Self-supervised Motion Representation via Scattering Local Motion Cues

Yuan Tian, Zhaohui Che, Wenbo Bao, Guangtao Zhai(✉), and Zhiyong Gao

Shanghai Jiao Tong University, Shanghai, China
{ee_tianyuan,chezhaohui,baowenbo,zhaiguangtao,zhiyong.gao}@sjtu.edu.cn

Abstract. Motion representation is key to many computer vision problems but has never been well studied in the literature. Existing works usually rely on the optical flow estimation to assist other tasks such as action recognition, frame prediction, video segmentation, etc. In this paper, we leverage the massive unlabeled video data to learn an accurate explicit motion representation that aligns well with the semantic distribution of the moving objects. Our method subsumes a coarse-to-fine paradigm, which first decodes the low-resolution motion maps from the rich spatial-temporal features of the video, then adaptively upsamples the low-resolution maps to the full-resolution by considering the semantic cues. To achieve this, we propose a novel context guided motion upsampling layer that leverages the spatial context of video objects to learn the upsampling parameters in an efficient way. We prove the effectiveness of our proposed motion representation method on downstream video understanding tasks, *e.g.*, action recognition task. Experimental results show that our method performs favorably against state-of-the-art methods.

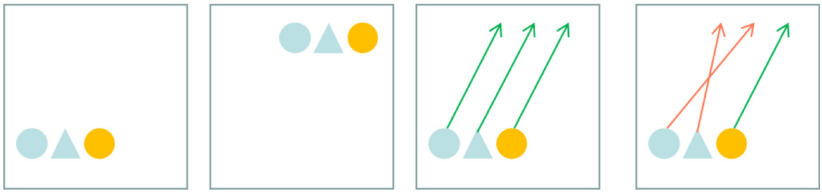
Keywords: Motion representation · Self-supervised learning · Action recognition

1 Introduction

Motion serves as an essential part of video semantic information, and has led to great breakthroughs in numerous tasks such as action recognition [17,21,62], video prediction [33,46], video segmentation [66,71], to name a few. Existing literature typically represents motions in the form of 2-dimensional optical flow vectors. However, optical flow estimation algorithms usually suffer from expensive computational cost or inaccurate estimates [15,24]. More seriously, recent deep learning based approaches rely on human-labeled ground-truths that are labor-consuming [4], or computer-generated synthetic samples [24,25] that may cause domain gap with natural realistic scenes. Therefore, there exists an urgent demand for unsupervised learning of motion representations.

Amounts of tasks rely on accurate motion representations. Action recognition methods [8,21,50,55,59] usually take motion modalities, *e.g.*, optical flow stream as the additional input besides RGB frames to further improve the performance. Many works in video generation [2,32,34,35,46,48] learn to predict

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12359, pp. 71–89, 2020.
https://doi.org/10.1007/978-3-030-58568-6_5



(a) Frame t (b) Frame $t+1$ (c) GT Motion (d) Possible Motion

Fig. 1. Schematic diagram of motion. The circle and the triangle shapes represent two different semantic groups, while the pixels of different brightness are marked with different colors. (a) and (b) are two consecutive frames. (c) is the ground-truth motion along the time step. (d) is one possible but wrong solution by only considering the brightness consistency rule.

the motions of the objects in the frame as an intermediate step. Video segmentation works [10,39,66,71] highly depend on accurate motion information to find prominent objects from frames. However, there are few works focusing on the learning of motion representation. Sun et al. [54] proposed a motion representation in feature space and shared a similar formula definition with optical flow. However, the training of their networks involves large-scale category labeled video datasets. Choutas et al. [11] proposed the PoTion method that aggregates human joints' heatmap in each frame to the pose motion representation, which is limited to the category of human motions.

Motivation: The basic self-supervised paradigm of learning motion representation is (1) first predicting a per-pixel transformation from a pair of consecutive frames and (2) then minimizing the photometric loss, *e.g.*, ℓ_1 loss between the transformed second frame and the ground-truth. So, what's the main obstacle for learning the accurate motion representation beneficial to down-stream tasks? **We argue that the current works ignore the correlation between the local motion and the high-level semantic constraints.** As shown in Fig. 1, by only considering the brightness consistency rule, one possible motion solution is (d), which is less semantic and is harmful to downstream tasks.

To tackle the above mentioned problem, we propose a coarse-to-fine motion network to extract motion maps of both high accuracy and semantics from the input video in a self-supervised manner. In the coarse stage, the network decodes the low-resolution motion maps from the video features. In the refined stage, the network upsamples the motion maps from the previous stage to high-resolution. Moreover, to make the upsampling operation learnable, the motion maps are interpolated by our proposed Context Guided Motion Upsampling Layer (CGMUL) instead of the traditional bilinear upsampling. CGMUL is carefully designed to exploit the local motion-semantics correlation in feature space for producing the full-scale motion features and aggregate the features into high-resolution motion maps in an efficient way.

To fully utilize the long-term temporal semantics in videos, our method takes video clips instead of frame pairs as input and adopt the off-the-shelf 3D CNNs,

e.g., C3D [56], 3DResNet [21] or SlowFast Network [17] as the video feature extractor. This reduces the semantic gap between our learned motion representations and other video understanding tasks based on these 3D CNNs. Additionally, our learning process can regularize the backbone 3D CNNs without increasing the computation cost at inference time in two ways, (1) improving the performances of other tasks in a multi-task fashion and (2) serving as a pre-training method for the backbone network.

Our contributions are summarized from the following aspects:

First, we further restrict the search space of self-supervised motion representation learning by leveraging the motion-semantics correlations in local regions. The resulting representations are accurate, of high semantics and beneficial to downstream video understanding tasks.

Second, we propose a Context Guided Motion Upsampling Layer (CGMUL) to learn the motion map upsampling parameters by exploiting the correlation between the semantic features of spatial contexts and local motions.

Third, we show that our method reaches a new state-of-the-art performance on action recognition task. Moreover, the motion representation ability of our method is competitive to other recent optical flow methods, *e.g.*, FlowNet2.0 [24].

