Dear Editors and Reviewers,

The authors would like to thank the reviewers for their valuable comments and constructive suggestions. We have tried to address all the comments on the revised manuscript. The reviewers' comments are rendered in blue whereas the authors' responses are marked in black. In the revised draft, the revised contents are marked in purple. The summary of the major modifications and added contents is as follows:

- In Section 5 line xxx, according to the comments of all three reviewers, we add an experiment on the large-scale dataset Kinetics for video action recognition. Results in Table 5 show that our method incorporated in TSM Lin et al. (2019) can surpass all state-of-the-art methods listed in the table. Despite this manuscript is submitted in the year of 2019, our method could be even better than many other state-of-the-art methods published in 2020.
- In Section 5, line xxx, we specify the number of segments while training on the UCF-101 and HMDB-51.
- In Section 6, Table 7, we update the result without applying SE block into the feature refinement subnetworks for fair comparison. Results show that the SE block does not lead to significant improvement for video object detection, which further validates the effectiveness of our method.
- We also corrected all the typos and ambiguous descriptions accordingly based on the comments and suggestions of all reviewers.

To Reviewer #1:

Q1: In addition to new experiments and analysis on new tasks, it seems that the technical improvement over conference version is limited. More technical improvement is expected to be performed for a journal extension.

A: Compared to the conference version, the journal version further achieves remarkable improvement on other high-level vision tasks (*e.g.* video object detection) and low-level vision tasks (*e.g.* video compression artifact removal). We emphasize that these tasks are also as challenging as the video action recognition. Obtaining such improvement on both high-level and low-level tasks also verify the generalization capacity of our method. The effectiveness of our method is also recognized by Reviewer #3. Apart from these extensive experiments on other tasks, in our revised paper, we further demonstrate the effectiveness of our OFF module on another large-scale video dataset Kinetics-400 for video action recognition.

Q2: I recommend the authors to report results on much larger datasets such as Kinetics and compare with more recent SOTA, such as non-local nets, slowfast etc.

A: In the revised paper, we performed experiments on Kinetics-400 dataset, which contains about 240k videos for training and 20k videos for validation. Videos in this dataset are categorized into 400 classes. The capacity of Kinetics-400 is way larger than the UCF-101 and HMDB-51. We select another 2D CNN based method TSM (Lin et al. 2019) instead of TSN (Wang et al. 2016) as our baseline method on Kinetics. As shown in Table 5, all the state-of-the-art has been surpassed with the help of OFF. Note that this manuscript is submitted in the year of 2019, but by incorporating our module into TSM, our method could be even better than many other state-of-the-art methods published in 2020.

Q3: It is unclear how many segments are used for training. It seems the training and testing segment is different, but this is OK TSN framework as it only uses a late fusion. However, for OFF network, it involves the early feature interaction in OFF subnetwork. If the training and testing segment number is different, it may cause the mis-alignment problem during training and testing.

Thanks for the reminder. We have added this technical details into Section 5.1. We use 3 segments for training for all ablation studies on UCF-101 and HMDB-51. As for the final configuration, we use 7 segments instead of 3 for better performance. We avoid the misalignment between the training and testing by fixing the time interval of every two segments between the training and testing. This strategy is described in Section 4.2.3.

To Reviewer #2:

Q1: The OFF method itself suggests optical flow but is only spatio-temporal gradient computation in TSN networks; therefore, in my opinion it is misleading as optical flow is not estimated, nor do any experiments suggest that optical flow computation is performed within the network.

A: For clarity, the proposed OFF is theoretically guided by the definition of optical flow, but is not designed for estimating the optical flow. Specifically, inspired by the derivation of brightness constancy equation from optical flow, we further generalize the optical flow representation from image-level to feature-level, and design the OFF to extract feature-wise motion information. Details of the derivation are illustrated at the beginning of Section 3.

Q2: I would expect for the journal version 2 years later at least to have experimental validation on a newer dataset (e.g. Kinetics, or if compute is a large factor Something-Something or Charades), and optionally on an updated network model that at least uses 3D convolution (e.g. I3D), as TSN is no longer representative of state-of-the-art video approaches.

A: Please check the answer of Q2 raised by Reviewer #1.

Q3: the new extra task for video object detection is approached with bells and whistles like SE blocks and, therefore, cannot assess the contribution (OFF) for this task; I would recommend to remove these bells-and-whistles for a fair comparison to the baseline R-FCN detector in FGFA.

A: In our experiments, SE blocks are only added to the residual blocks within the feature refining sub-networks followed by the ResNet-101 backbone. We did not apply any bells-and-whistles in the original backbone. The feature refinement network is also part of our OFF module, so the comparison is fair. Besides, shown in Table 7, the SE module within the feature refinement sub-networks will only lead to about 0.27% gain in terms of mAP. Compared with the gain of the entire module (2.11%), the gain introduced by the SE block does not affect us getting into the final conclusion.

To Reviewer #3:

Q1: since this paper claims to be SoTA on action recognition, it is necessary to have some experiments on Kinetics-400 or Something-Something-v1 and compared with current methods, since most of recent work on action recognition report result on either these two benchmarks and do not report result on UCF101 and HMDB51, it is hard to compare OFF with the latest methods.

A: Please check the answer of Q2 raised by Reviewer #1.

Q2 Typos and comments.

(1) line 39-40, page 2, second column: ”over 200 frames per second” $->$ on GPU?

A: Corrected. This is evaluated on GPU.

(2) line 49-50, page 2, second 9-40lumn: ”state-of-the-art among RGB-based on recognition methods” $->$ is this on UCF101 and HMDB51 only? Note that I3D get better results with Kinetics-pretraining?

