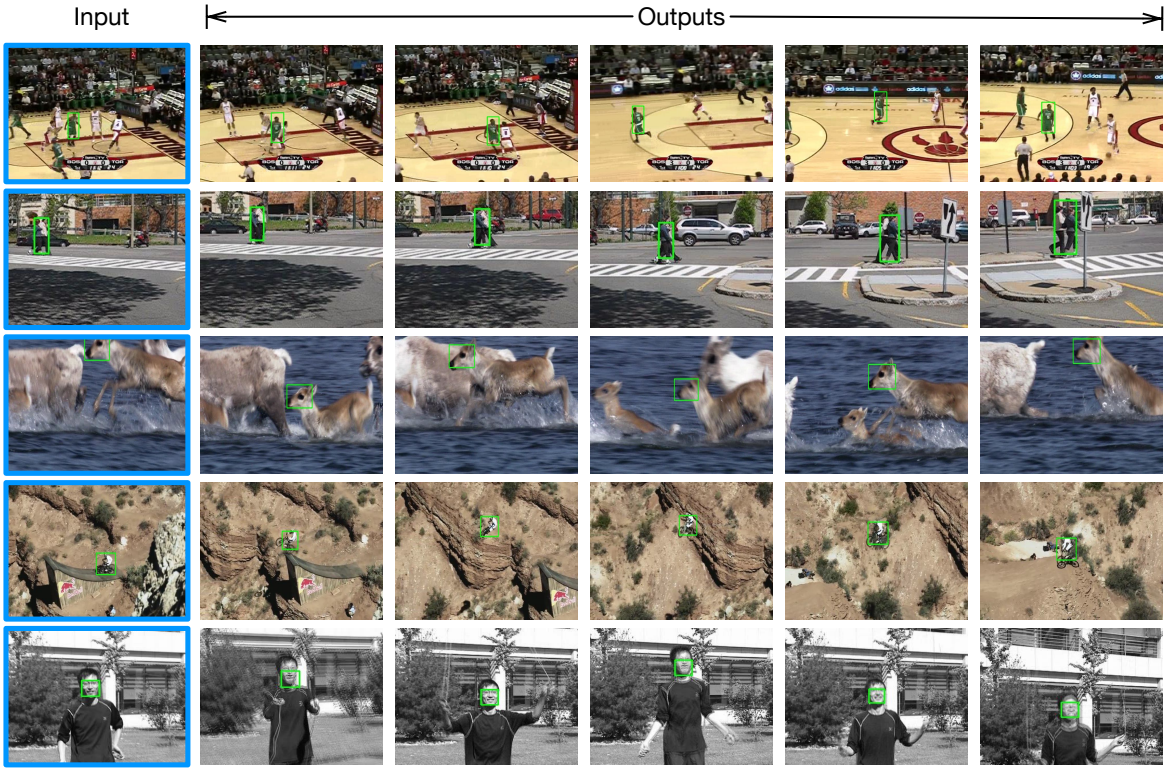

Figure B.5. Qualitative Results for visual object tracking on OTB-100 [73].