2 Related Work

Motion Representation Learning. The most common motion representation is optical flow. Numerous works [24,25] attempt to produce flow map from coupled frames by CNN in an efficient way. However, most of their training datasets are synthetic and thus they perform poorly on real word scenes. Recently, other motion representations have been proposed. TSN [62] leverages the RGB difference between consecutive frames. OFF [54] proposes an optical flow alike motion representation in feature space. PoTion [11] temporally aggregates the human joints' heatmap in each frame to a clip-level representation with fixed dimension. In contrast, our method is self-supervised and learns more general motion representations for both articulated objects and dynamic textures.

Dynamic Filter Networks. DFN [3] first proposes to generate the variable filters dynamically conditioned on the input data. DCN [12] can also produce position-specific filters. PAC [52] proposes a pixel-adaptive convolution operation, in which the convolution filter's weights are multiplied with a spatially varying kernel. Unlike them, we produce the dynamic motion filters directly from the video features for individual spatial position.

Video Prediction. Video prediction relies on the motion cues in feature space or primal space to synthesize the future frame from past frames. BeyondMSE [36] adopts a cGAN model to make use of temporal motion information in videos implicitly. Recent works take advantage of motion cues embodied in videos by flow consistency [33,42], the retrospective cycle nature of video [29], the static and dynamic structure variation [69], etc. However, their motion generators usually adopt the frame-level spatial feature extractor. Some works such as SDC [46] rely on optical flow map and dynamic kernel simultaneously to synthesize the

future frames. In comparison, our method makes use of the rich spatial-temporal features from long-term videos without leveraging optical flow maps.

Action Recognition. Recently, convolutional networks are widely adopted in many works [19,50,56,61] and have achieved great performance. Typically, two-stream networks [19,50,61] learn motion features based on extra optical flow stream separately. C3D network [56] adopts 3D convolution layers to directly capture both appearance and motion features from raw frames volume. Recent deep 3D CNN based networks [8,17,21] such as 3D-RestNet [21] have been trained successfully with the promising results upon the large scale video datasets. Our work is built upon the 3D CNNs and surpass their performance.

3 Proposed Method

In this section, we first provide an overview of our motion representation algorithm. We then introduce the proposed context guided motion upsampling layer, which plays a critical role in learning the accurate full-resolution motion maps. Finally, we demonstrate the design of all the sub-modules and clarify the implementation details of the proposed model.

3.1 Overview

Different from the previous motion representation methods that only take two consecutive frames as input, we feed a video clip consisting of T frames into the network to craft T motion maps simultaneously, where the first $T-1$ motion maps reflect the motion representations between every consecutive frame pair, while the last one is a prediction of the possible motion *w.r.t.* the next unknown future frame. Our method shares the video's spatial-temporal features with other tasks, *e.g.*, action recognition and benefits them in a multi-task paradigm. Moreover, the learned motion maps can serve as another input modality to further improve the performances of these downstream tasks.

3.2 Context Guided Motion Upsampling Layer

Motion Map: We first give a principled definition to the motion map in our method. Given input video $\mathbf{X} \in \mathbb{R}^{t \times w \times h \times c}$, the motion maps are composed of a series of local filters of size $k \times k$, each of which models the localized motion cues around the center pixel, where t, w, h, c and k denote video temporal length, video frame width, video frame height, the number of video frame channels and the constant parameter indicating the maximum displacement between consecutive frames. Let us denote the motion map by $\mathbf{M}_t \in \mathbb{R}^{k \times k \times w \times h}$, which describes the motions between $\mathbf{X}_t$ and $\mathbf{X}_{t+1}$. These three tensors are related by the pixel-adaptive convolution [52] operation, which can be precisely formulated as:

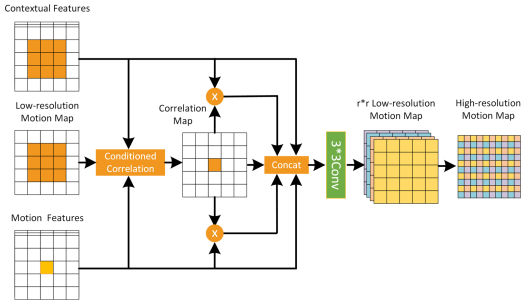


Fig. 2. Context Guided Motion Upsampling Layer. The layer exploits the correlation between the contextual features and the motion features to estimate the higher resolution motion maps. "x" and "Concat" denote the channel-wise multiplication and concatenation operations respectively.

$$\tilde{\mathbf{X}}_{t+1,x,y,c_i} = \sum_{m=-r}^{r} \sum_{n=-r}^{r} \mathbf{M}_t \left(m+r, n+r, x, y\right) \times \mathbf{X}_{t,c_i} \left(x-m, y-n\right). \tag{1}$$

where $r = \frac{k-1}{2}$, c_i denotes the color channel. Each filter of the motion map is adaptive to a single pixel of $\mathbf{X}_t$ while shared across the color channels. Compared to optical flow, this definition can represent the motions in sub-pixel level and synthesize novel pixel values to address subtle brightness changes of the moving pixels, which is common in natural scenes.

Learn to Upsample Motion Maps: To learn the non-linear interpolating parameters for upsampling the motion map, we propose the Context Guided Motion Upsampling Layer (CGMUL) to estimate the high-resolution (HR) motion maps from the low-resolution (LR) motion maps, as shown in Fig. 2. The estimation process is guided by the semantic context in the local regions.

Precisely, we denote the contextual feature, the LR motion map, and the motion feature as $F_{context} \in R^{C\times\hat{w}\times\hat{h}}$, $M_{LR} \in R^{k\times k\times\hat{w}\times\hat{h}}$ and $F_{motion} \in R^{C\times\hat{w}\times\hat{h}}$, where $\hat{w} = \frac{w}{r}$, $\hat{h} = \frac{h}{r}$ and r is the upsampling scale.