A: Yes previously this is for UCF-101 and HMDB-51 only. But after the revision we also surpass the performance of I3D on Kinetics. Now it is fair to say that we are state-of-the-art on all these datasets.

(3) line 4-5, page3, first column, ”xxx%” needs to be corrected. page 7, first column, line 39, ”While as” grammar?

A: Thanks for the justification. Corrected.

(4) Section 5.2, Table 1 present all splits, then table 2, 3, 4 on only split 1, then Table 5 on all split which is confused, why don't either 1) do 2,3,4 on all splits or 2) move Table 1 to the end, right just before Table 5, and make it Table 4. Also Page 9 (second column, line 44) and Page 10 (first column, line 31) wrongly referred to Table 4 and 3 (swapping them).

A: Thanks for the justification. We have corrected all typos here. As for the setting of cross validation on different splits, we argue that no matter what strategy we choose for evaluation (over 3 splits or just on split 1), the conclusion is still the same as the gain on each split is quite close. Full cross validation is conducted here for Table 1 and 5 just for fair comparison with other methods as they did so. Table 2, 3, 4 are only responsible for ablation studies, so there is no need to run over 3 splits.

(5) Table 5 I3D results w/o pretraining if it is on split1 only, then it needs to be marked with caption to differentiable from the others.

A: Corrected. Comparison on Kinetics is also added in Table 5.

International Journal of Computer Vision manuscript No.
(will be inserted by the editor)

Optical Flow Guided Feature: A Fast and Robust Motion Representation for Video Analysis

Shuyang Sun · Yi Zhou · Peiqin Zhuang · Lu Sheng · Zhanghui Kuang · Wei Zhang · Wanli Ouyang · Philip H. S. Torr

Received: date / Accepted: date

Abstract Motion representations play a vital role in video analyses. In this study, we introduce a novel Optical Flow Guided Features (OFF) for the network to compactly distill temporal information of a video via a fast and robust way. The OFF features are inspired by the derivation of optical flow constrained by brightness constancy, acting as the spatio-temporal gradients of deeply learned appearance features, which lie in the orthogonal space against that of the optical flows. The OFF features can be embedded in any existing video-oriented CNN models with only a slight budget of costs. The simple but powerful OFF features are proven effective for different vision tasks, from high-level tasks such as video action recognition, video object detection, to low-level tasks such as video compression artifact removal. For example, video action recognition models equipped by OFF module, even just fed by a RGB stream, achieves a competitive accuracy of 93.3% on the well-known UCF-101 dataset as the previous state-of-the-art with two streams (RGB and optical flow), but is 15 times faster in speed. Experimental results also show that OFF is complementary to other motion modalities such as optical flow, and is effective for multiple video-related tasks. When the proposed method is plugged into the state-of-the-art video action recognition framework, it has 96.0% and 74.2% accuracy on UCF-101 and HMDB-51 respectively. We also show that our method could surpass the state-of-the-art methods on another large-scale video dataset Kinetics-400. As for video object detection, the OFF module can achieve 2.1% (mAP) gain on the ImageNet VID dataset compared to the baseline model, and the the OFF module can improve the baseline video artifact removal model by 0.8dB (PSNR) on the Vimeo-90K dataset. The code for this project is available at: https://github.com/kevin-ssy/Optical-Flow-Guided-Feature.

Keywords motion representation · video analysis · optical flow

1 Introduction

Video analysis has received longstanding attentions in the community of computer vision. It is required to describe visual dynamics in addition to the spatial appearances, and inferring video-based perceptual tasks with the combinational interpretation of spatio-temporal semantics. Since convolutional neural networks (CNNs) have achieved great successes in image classification (Krizhevsky et al. 2012; Simonyan and Zisserman 2014b; Szegedy et al. 2015; He et al. 2016a; Zeng et al. 2017; Zhao et al. 2017; Ouyang et al. 2016), lots of CNN-based methods have been proposed to video analysis, such as video action recognition (Carreira and Zisserman 2017; Wang et al. 2016; Ng et al. 2016; Zhang et al. 2016; Feichtenhofer et al. 2016; Diba et al. 2016b,a; Wang et al. 2015a,b; Sun et al. 2015; Simonyan and Zisserman 2014a) and video object detection (Dai et al. 2016; Kang et al. 2017a, 2016). Compared to their image-based counterparts, temporal information extraction is the key ingredient.

Shuyang Sun · Philip H. S. Torr
University of Oxford, Oxford, United Kingdom

Yi Zhou · Peiqin Zhuang · Wanli Ouyang
University of Sydney, Darlington, NSW, Australia

Lu Sheng
Beihang University, Beijing, China

Zhanghui Kuang · Wei Zhang
SenseTime Research, Hong Kong SAR, China

Shuyang Sun and Yi Zhou contribute equally to this paper.

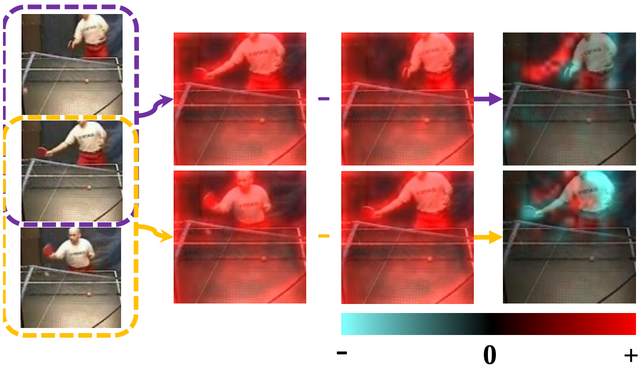

Fig. 1 **The Optical Flow guided Feature (OFF).** Left column: input frames. Middle two columns: standard deep features before applying OFF onto two frames. Right column: temporal difference in OFF. The colors red and cyan are used respectively for positive and negative values. The feature differences between two frames are valid and comprehensive in representing motion information. Best viewed in color and zoomed in.

Optical flow is found to be a useful motion representation in various video tasks, including video action recognition (Simonyan and Zisserman 2014a; Wang et al. 2016; Carreira and Zisserman 2017) and video synthesis (Pan et al. 2019; Li et al. 2018; Wang et al. 2018a). However, extracting dense optical flows is still inefficient, *i.e.*, it costs over 90% of the overall execution time in a two-stream (a.k.a. RGB and optical flow streams) based pipeline, both at the training and testing phases. Another set of category for motion modeling explicitly describe the video dynamics as 3DHOG (Klaser et al. 2008), improved Dense Trajectory (Wang and Schmid 2013), and motion vector (Zhang et al. 2016). However, they are either inefficient or not so effective as optical flow. Recent advances also use the 3D convolutional neural networks (3D-CNNs) for a flexible temporal modeling from the input RGB data, but it is still hard for 3D-CNNs to extract comprehensive temporal dynamics if just straightforwardly feeding RGB data into the network. To solve this problem, some motion-aware components were proposed to explicitly extract cross-frame motion information (Wang et al. 2018b; Chen et al. 2019).

How to design a good motion representation that is both fast and robust? To this end, the required computation should be economical and the representation should sufficiently cover the motion information. Taking the above requirements into consideration, we propose the Optical Flow guided Feature (OFF), which is fast to compute and can comprehensively represent motion dynamics in a video clip. Moreover, this module can also be instantaneously plugged into various video learning CNN models.

In this paper, we define the proposed Optical Flow guided Feature (OFF) as a new feature representation from the orthogonal space of the optical flow defined on the feature level (Horn, Berthold K.P.; Schunck 1981). This particular feature consists of 1) spatial gradients of the input feature maps in horizontal and vertical directions, and 2) temporal gradients obtained from the difference between feature maps from different frames. Since all the operations in OFF are differentiable, the whole process can be end-to-end trained when OFF is plugged into one CNN architecture. The OFF unit only consists of fast pixel-wise operators on CNN features and enables the network with RGB input to capture spatial and temporal information simultaneously.

One vital component in OFF is the difference between features from the images/segments from different time stamps. As shown in Fig. 1, the difference between the features from two images provides representative motion information that can be conveniently employed by CNNs. The negative values in the difference image depict the locations where the body parts/objects disappear, while the positive values represent where they emerge. This pattern of disappearing at one location and emerging at another location can be easily treated as a specific motion pattern and captured by later CNN layers. The temporal difference could be further combined with the spatial gradients such that the constituted OFF is guided by the optical flow on feature level according to our derivation in later section. Moreover, calculation of the motion dynamics at the feature level is faster and also more robust because 1) it enables the spatial and temporal networks with the capability of weight sharing, and 2) deeply learned features convey more semantic and discriminative representations with reliable elimination of local and background noises in the raw frames.

Our work has two main contributions.

First, **OFF is a fast and robust video motion representation.** OFF is fast to enable over 200 frames per second and is derived from and guided by the optical flow. Taking only the RGB stream from videos, experimental results show that the CNN-based model with OFF is close in performance when compared with the state-of-the-art two-stream algorithms (Carreira and Zisserman 2017). The models with OFF can achieve an accuracy of 93.3% on the UCF-101 dataset (Soomro et al. 2012) with *only RGB* data as the input, which is currently state-of-the-art among the RGB-based action recognition methods. When plugging OFF in the video action recognition method (Wang et al. 2016) in a two-stream manner (RGB + Optical Flow), the performance of our algorithm could result in an accuracy of 96.0% on the UCF-101 dataset and 74.2% on the

HMDB-51 dataset (Kuehne et al. 2011). As an extension, we show that our method can also be generalized onto another large-scale video dataset Kinetics-400 and achieve state-of-the-art results. Apart from the success achieved on the the video action recognition, the OFF is also applicable on other video-related tasks. For the task of video object detection, the network plugged with OFF could achieve 2.1% (mAP) gain compared to the baseline on the benchmark of ImageNet VID (Russakovsky et al. 2015), while keeps running by over 5 times faster. Our OFF is also effective on low-level vision tasks like video artifact removal. By incorporating the OFF together with the backbone DnCNN (Zhang et al. 2017), the new framework could achieve 0.8dB in terms of PSNR gain on the Vimeo-90K dataset (Xue et al. 2017).

Second, **the OFF equipped convolutional neural networks can be trained in an end-to-end fashion.** In this way, the spatial and motion representations can be jointly learned through a single network. This property is friendly for video tasks on large-scale datasets, as it avoids pre-computing and storing the motion modalities (*e.g.*, optical flow) for training. Besides, the OFF can be used between images/segments in a video clip both on image level and feature level.

This paper extends our conference paper (Sun et al. 2018) in three aspects. First, we conduct additional component analyses to investigate the effectiveness of the spatial and temporal components of the OFF, respectively. Second, we extend the OFF onto other high-level vision tasks like the video object detection, achieving remarkable performance with only tiny extra computational costs. Third, we also apply OFF onto low-level video tasks like the video artifact removal. Together with the theoretical analysis, our experimental results reveal that the OFF module could also benefit the low-level tasks.