We first compute the correlational similarity map $S \in R^{\hat{w}\times\hat{h}}$ conditioned by M_{LR} between $F_{context}$ and F_{motion}:

$$\begin{aligned} S(x,y) = \sum_{c=0}^{C} \sum_{m=-r}^{r} \sum_{n=-r}^{r} (M_{LR}(m+r, n+r, x, y)\times \\ F_{motion}(c,x,y) \times F_{context}(c, x-m, y-n)) \times \frac{1}{C}, \end{aligned} \tag{2}$$

where $r = \frac{k-1}{2}$ and $\frac{1}{C}$ are used for normalization. The similarity describes the relationship between three inputs explicitly.

Recent studies [31] proposed to enhance the feature's discrimination by dot-producting the channel-wise pooling features. By analogy, Eq. 2 can be viewed

as the soft fusion of the channel-wise pooling features derived from $F_{context}$ and F_{motion}. Thus, we produce the enhanced features in the following way:

$$\begin{aligned} F'_{context}(c) &= S \cdot F_{context}(c), \\ F'_{motion}(c) &= S \cdot F_{motion}(c), \end{aligned} \tag{3}$$

The final context guided motion feature is the concatenation of the features above along the channel dimension, given by

$$F = cat(F_{context}, F_{motion}, S, F'_{context}, F'_{motion}), \tag{4}$$

We perform a learnable 3×3 convolution on F to produce feature maps $F' \in R^{(r \cdot r \cdot \hat{k} \cdot \hat{k}) \times \hat{w} \times \hat{h}}$. Finally, we utilize the periodic shuffling operator [49] on the feature maps above to get the HR motion map $M_{HR} \in R^{(\hat{k} \cdot \hat{k}) \times (r\hat{w}) \times (r\hat{h})}$:

$$M_{HR}(c, x, y) = F'_{C \cdot r \cdot \bmod (y,r) + C \cdot \bmod (x,r) + c, \lfloor x/r \rfloor, \lfloor y/r \rfloor}, \tag{5}$$

where $C = \hat{k} \cdot \hat{k}$ and c denotes the channel index. Noting that $\hat{k}$ is bigger than k , for the motion filters in motion maps of higher resolution require wider receptive field.

3.3 Context Guided Motion Network

As shown in Fig. 3, the proposed Context Guided Motion Network (CGM-Net) consists of the following submodules: the video encoder, the LR motion decoder, the context extractor, and the motion upsampler. We adopt the proposed context guided motion upsampling layer to upsample the LR motion map to the HR motion map in a learnable way. We illustrate every component in detail as follows.

Video Encoder. This module extracts compact video features $\mathcal{F}_v$ from the input video clip $\mathbf{X}$, which mainly consists of a series of 3D convolution operations. Notably, the proposed method is compatible with most recent off-the-shelf 3D CNNs [8,17,56]. In our experiment, due to the space limitation, we only report the performance when using two landmark 3D CNNs (*i.e.* 3D-ResNet [21] and SlowFast network [17]) as feature extractors to derive spatial-temporal features.

LR Motion Decoder. This module reconstructs the LR motion features $\mathcal{F}_{LR}$ from video features $\mathcal{F}_v$ by deconvolution operations. To facilitate the network convergence, we replace all deconvolution operations in the network with a bilinear upsample operation followed by a convolution operation with the kernel size of 3 and the stride of 1 as suggested by the previous research [70].

Context Extractor. This module extracts semantic contextual information from each frame of the input video. We utilize the response of the $conv3\_x$ layer from ResNet-18 [23] as the contextual features and remove the max-pooling layer between the $conv\_1$ and $conv2\_x$ to maintain a high spatial resolution of the contextual features.

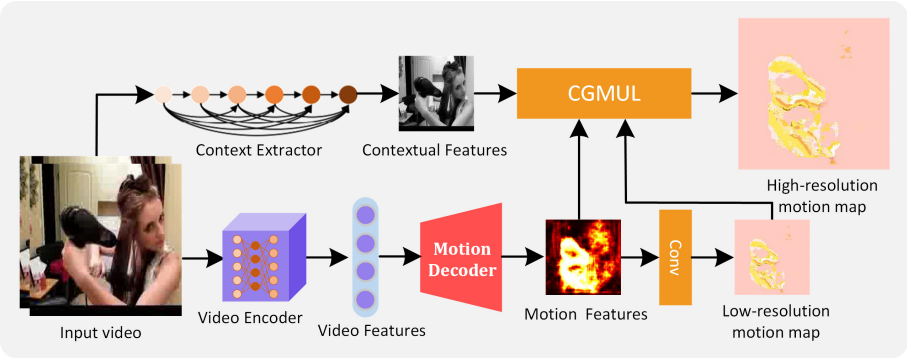


Fig. 3. Architecture of the proposed context guided motion network. CGMUL denotes the Context Guided Motion Upsampling Layer, as illustrated in Fig. 2. Given the input video, we extract the rich spatial-temporal features from the entire video clip, and extract a group of contextual features from every single frame, respectively. We first decode the low-resolution motion maps from spatial-temporal features directly following an encoder-decoder paradigm. We then adopt the proposed CGMUL to upsample the motion maps adaptively following the semantic guide of the contextual features. The final output high-resolution motion map is of both high accuracy and semantics.

3.4 Enhancing Action Recognition

After obtaining the motion maps, we feed them into a light-weight action CNN to boost the video action recognition task, because our motion maps capture more semantics than the vanilla RGB images. Concretely, the action CNN utilizes six convolution layers and one fully-connected layer to predict the action category of the input video. It is worth mentioning that we can also perform classification by adding one fully-connected layer after the backbone network directly. We fuse the prediction scores of these two methods to boost action recognition in the test.

3.5 Training Strategy

Self-supervised Learning. When learning motion representations, CGM-Net aims to (1) reconstruct all input frames and (2) predict the next future frame after the input clip simultaneously. The output frames are computed as $\tilde{\mathbf{X}}_{t+1} = \mathbf{M}_t \otimes \mathbf{X}_t$ as defined in Eq. 1, where $\mathbf{M}_t$ is the predicted motion maps. We train the network by optimizing the following reconstruction loss:

$$\mathcal{L}_{HR} = \sum_{t=0}^{T} \rho\left(\tilde{\mathbf{X}}_{t+1} - \mathbf{X}_{t+1}\right), \tag{6}$$

where $\rho(x) = \sqrt{x^2 + \epsilon^2}$ is the Charbonnier penalty function [9]. We set the constant ϵ to 0.000001.