The rest of this paper is organized as follows. Section 2 introduces recent methods that are related to our work. Section 3 illustrates the definition of the OFF module and the details our proposed method. Section 4 explains our implementation in CNN based models. Our experimental results and ablation studies for video action recognition are summarized in the section 5. Further explorations about the video object detection and video artifact removal are illustrated in the Section 6 and Section 7 respectively. The conclusion is remarked in the Section 8.

2 Related Work

Traditional methods extracted hand-craft local visual features such as 3D HOG (Klaser et al. 2008), Motion Boundary Histograms (MBH) (Dalal et al. 2006), improved Dense Trajectory (iDT) (Wang and Schmid 2013; Wang et al. 2011) and then encoded them into sparse or compact feature vectors which were fed into classifiers (Scovanner et al. 2007; Peng et al. 2016). Deeply learned features were then found to perform better than hand-crafted features for action recognition (Simonyan and Zisserman 2014a; Wang et al. 2015a).

As a significant breakthrough in action recognition, two-Stream based frameworks used the deep convolutional neural networks to learn from the hand-craft motion features like optical flow and iDT (Simonyan and Zisserman 2014a; Wang et al. 2015a; Zhang et al. 2016; Wang et al. 2016; Diba et al. 2016a; Yue-Hei Ng et al. 2015; Carreira and Zisserman 2017; Tran et al. 2015; Feichtenhofer et al. 2016, 2017a). These attempts have achieved remarkable progress in improving the recognition accuracy, but still rely on the pre-computed optical flow or iDT, which tremendously hamper the efficiency of the whole framework.

In order to obtain the motion modality in a fast way, recent works used optical flow at the training stage only (Ng et al. 2016), or proposed motion vector as the simplified version of optical flow (Zhang et al. 2016). These methods have produced degraded optical flow results and still did not perform on par with the approaches using traditional optical flow as the input.

Many approaches learn to capture the motion information directly from input frames using 3D CNNs (Tran et al. 2015; Varol et al. 2016; Carreira and Zisserman 2017; Tran et al. 2017; Diba et al. 2016a; Varol et al. 2017). Boosted by the temporal convolution and pooling operations, 3D CNN could distill the temporal information between consecutive frames without segmenting them into short snippets. Compared with the learning of filters to capture motion information, our OFF is a representation mathematically derived from the optical flow. 3D CNNs that are constrained by network design, training sample and parameter regularization like weight decay, may not be able to learn good motion representation such as those in OFF. By incorporating the 3D CNNs together with some temporal-aware modules, *e.g.* the Non-Local Module (Wang et al. 2018b), GloRe (Chen et al. 2019) and *etc.*, the network itself could be empowered to better reason the frame-wise correlation. These operators, which may have similar motivations as the proposed OFF module, could be applied complementarily with the OFF module. However, without these modules, it may be hard for the 3D CNNs to extract similar temporal information that is comparable with those obtained from the optical flows.

Therefore, current state-of-the-art 3D CNN based algorithms still rely on traditional optical flow to help

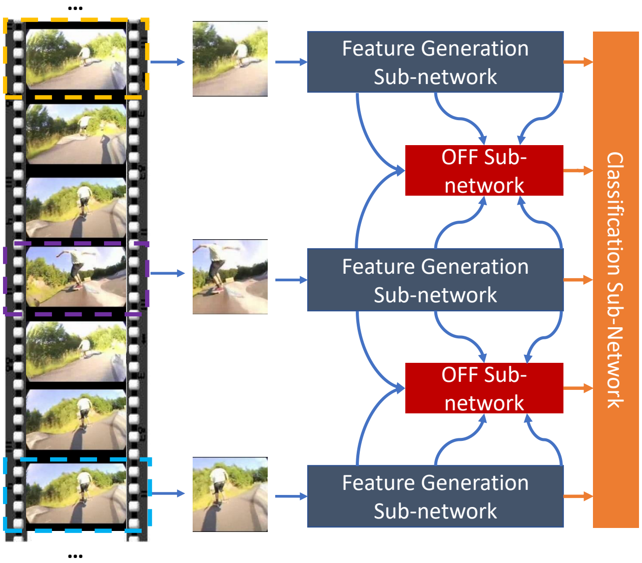
...
Feature Generation Sub-network
OFF Sub-network
Feature Generation Sub-network
OFF Sub-network
Feature Generation Sub-network
Classification Sub-Network
...

Fig. 2 Network architecture overview. The feature generation sub-network extracts feature for each frame sampled from the video. Based on the features from two adjacent frames extracted by the feature generation sub-networks, an OFF sub-network is applied to generate the OFF for further classification. The scores from all sub-networks are fused to get the final result. Best viewed in color.

the networks to capture motion patterns. In comparison, our OFF 1) well captures the motion patterns so that RGB stream with OFF performs on par with two-stream based methods, and 2) is also complementary to other motion representations like optical flow.

To capture long-term temporal information from videos, one intuitive approach is to introduce the Long Short-Term Memory (LSTM) module as an encoder to extract the relationship between the sequence-illustrating deep features (Yue-Hei Ng et al. 2015; Sun et al. 2017; Shi et al. 2017). LSTM can still be applied on the OFF without elaborative network modification. Therefore, our OFF is complementary to these methods.

Concurrent with our work, another state-of-the-art method applies a strategy called *ranked pool* (Fernando et al. 2017) that generates a fast video-level descriptor, namely, the *dynamic images* (Bilen et al. 2016). However, the very nature in design and implementation between the dynamic images and ours are different. The dynamic images are designed to summarize a series of frames while our method is designed to capture the motion information related to optical flow.

3 Optical Flow Guided Feature

Our proposed Optical Flow guided Feature (OFF) is inspired by the famous brightness constant constraint that is widely used in traditional optical flow estimation Horn, Berthold K.P.; Schunck (1981). It is formulated as follows:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t), \tag{1}$$

where $I(x, y, t)$ denotes the pixel at the location (x, y) of a frame at time t. For frames t and $t + \Delta t$, Δx and Δy are the spatial pixel displacements along the x and y axes. It assumes that for any point that moves from (x, y) at frame t to $(x+\Delta x, y+\Delta y)$ at frame $t+\Delta t$, its brightness keeps unchanged over time. When we apply this constraint at the feature level, we have

$$f(I; w)(x, y, t) = f(I; w)(x + \Delta x, y + \Delta y, t + \Delta t), \tag{2}$$

where f is a mapping function for extracting features from the image I. w denotes the parameters in the mapping function. The mapping function f can be any differentiable function. In this paper, we employ trainable CNNs consisted of stacks of convolution, ReLU, and pooling operations, note that we do not apply the Batch Normalization (BN) (Ioffe and Szegedy 2015a) in our network to avoid the overfitting problem. According to the definition of optical flow, we assume that $p = (x, y, t)$ and obtain the equation as follows:

$$\frac{\partial f(I; w)(p)}{\partial x}\Delta x + \frac{\partial f(I; w)(p)}{\partial y}\Delta y + \frac{\partial f(I; w)(p)}{\partial t}\Delta t = 0. \tag{3}$$

By dividing Δt in both sides of Equation (3), we obtain

$$\frac{\partial f(I; w)(p)}{\partial x}v_x + \frac{\partial f(I; w)(p)}{\partial y}v_y + \frac{\partial f(I; w)(p)}{\partial t} = 0, \tag{4}$$

where $[v_x, v_y]$ denotes the two dimensional velocities of feature point at p along the horizontal and vertical directions. $\frac{\partial f(I;w)(p)}{\partial x}$ and $\frac{\partial f(I;w)(p)}{\partial y}$ are the spatial gradients of $\partial f(I; w)(p)$ in x and y axes respectively. $\frac{\partial f(I;w)}{\partial t}$ is the temporal gradient along time axis.

As a special case, when $f(I; w)(p) = I(p)$, then the $f(I; w)(p)$ simply represents a single pixel at location p, and (v_x, v_y) are called optical flow. Optical flow is obtained by solving an optimization problem with the constraint in Equation (4) for each p (Barron et al. 1994; Brox et al. 2004; Bigun et al. 1991). Here in this case, the term $\frac{\partial f(I;w)(p)}{\partial t}$ represents the difference between RGB frames. Previous works have shown that the temporal difference between frames is useful in video related tasks (Wang et al. 2016), however, there is no theoretical evidence to help explain why this simple idea

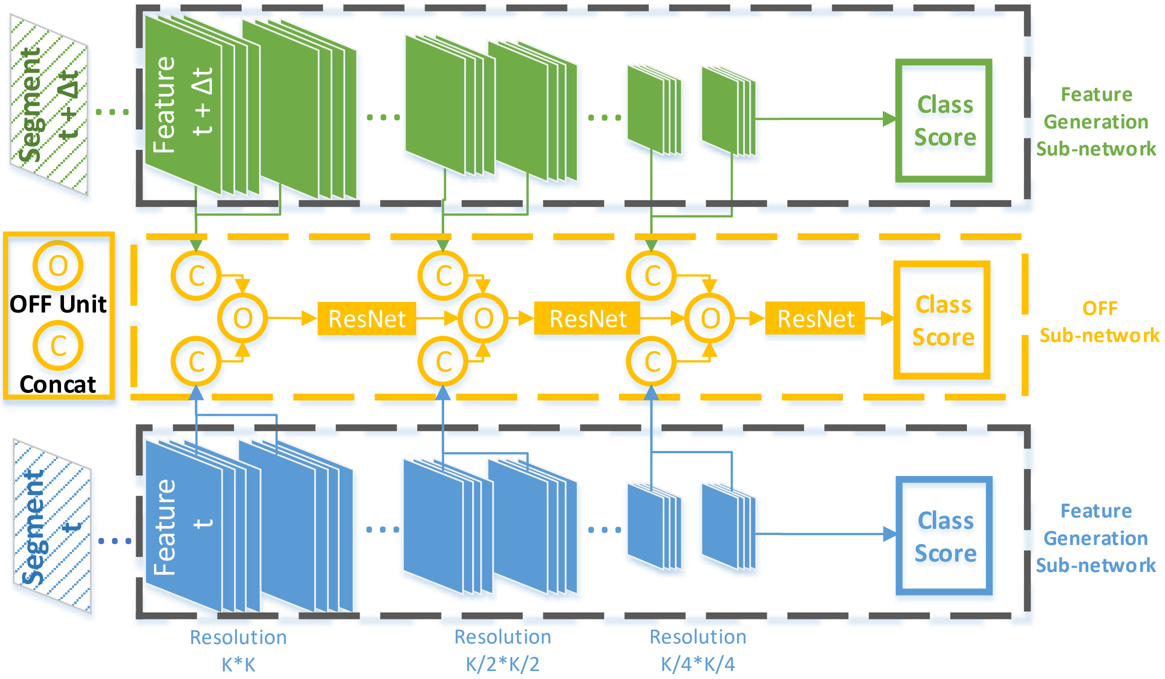

Fig. 3 Network architecture for two segments in video action recognition. The inputs are two segments in blue and green colors that are separately fed into the feature generation sub-network to obtain basic features. In our experiment, the backbone for each feature generation sub-network is the BN-Inception (Szegedy et al. 2015). Here K represents the largest side length of the square feature map selected to undergo the OFF sub-network for obtaining the OFF features. The OFF sub-network consists of several OFF units, and several residual blocks (He et al. 2016a) are connected between OFF units from different levels of resolution. These residual blocks constitute a ResNet-20 when seen as a whole. The scores obtained by different sub-networks are supervised independently. Detailed structure of the OFF unit is shown in Figure 4. Best viewed in color.

works that well. Here, we can find its correlation to spatial features and optical flow.