Intermediate Supervision: To facilitate the optimization of the internal LR motion maps, we also exploit the downsampled input videos as the intermediate self-supervised supervision. The LR resconstruction loss $\mathcal{L}_{LR}$ follows the same formulation as Eq. 6.

Multi-task Loss: When learning with the full-supervised classification task, we formulate a multi-task loss as

$$\mathcal{L} = \mathcal{L}_{HR} + \lambda_1 \mathcal{L}_{LR} + \lambda_2 \mathcal{L}_c. \tag{7}$$

where $\mathcal{L}_c$ is the action classification loss (*e.g.* the cross entropy), λ_1 and λ_2 are the hyper-parameters to trade-off these losses.

4 Experimental Results

We first carry out comparisons between our method and other recent methods regarding the motion representation ability. Then, we show that our method can facilitate the action recognition performance of the very recent 3D CNNs to achieve the new state-of-the-arts while keeping efficient. Finally, we conduct extensive ablation studies to verify the every aspects of the proposed method.

4.1 Comparison with Other Motion Representation Method

To solve the occlusion and color noise problems in the natural scenes, our method synthesizes novel pixels not in the previous frame. Therefore, we compare the motion representation errors on the natural scene dataset, *i.e.*, the UCF101 dataset [51]. We compare our method with other methods in terms of (1) motion estimation and (2) 1-step frame prediction.

Dataset. UCF-101 is a widely-used video benchmark including 101 human action classes. We choose 20 videos with neat backgrounds and obvious motions, named **UCF-Flow**, to compare the performance of motion estimation. We select 101 videos of different actions, named **UCF-Pred**, to compare the performance of frame synthesis on the video prediction task.

Implementation Details. For our method, we split the videos into clips of 16 frames and discard the too short clips. For other optical flow methods, we compute the optical flow for every two consecutive frames. We apply the ℓ_1 error between the warped second image and the ground-truth second image instead of End-Point-Error (EPE) to measure the motion representation error, for our motion map can't be transformed to an optical flow map losslessly. All the images are normalized to the range $[-1, 1]$ before computing the error. In the training process, we set the λ_1 and λ_2 of Eq. 7 as 1.0 and 0 (we do not leverage the video ground-truth label in this part) respectively. We adopt Adam optimizer [26] with a start learning rate as 0.001 and reduce the learning rate every 50 epochs.

Motion Estimation Results. As shown in Table 1 (left), our method substantially outperforms the best optical-flow methods by a large margin in terms of

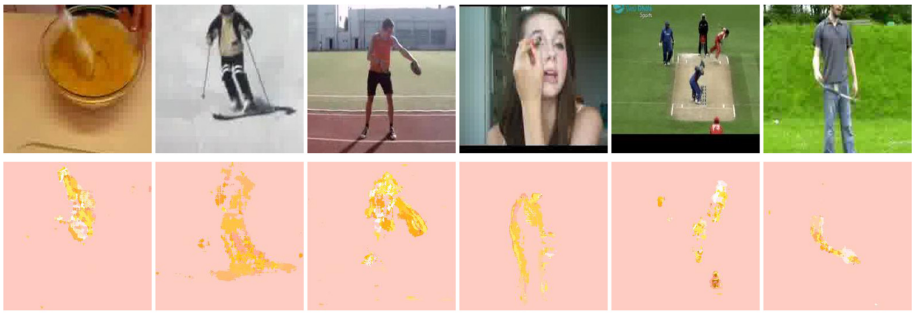


Fig. 4. The visualization of our high-resolution motion maps on UCF-Flow set. The predominant motions are consistent with the semantics of the objects. Our method is robust to human action and natural object's motion. For example, in (a) and (d), the motion of objects (*i.e.* spoon and human body) are predicted accurately. In (e), our motion map shows excellent performance in the multi-objects scenario.

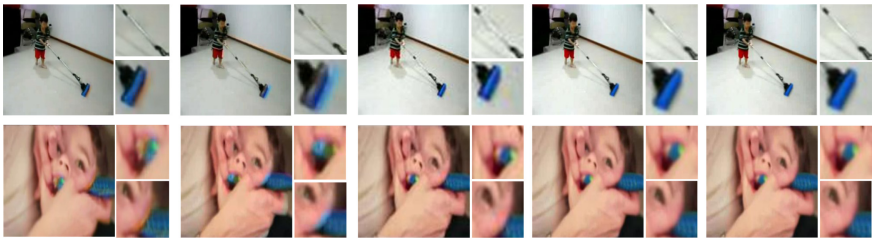
Table 1. *left*: Comparison of motion estimation methods on UCF-Flow. The top part shows the performance of the current best traditional optical flow estimation methods. The middle part shows the results of the CNN based methods. *right*: Comparison of video prediction methods on UCF-Pred.

Method	$\ell 1$ Err	Train
DIS-Fast [27]	0.055	No
Deepflow [65]	0.058	No
TV-L1 [44]	0.037	No
Flownet2.0 [24]	0.057	Yes
PWC-Net [53]	0.049	Yes
TV-Net-50 [16]	0.040	Yes
Ours	**0.018**	Yes

Method	PSNR	SSIM
BeyondMSE [36]	32	0.92
ContextVP [5]	34.9	0.92
MCnet+RES [58]	31	0.91
EpicFlow [47]	31.6	0.93
DVF [35]	33.4	0.94
Ours (112px)	**35.0**	**0.96**
Ours (256px)	**36.3**	**0.96**

ℓ_1 Error. The explanation for the obvious improvement upon the optical flow based methods is that the environment illuminations change constantly and most objects are not rigid in natural scenes. Our motion map can synthesize new pixels around the moving objects. To further prove our method represents the motions precisely and robustly, we show the visualization of the motion map for diverse human actions on UCF101 in Fig. 4.