We generalize the representation of optical flow from pixel $I(p)$ to feature $f(I;w)(p)$. In this general case, $[v_x, v_y]$ are called the feature flows. As shown in Equation (4), $\mathbf{F}(I;w)(p) = \left[\frac{\partial f(I;w)(p)}{\partial x}, \frac{\partial f(I;w)(p)}{\partial y}, \frac{\partial f(I;w)(p)}{\partial t}\right]$ is orthogonal to the vector $[v_x, v_y, 1]$ about the feature-level optical flow. $\mathbf{F}(I;w)(p)$ changes as the feature-level optical flow changes. Therefore, $\mathbf{F}(I;w)(p)$ is guided by the feature-level optical flow. We call $\mathbf{F}(I;w)(p)$ as *Optical Flow guided Feature (OFF)*. The OFF $\mathbf{F}(I;w)(p)$ encodes the spatial-temporal information orthogonally and complementary to the optical flow (v_x, v_y) in the feature level.

4 Using OFF for Video Action Recognition

In this section, the detailed implementation of the Optical Flow guided Feature (OFF) and its usage for action recognition are introduced.

4.1 Network Architecture

4.1.1 Network Architecture Overview

Figure 2 shows an overview of the whole network architecture. The network consists of three sub-networks for different purposes: 1) feature generation sub-network, 2) OFF sub-network and 3) classification sub-network. The feature generation sub-network generates basic features using shared CNN backbone structures. In the OFF sub-network, the OFF features are extracted using the features from the feature generation sub-network, and then several residual blocks are stacked for obtaining the refined features. The features from the previous two sub-networks are then used by the classification sub-network for obtaining the action recognition results. Figure 3 demonstrates the detailed network structures with two input video segments. As shown in Figure 3, we extract features from multiple layers on a specific level with the same resolution by concatenating them together and feed them into one OFF unit. The whole network has 3 OFF units with different scales. The details about the structure of each sub-network is discussed as follows.

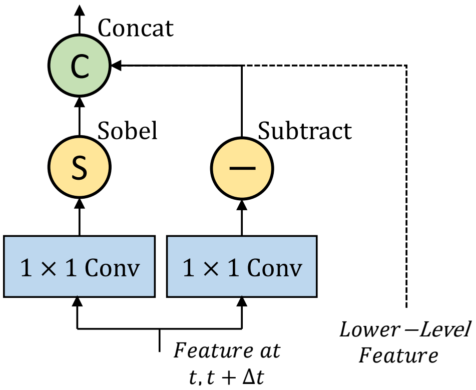

Fig. 4 Detailed architecture of OFF unit. Feeding the concatenated features as the inputs of a 1x1 convolution layer, the dimension of the features are reduced for further optimization. A Sobel operator and an element-wise subtraction operation are applied on the reduced features to calculate the spatial and temporal gradients respectively. Concatenated with the low-level features from previous layers, the gradients together with the features will be then fed into its following layers. The combination of gradients constitutes the OFF, and the sobel operator, subtracting operator and the 1 × 1 convolution layers before them constitute a OFF unit.

4.1.2 Feature Generation Sub-network

The basic features $f(I)$[1] are extracted from the input image using several 2D convolutional layers with the Rectified Linear Unit (ReLU) as the activation function and max-pooling for down-sampling and feature aggregation. We select BN-Inception (Szegedy et al. 2015) as the network structure to extract these base feature maps. The feature generation sub-network can be replaced by any other network architecture that are compatible to the video learning tasks.

4.1.3 OFF Sub-network

The OFF sub-network consists of several OFF units. Different units use basic features $f(I)$ from different depths. As shown in Figure 4, an OFF unit contains an OFF layer to generate the optical flow guided features. Each OFF layer contains a 1 × 1 convolutional layer to align each piece of feature, and a set of operators for OFF generation including Sobel operation and element-wise subtraction. After the OFF is obtained, the OFF unit will concatenate with the features from the lower level, then the combined features will be output to the following residual blocks.

OFF Layer. The OFF layer is responsible for generating the OFF from the basic features $f(I)$. Figure 4 shows the detailed implementation the OFF layer. According to Equation (3), the OFF should consist of both spatial and temporal gradient of the feature. Denote $f(I, c)$ as the c^{th} channel of the basic feature $f(I)$. Denote $\mathcal{F}_x$ and $\mathcal{F}_y$ as the gradients along the x and y directions respectively, which correspond to spatial gradients within the OFF. We apply the Sobel operator for spatial gradient generation as follows:

$$\mathcal{F}_x = \left\{ \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} * f(I,c) \middle| \, c = 0 \ldots, N_c - 1 \right\}, \quad (5)$$

$$\mathcal{F}_y = \left\{ \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} * f(I,c) \middle| \, c = 0 \ldots, N_c - 1 \right\}, \quad (6)$$

where $*$ denotes a 2D convolution operation along the spatial domain, and the constant N_c indicates the number of channels of the feature $f(I)$. Denote $\mathcal{F}_t$ as the OFF for gradients at the temporal directions. Temporal gradient is obtained by element-wise subtraction as follows:

$$\mathcal{F}_t = \{f_t(I,c) - f_{t-\Delta t}(I,c) \mid c = 0, \ldots, N_c - 1\}. \quad (7)$$

We concatenate the aforementioned features $\mathcal{F}_x$, $\mathcal{F}_y$, and $\mathcal{F}_t$ obtained above together with the features from the lower level, as the output of the OFF layer. We use a 1 × 1 convolutional layer before the Sobel and subtraction operations to reduce the number of channels. In our experiments, the channel dimension is reduced to 128 regardless of how many the input channels are. Then the channel-reduced features are fed into the OFF unit to calculate the OFF.

Multi-level Feature Refinement. The OFF unit can be applied for CNN layers on different levels. The inputs of one OFF unit include the basic deep features from two segments, the features from the previous OFF unit on the lower feature level (if exist). In this way, the OFF at the previous semantic level can be used for refining the OFF at the current semantic level. After the OFF of the current level is obtained, to refine the obtained features, several residual blocks as designed by He et al. (2016a) are employed after the concatenation between the OFF units at adjacent levels of resolutions. The dimensionality of OFF is further reduced in the residual block aligned to the OFF unit, for saving computation and the number of parameters. The residual blocks after multi-level concatenation of OFF features follows a ResNet-20 network structure. Note that there is no Batch Normalization (Ioffe and Szegedy 2015b) operation applied in our residual network in order to avoid the over-fitting problem.

[1] They are equivalent to the representation $f(I; w)$ depicted in Section 3.

4.1.4 Classification Sub-network

The classification sub-network takes features from different sources and uses multiple inner-product classifiers to obtain multiple classification scores. The classification scores of all sampled frames are then combined by averaging each feature generation sub-network, or OFF sub-network. The OFF at a semantic level can be used to produce an auxiliary classification score at the training stage, which is learned using its corresponding loss. Such strategy has been proved to be useful in many tasks (Szegedy et al. 2015; Wei et al. 2016; Newell et al. 2016). In the testing phase, scores from different sub-networks could be assembled for better performance.

4.2 Network Training

Note that in this section, the strategies we introduced are only applied onto the task of video action recognition. All training and testing strategies mentioned here could not be directly transferred onto other tasks if not specified.

Action recognition is treated as a multi-class classification problem. Followed by the settings in TSN (Wang et al. 2016), as there are multiple classification scores produced by each segment, we need to fuse all separated scores to generate a video-level score for loss calculation. Here, for the OFF sub-networks, the features produced by the output of OFF sub-network for the t^{th} segment on level l is denoted by $\mathcal{F}_{t,l}$. The classification score for segment t on the level l using $\mathcal{F}_{t,l}$ is denoted by $\mathbf{G}_{t,l}$. The aggregated video-level score at level l is denoted by G_l, which is obtained by:

$$G_l = \mathcal{G}(\mathbf{G}_{0,l}, \dots, \mathbf{G}_{1,l}, \dots, \mathbf{G}_{N_t-1,l}), \tag{8}$$

where N_t denotes the number of frames for extracting features. The aggregation function denoted by $\mathcal{G}$ is used for summarizing the scores predicted from different segments along time. Following the investigations in TSN, $\mathcal{G}$ is implemented as the average pooling (Wang et al. 2016). As for the feature generation sub-network, the above equation is also applicable. As we do not need intermediate supervisions for feature generation sub-network, the feature $\mathcal{F}_{t,l}$ at level l for segment t is simply used as the final feature output of the sub-network.

To update the parameters of the whole network, the loss is set to be the standard categorical cross-entropy loss. As the sub-network for each feature level is supervised independently, a loss function is used for each level as:

$$\mathcal{L}_l(y, G_l) = -\sum_{c=1}^{C} y_c \left(G_{l,c} - \log \sum_{j=1}^{C} \exp(G_{l,j}) \right), \tag{9}$$

where C is the number of action categories, $G_{l,c}$ is the estimated score for class c from the features at level l, and y_c represents the ground-truth class label. By using this loss function we can optimize the network parameters through back-propagation. Detailed implementation of training is described as follows.

4.2.1 Two-stage Training Strategy

Training of the whole network consists of two stages. The first stage indeed is to apply existing approaches, *e.g.*, TSN (Wang et al. 2016), to train the feature generation sub-network. At the second stage, we train the OFF and classification sub-network with all the weights in feature generation sub-network fixed. The weights of the OFF sub-network and the classification sub-network are learned from scratch. The whole network could be further fine-tuned in an end-to-end manner, however, we do not find significant gain by this manner. To simplify the training process, we only train the network using the first two stages.

4.2.2 Intermediate Supervision During Training

Intermediate supervision has been proven to be practical training strategy in many other computer vision tasks (Newell et al. 2016; Wei et al. 2016; Yang et al. 2017; Ouyang et al. 2017; Chu et al. 2017). As the OFF sub-networks are fed by intermediate inputs, here we add the intermediate supervision on each level to get better OFFs on each level of resolution.

4.2.3 Reducing the Memory Cost

As our framework includes several sub-networks, it costs more memory than the original TSN framework, which extracts and stores motion frames before training CNNs, and trains several networks independently. In order to reduce the computational and memorial costs, we sample less frames in the training phase than in the testing phase, and still obtain satisfactory results.