1-Step Frame Prediction Results. For quantitative evaluation, we utilize the SSIM and PSNR [63] as the evaluation metrics. The higher SSIM and PSNR, the better prediction performance. Table 1 (right) describes the quantitative evaluation results of the state-of-the-art methods and the proposed method on UCF101. Our method achieves the best results in terms of SSIM and PSNR. Moreover, even with lower input resolution of 112×112, the performance of our method



(a) MCNet [59] (b) ConvLSTM [5] (c) DVF [35] (d) **Ours** (e) Ground-truth

Fig. 5. Qualitative comparisons of the predicted frame on the UCF-Pred set. Our method not only mitigates the blur artifacts around the ambiguity region but also reduces the distortion on the background.

also keeps stable and competitive. This demonstrates our motion representations are learned from either spatially localized textures or global semantic information that are not sensitive to video resolution.

Figure 5 depicts a few results of the preceding methods, where the frames predicted by our method obtain better perceptual quality [37,38,73], even our method is not equipped with the perceptual loss. MCNet [58] and ConvLSTM [5] cause ghosts in the regions around the blob objects, *e.g.*, the mop head, because they encode both the motion and content features into their models. DVF [35] shows unexpected artifacts in background regions around the motion objects because it doesn't synthesize novel pixels around the motion regions. However, the motion filters of our motion map are adaptive to the position and thus only acts on the activities related regions in a pixel synthesis way.

4.2 Action Recognition

Datasets. We evaluated our method on three large-scale general human action and activity datasets, including Kinetics [8], UCF101 [51], and HMDB51 [28]. We follow the original training/test splits and protocols provided by the datasets. We report the mean average accuracy over the three splits for HMDB51 and UCF101. For Kinetics, we report the performance on the validation set.

Kinetics. Kinetics is a challenging human action recognition dataset. We evaluate our method on Kinetics-400 and Kinetics-600. We report top-1 and top-5 classification accuracy (%) on Kinetics.

UCF101. UCF-101 includes 13,320 action instances from 101 human action classes. To evaluate the action recognition performance, we first train it on the Kinetics-400 dataset and then fine-tune on UCF101. For this dataset, we also study the effectiveness of our method as a pre-training strategy compared with other self-supervised pre-training methods.

HMDB51. HMDB51 includes 6,766 videos from 51 human action classes. On this dataset, we conduct all action recognition experiments mentioned in UCF101.

Table 2. *left*: Comparison of self-supervised action representation methods. The baseline methods in first group are without self-supervised pretraining. *right*: Evaluation of training on different sets of Kinetics-400. ResNext-101 and STC-ResNext101 are abbreviated as R101 and S-R101 respectively. * indicates that the corresponding method uses extra unlabeled data.

Method	UCF101	HMDB51
3D-R101 [21]	56.2	30.3
S-R101 [13]	56.7	30.8
Shuffle and learn [40]	50.9	19.8
OPN-RGB [30]	71.8	36.7
Order Prediction [68]	72.4	30.9
Odd-One-Out [20]	60.3	32.5
AOT-RGB* [64]	86.5	-
ActionFlowNet* [41]	83.9	56.4
DynamoNet(3D-R101)* [14]	87.3	58.6
DynamoNet(S-R101)* [14]	**88.1**	**59.9**
Ours(3D-R101)*	**88.1**	**59.0**

Method	Data	Top1
3D-R101 [21]	half	53.9
3D-R101 [21]	full	65.1
S-R101 [13]	half	55.4
S-R101 [13]	full	66.2
St-Net(R101) [22]	half	56.7
St-Net(R101) [22]	full	71.38
DynamoNet(S-R101) [14]	half	63.6
DynamoNet(S-R101) [14]	full	67.67
Ours(3D-R101)	half	**69.8**
Ours(3D-R101)	full	**76.2**

Implementation Details. We train the network with only motion representation branch (λ_2 is set to 0) as the pre-training step on 500K unlabeled video clips from YouTube8M [1] dataset. We first resize the video frames to 128px when smaller and then randomly perform 5 crops (and flips) of size 112×112 as the main network input size. When using the SlowFast [17] as backbone network, we follow the same input size as them. We adopt Adam optimizer [26] with an initial learning rate as 0.001 and batch size of 64 to train the model. In our experiments, we use the different versions of 3D-ResNet/ResNeXt as the backbone networks. Empirically, we obtain the best results when setting the motion map upsampling scale factor = 4, $\lambda_1 = 1$ and $\lambda_2 = 10$. We use the PyTorch [43] framework for the implementation and all the experiments are conducted on sixteen 2080 Ti NVIDIA GPUs.

Self-supervised Action Representation. Since motion is an important cue in action recognition, we argue the learned motion representation implied in the backbone 3D CNN can be adopted as a good initial representation for the action recognition task. Our network is firstly trained on unlabeled video clips to learn motion representation. Then, we fine-tune the full network carefully, with all losses in Eq. 7 activated.

In Table 2 (left), we observe that our method performs better in comparison to state-of-the-art self-supervised methods [14,20,30,40,41,64,68] on UCF101 and HMDB51. The performance gap between our method pretrained by unlabeled data and the 3D-ResNet101 trained on Kinetics-400 (shown in Table 5) is largely reduced to **0.8%** on UCF101. DynamoNet with the STC-ResNeXt101 indeed outperforms our method with 3D-ResNeXt101 by 0.9% on HMDB51 because STC-ResNeXt101 has a stronger ability to capture spatial-temporal correlations compared to vanilla 3D-ResNeXt101.

Table 3. Performance comparisons of our method with other state-of-the-art 3D CNNs on Kinetics-400 dataset. Y500K indicates the subset of Youtube8M.

Method	Flow	Backbone	Pretrain	Top1	Top5
3D-ResNet18 [21]	✗	✗	✗	54.2	78.1
C3D [56]	✗	✗	Sports1m	55.6	-
3D-ResNet50 [21]	✗	✗	✗	61.3	83.1
3D-ResNet101 [21]	✗	✗	✗	62.8	83.9
3D-ResNeXt101 [21]	✗	✗	✗	65.1	85.7
R(2+1)D [57]	✗	✗	✗	73.9	90.9
STC-Net [13]	✗	3D-ResNeXt101	✗	68.7	88.5
DynamoNet [14]	✗	3D-ResNeXt101	Y500K	68.2	88.1
StNet [22]		ResNet101	✗	71.4	-
DynamoNet [14]	✗	STC-ResNeXt101	Y500K	77.9	94.2
SlowFast 16×8 [17]	✗	ResNeXt101	✗	78.9	93.5
R(2+1)D Flow [57]	✓	✗	✗	67.5	87.2
I3D [8]	✓	✗	✗	71.6	90.0
R(2+1)D [57]	✓	✗	✗	73.9	90.9
Two-Stream I3D [8]	✓	BN-Inception	ImageNet	75.7	92.0
S3D-G [67]	✓	✗	ImageNet	77.2	93.0
Ours	✗	3D-ResNet50	Y500K	70.1	90.2
Ours	✗	3D-ResNeXt101	Y500K	**76.2**	**92.3**
Ours	✗	SlowFast16×8	Y500K	**80.8**	**94.5**

Table 4. Performance comparisons of our method with other state-of-the-art 3D CNNs on Kinetics-600 dataset. Y500K indicates the subset of Youtube8M.

Method	Backbone	Pretrain	Top1	Top5
P3D [45]	ResNet152	ImageNet	71.3	-
I3D [7]	BN-Inception	✗	71.9	90.1
TSN [62]	IRv2	ImageNet	76.2	-
StNet [22]	IRv2	ImageNet	79.0	-
SlowFast 16×8 [17]	ResNeXt101	✗	81.1	95.1
Ours	3D-ResNet50	Y500K	76.2	90.7
Ours	3D-ResNeXt101	Y500K	80.2	94.0
Ours	SlowFast16×8	Y500K	**81.9**	**95.1**

Table 2 (right) shows the self-supervised pre-training backbone network based on our method can alleviate the need for labeled data and achieves the best results with datasets of different sizes. Moreover, the performance of our pipeline

trained with half data is competitive with other state-of-the-art methods (*e.g.*, St-Net) trained with full data.

Table 5. Performance comparisons of our method with other state-of-the-art methods on UCF101 and HMDB51. The number inside the brackets indicates the frame number of the input clip. † and * indicate the backbone network is 3D-ResNeXt101 or STC-ResNeXt101 respectively.

UCF101		HMDB51	
Method	Top1	Method	Top1
DT+MVSM [6]	83.5	DT+MVSM [6]	55.9
iDT+FV [59]	85.9	iDT+FV [59]	57.2
C3D [56]	82.3	C3D [56]	56.8
Two Stream [50]	88.6	Two Stream [50]	-
TDD+FV [60]	90.3	TDD+FV [60]	63.2
RGB+Flow-TSN [62]	94.0	RGB+Flow-TSN [62]	68.5
ST-ResNet [18]	93.5	ST-ResNet [18]	66.4
TSN [62]	94.2	TSN [62]	69.5
3D-ResNet101 [21]	88.9	3D-ResNet101 [21]	61.7
3D-ResNeXt101 [21]	90.7	3D-ResNeXt101 [21]	63.8
DynamoNet (16)† [14]	91.6	DynamoNet (16)† [14]	66.2
DynamoNet (32)† [14]	93.1	DynamoNet (32)† [14]	68.5
DynamoNet (64)* [14]	94.2	DynamoNet (64)* [14]	**77.9**
Ours (32)†	**94.1**	**Ours (32)†**	**69.8**

Comparison with the State-of-the-Art. Table 3 presents results on Kinetics-400 for our method. With 3D-ResNeXt101 backbone, our method outperforms DynamoNet, which also ensembles the motion representations in a self-supervised way, with large margins: **8.0%** and **4.2%** improvements in terms of Top1 and Top5 accuracies respectively. This indicates the superiority of our semantic guided motion maps, compared with DynamoNet [14] directly adopting the spatially shared motion kernel weights. Interestingly, we find that our method based on 3D-ResNet50 outperforms the vanilla 3D-ResNet101 obviously, by **7.3%** and **6.3%** improvements in terms of Top1 and Top5 accuracies. As shown in Table 4, our method with SlowFast backbone also achieves the best performances. We also compare our method with the other most recent 3D CNNs taking inputs RGB and optical flow modalities and verify that our method outperforms the best of them by **3.6%** while saving the inference cost *w.r.t.* the computation of optical flow maps. Table 5 demonstrates the state-of-the-art performances achieved by our method compared with the very recent methods on

UCF101 and HMDB51 datasets. DynamoNet [14] outperforms our method on HMDB51 with more input frames (64 vs. 32), because it has been verified [13,14] that the number of input frames has a strong impact on the final performance, and the more input frames, the better performance.

4.3 Ablation Study

In this part, to facilitate the training process, we adopt the 3D-ResNet18 as the backbone network.

Learnable vs. Unlearnable Upsampling Methods. We first emphasize the superiority of our learnable motion upsampling method compared with the traditional methods: (1) nearest neighbour interpolation and (2) bilinear interpolation. For traditional methods, we upsample each channel of the motion maps and exaggerate each motion filter with zero holes following the similar expanding method as dilation convolution kernels [72]. As shown in Table 7 and Table 7, our method substantially outperforms the traditional baselines in both motion representation and action recognition. The traditional motion upsampling methods result in coarse output motion maps whereas our method hallucinates the motion details thanks to the static contexts and the motion prior learned from massive videos. It's also interesting to notice from Table 7 that the LR motion map also benefits the action recognition task obviously by **3.8%** despite the motions in this scale are imperceptible, which indicates the advantage of our motion representation in sub-pixel level.

Impact of Different HR/LR Motion Map Scale Factors. As shown in Fig. 6 (left), the motion estimation performance decreases as the scale factor increases. Besides, when the scale factor < 8, the performance drop is moderate. The trend of Fig. 6 (right) is quite different. When the scale factor $= 1$, we got the worse performance because the motion maps are only decoded from the video features without considering the motion-semantics correlation. When the scale factor is quite large, *e.g.*, 16, the deficiency of the motion details causes the performance drop. The scale factor of 4 produces the best performance result that surpasses the baseline by **6.2%**. Therefore, in all experiments in our paper, we select the scale factor as **4** as a good trade-off between the accuracy and the semantics of the motion map if not specified otherwise.

Computation Cost Analysis. We list the performance and the computation cost of each pipeline above in Table 8. The pipeline only adopting the features from the 3D-ResNet18 backbone CNN outperforms the corresponding baseline by **2.1% without any extra inference-time computation cost**. When fused with the results from the LR motion map, our method outperforms the baseline 3D-ResNet18 by **3.8%**. More importantly, despite using a shallower backbone (*i.e.*, 3D-ResNet18), our method outperforms the stronger baseline 3D-ResNet34 by **0.5%**, demonstrating the lower inference-time computation cost and the better performance. The pipeline fusing the results from both LR and HR motion map shows a superior performance - **90.6%**.

Table 6. Comparison of different motion map upsampling methods.

Method	$\ell 1$ Err
Nearest	0.042
Bilinear	0.037
Ours	**0.024**

Table 7. Comparison of different motion maps for action recognition.

Method	Top1
Backbone	84.4%
+LR	**88.2%**
+LR+HR (nearest)	88.2%
+LR+HR (bilinear)	88.2%
+LR+HR (Ours)	**90.6%**

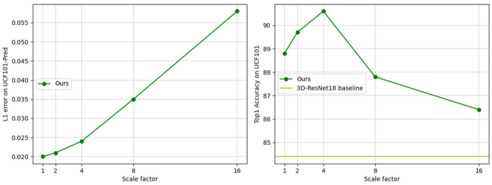


Fig. 6. *left*: The motion representation performances. Lower $\ell 1$ error indicates better motion estimation. *right*: The action recognition performances. Higher Top1 accuracy indicates better performance.

Table 8. Comparison of different pipelines on UCF101. † indicates the result is averaged with the prediction of backbone CNN.

Inference pipeline	Top1	Parameters	GFLOPs
Baseline (3D-ResNet18)	84.4	33.2M	19.3
Baseline (3D-ResNet34)	87.7	63.5M	36.7
Backbone CNN (3D-ResNet18)	**86.5**	**33.2M**	**19.3**
LR (3D-ResNet18)†	**88.2**	**45.73M**	**30.3**
HR (3D-ResNet18)†	89.4	48.12M	155.7
LR+HR (3D-ResNet18)†	**90.6**	**53.63M**	156.01

5 Conclusion

In this paper, we propose a context guided motion network, which integrates a novel context guided motion upsampling layer, in order to learn the semantic motion representation in a self-supervised manner. The learned motion representation is versatile and can be applied to boost the performance of various video-related tasks, *e.g.*, frame prediction and video recognition. We experimentally verified the superiority of the proposed method from various perspectives, showing the state-of-the-art performances over several popular video-related tasks.

Acknowledgement. This work was supported by National Natural Science Foundation of China (61831015, U1908210).

References

1. Abu-El-Haija, S., et al.: Youtube-8m: A large-scale video classification benchmark. arXiv (2016)
2. Bao, W., Lai, W.S., Ma, C., Zhang, X., Gao, Z., Yang, M.H.: Depth-aware video frame interpolation. In: CVPR (2019)

3. Brabandere, B.D., Jia, X., Tuytelaars, T., Gool, L.V.: Dynamic filter networks. In: NeurIPS (2016)
4. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7577. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33783-3_44
5. Byeon, W., Wang, Q., Srivastava, R.K., Koumoutsakos, P.: ContextVP: fully context-aware video prediction. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11220. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01270-0_46
6. Cai, Z., Wang, L., Peng, X., Qiao, Y.: Multi-view super vector for action recognition. In: CVPR (2014)
7. Carreira, J., Noland, E., Banki-Horvath, A., Hillier, C., Zisserman, A.: A short note about kinetics-600. arXiv (2018)
8. Carreira, J., Zisserman, A.: Quo vadis, action recognition? A new model and the kinetics dataset. In: CVPR (2017)
9. Charbonnier, P., Blanc-Feraud, L., Aubert, G., Barlaud, M.: Two deterministic half-quadratic regularization algorithms for computed imaging. In: ICIP (1994)
10. Che, Z., Borji, A., Zhai, G., Min, X., Guo, G., Le Callet, P.: How is gaze influenced by image transformations? Dataset and model. TIP **29**, 2287–2300 (2019)
11. Choutas, V., Weinzaepfel, P., Revaud, J., Schmid, C.: PoTion: pose motion representation for action recognition. In: CVPR (2018)
12. Dai, J., et al.: Deformable convolutional networks. In: ICCV (2017)
13. Diba, A., et al.: Spatio-temporal channel correlation networks for action classification. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11208. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01225-0_18
14. Diba, A., Sharma, V., Gool, L.V., Stiefelhagen, R.: DynamoNet: Dynamic action and motion network. arXiv (2019)
15. Dosovitskiy, A., et al.: FlowNet: learning optical flow with convolutional networks. In: ICCV (2015)
16. Fan, L., Huang, W., Gan, C., Ermon, S., Gong, B., Huang, J.: End-to-end learning of motion representation for video understanding. In: CVPR (2018)
17. Feichtenhofer, C., Fan, H., Malik, J., He, K.: SlowFast networks for video recognition. In: ICCV (2019)
18. Feichtenhofer, C., Pinz, A., Wildes, R.P.: Spatiotemporal residual networks for video action recognition. In: NeurIPS (2016)
19. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: CVPR (2016)
20. Fernando, B., Bilen, H., Gavves, E., Gould, S.: Self-supervised video representation learning with odd-one-out networks. In: CVPR (2017)
21. Hara, K., Kataoka, H., Satoh, Y.: Learning spatio-temporal features with 3D residual networks for action recognition. In: ICCVW (2017)
22. He, D., et al.: StNet: local and global spatial-temporal modeling for action recognition. In: AAAI (2019)
23. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
24. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: FlowNet 2.0: evolution of optical flow estimation with deep networks. In: CVPR (2017)

25. Ilg, E., Saikia, T., Keuper, M., Brox, T.: Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11216. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01258-8_38
26. Kingma, D.P., Ba, J.L.: Adam: a method for stochastic optimization. In: ICLR (2015)
27. Kroeger, T., Timofte, R., Dai, D., Van Gool, L.: Fast optical flow using dense inverse search. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_29
28. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: a large video database for human motion recognition. In: ICCV (2011)
29. Kwon, Y.H., Park, M.G.: Predicting future frames using retrospective cycle GAN. In: CVPR (2019)
30. Lee, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Unsupervised representation learning by sorting sequences. In: ICCV (2017)
31. Li, X., Hu, X., Yang, J.: Spatial group-wise enhance: Improving semantic feature learning in convolutional networks. arXiv (2019)
32. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Flow-grounded spatial-temporal video prediction from still images. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11213. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01240-3_37
33. Liang, X., Lee, L., Dai, W., Xing, E.P.: Dual motion gan for future-flow embedded video prediction. In: ICCV (2017)
34. Liu, W., Luo, W., Lian, D., Gao, S.: Future frame prediction for anomaly detection - a new baseline. In: CVPR (2018)
35. Liu, Z., Yeh, R.A., Tang, X., Liu, Y., Agarwala, A.: Video frame synthesis using deep voxel flow. In: ICCV (2017)
36. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. In: ICLR (2016)
37. Min, X., Gu, K., Zhai, G., Liu, J., Yang, X., Chen, C.W.: Blind quality assessment based on pseudo-reference image. TMM **20**, 2049–2062 (2017)
38. Min, X., Zhai, G., Gu, K., Yang, X., Guan, X.: Objective quality evaluation of dehazed images. IEEE Trans. Intell. Transp. Syst. **20**, 2879–2892 (2018)
39. Min, X., Zhai, G., Zhou, J., Zhang, X.P., Yang, X., Guan, X.: A multimodal saliency model for videos with high audio-visual correspondence. TIP **29**, 3805–3819 (2020)
40. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and learn: unsupervised learning using temporal order verification. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_32
41. Ng, J.Y.H., Choi, J., Neumann, J., Davis, L.S.: ActionFlowNet: learning motion representation for action recognition. In: WACV (2018)
42. Pan, J., et al.: Video generation from single semantic label map. In: CVPR (2019)
43. Paszke, A., et al.: Automatic differentiation in PyTorch (2017)
44. Pérez, J.S., Meinhardt-Llopis, E., Facciolo, G.: Tv-l1 optical flow estimation. Image Process. On Line **3**, 137–150 (2013)
45. Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with pseudo-3D residual networks. In: ICCV (2017)

46. Reda, F.A., et al.: SDC-Net: video prediction using spatially-displaced convolution. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_44
47. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: EpicFlow: edge-preserving interpolation of correspondences for optical flow. In: CVPR (2015)
48. Shen, W., Bao, W., Zhai, G., Chen, L., Min, X., Gao, Z.: Blurry video frame interpolation. In: CVPR (2020)
49. Shi, W., et al.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: CVPR (2016)
50. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NeurIPS (2014)
51. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv (2012)
52. Su, H., Jampani, V., Sun, D., Gallo, O., Learned-Miller, E., Kautz, J.: Pixel-adaptive convolutional neural networks. arXiv (2019)
53. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In: CVPR (2018)
54. Sun, S., Kuang, Z., Sheng, L., Ouyang, W., Zhang, W.: Optical flow guided feature: a fast and robust motion representation for video action recognition. In: CVPR (2018)
55. Tian, Y., Min, X., Zhai, G., Gao, Z.: Video-based early ASD detection via temporal pyramid networks. In: ICME (2019)
56. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: ICCV (2015)
57. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: CVPR (2018)
58. Villegas, R., Yang, J., Hong, S., Lin, X., Lee, H.: Decomposing motion and content for natural video sequence prediction. arXiv (2017)
59. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: ICCV (2013)
60. Wang, L., Qiao, Y., Tang, X.: Action recognition with trajectory-pooled deep-convolutional descriptors. In: CVPR (2015)
61. Wang, L., Xiong, Y., Wang, Z., Qiao, Y.: Towards good practices for very deep two-stream convnets. arXiv (2015)
62. Wang, L., et al.: Temporal segment networks: towards good practices for deep action recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_2
63. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. TIP **13**, 600–612 (2004)
64. Wei, D., Lim, J., Zisserman, A., Freeman, W.T.: Learning and using the arrow of time. In: CVPR (2018)
65. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: DeepFlow: large displacement optical flow with deep matching. In: ICCV (2013)
66. Xiao, H., Feng, J., Lin, G., Liu, Y., Zhang, M.: MoNet: deep motion exploitation for video object segmentation. In: CVPR (2018)
67. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: speed-accuracy trade-offs in video classification. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11219. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01267-0_19

68. Xu, D., Xiao, J., Zhao, Z., Shao, J., Xie, D., Zhuang, Y.: Self-supervised spatiotemporal learning via video clip order prediction. In: CVPR (2019)
69. Xu, J., Ni, B., Li, Z., Cheng, S., Yang, X.: Structure preserving video prediction. In: CVPR (2018)
70. Xu, L., Ren, J.S., Liu, C., Jia, J.: Deep convolutional neural network for image deconvolution. In: NeurIPS (2014)
71. Xu, X., Cheong, L.F., Li, Z.: Motion segmentation by exploiting complementary geometric models. In: CVPR (2018)
72. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv (2015)
73. Zhai, G., Min, X.: Perceptual image quality assessment: a survey. Sci. China Inf. Sci. **63**, 211301 (2020). https://doi.org/10.1007/s11432-019-2757-1