However, the time duration between segments may be varied if we sample different number of segments between training and testing. According to our definition in equation (3), only when the denotation Δt is a fixed constant, the equation (4) could be derived from the equation (3). If we sample different frames between training and testing, the time interval Δt may be inconsistent, which makes our definition to be invalid and harms the final performance. In order to keep time interval consistent between the training and testing phases, we need to carefully design the sampling scheme. We sample frames from a video as follows:

Let α be the number of frames sampled for training, and β be the number for testing. In the training phase, a video with length $L, L \leq \beta$ will be divided into β segments. Each segment has length $\lfloor L/\beta \rfloor$. We randomly select p from $\{0, 1, \ldots, L-1-(\alpha-1) * \lfloor L/\beta \rfloor\}$, where p is treated as a frame seed. Then the whole training set is constructed as $\{p, p+\lfloor L/\beta \rfloor, \ldots, p+(\alpha-1) * \lfloor L/\beta \rfloor\}$, which has interval $\lfloor L/\beta \rfloor$. In testing phase, we sample the images using the same interval $\lfloor L/\beta \rfloor$ as that in the training phase.

4.3 Network Testing

As there are multiple classification scores produced by different sub-networks, we need to fuse them all for better performance. In this study, we assemble scores from the feature generation sub-network and the last level of OFF sub-network by a simple summation operation. We test our model based on a state-of-the-art framework TSN (Wang et al. 2016). The testing setting under the TSN framework is illustrated as follows:

4.3.1 Testing following the TSN strategy

In the testing stage of TSN (Wang et al. 2016), 25 segments are sampled from different modalities, *i.e.*, RGB, RGB difference, and optical flow. However, the number of frames in each segment is different among these modalities. We use the original settings adopted by TSN to sample $1, 5, 5$ frames per segment for RGB, RGB difference, and optical flow respectively. The input of our network is 25 segments, where the t^{th} segment is treated as the frame t as shown in Figure 3. In this case, the features extracted by individual branches of our feature generation sub-network are for a segment instead of a frame when using TSN. Other settings are kept to be the same as those in TSN.

5 Experiments on Action Recognition.

In this section, datasets and implementation details used in experiments for video action recognition will be first introduced. Then we will explore the OFF and compare it with other modalities under current state-of-the-art frameworks. Moreover, as our method can be extended to other modalities such as RGB difference and optical flow, we will show how such a simple operation could improve the performance for input with different modalities. Finally, we will discuss the meaning and difference between the OFF and other motion modalities such as optical flow and RGB difference.

5.1 Datasets and Implementation Details

Evaluation Datasets. The experimental results are evaluated on two popular video action datasets, UCF-101 (Soomro et al. 2012) and HMDB-51 (Kuehne et al. 2011). The UCF-101 dataset has 13320 videos and is divided into 101 classes, while the HMDB-51 contains 6766 videos and 51 classes. Our experiments follow the officially offered scheme which divides a dataset into 3 training and testing splits and finally calculating the average accuracy over all 3 splits. We prepare the optical flow between frames before training by directly using the OpenCV implemented algorithm (Zach et al. 2007).

Implementation Details. We train our model with 4 Nvidia TITAN X GPU, under the implementation on Caffe (Jia et al. 2014) and OpenMPI. We first train the feature generation sub-networks using the same strategy provided in the corresponding method (Wang et al. 2016). Then at the second stage, we train the OFF sub-networks from scratch with all parameters in the feature generation sub-networks frozen. The mini-batch stochastic gradient descent algorithm is adopted here to learn the network parameters. When the feature generation sub-networks are fed by RGB frames, the whole training procedure for OFF sub-network takes 20000 iterations to converge with the learning rate initialized at 0.02 and decreased to its 0.1 using multi-step policy at the iteration 10000, 15000 and 18000. When input changes to temporal modality like optical flow, the learning rate is initialized at 0.05, and other policies are kept the same with what have been proposed in RGB. The batch size is set to 128 and all the training strategies described in previous sections are applied. When evaluating on UCF-101 and HMDB-51, we add dropout modules on spatial stream of OFF. When training for the TSN framework, we fix the segment number as 7 in both spatial and temporal networks as the final setting, which is reported in Table 6. For ablation studies, the number of segments is set to be 3 for faster training and evaluation. In both settings, the segment length of temporal network is 5, which means that each segment contains 5 continuous frames.There is no difference on training parameters for different modalities. However, when the input is RGB difference or optical flow, it would cost more time in both training and testing stages as more frames are read into the network.

5.2 Experimental Investigations on OFF.

In this section, we will investigate the performance of OFF under the TSN framework. The analysis for the performance of single and multiple modalities, and the performance comparison between the state-of-the-art will be shown. All the results for OFF based networks are trained with the same network backbone and strategies illustrated in previous sections for fair comparison.

Efficiency Evaluation. In this experiment, we evaluate the efficiency between the OFF based method and other state-of-the-art methods. The experimental results for efficiency and accuracy for different algorithms are summarized in Table 1. OFF(RGB) denotes our use of OFF for the network with RGB input, in this case, the OFF is acquired from spatial deep features. As a special case, the denotation *RGB Diff* represents the OFF calculated directly from consecutive RGB frames on the input level instead of on the feature level. After applying the OFF calculation to RGB frames, the processed inputs could be fed into the feature generation sub-network and the generated feature maps could be again used to calculate their corresponding OFF features on the feature level. The other methods we compared here includes TSN (Wang et al. 2016) with different inputs, motion vector based RGB+EMV-CNN (Zhang et al. 2016), dynamic image based CNN (Bilen et al. 2016) and current state-of-the-art 3D-CNN with two stream (Carreira and Zisserman 2017). From the Table 1, by applying the OFF to the spatial features and the RGB inputs, we can achieve a competitive accuracy 93.3% with only RGB inputs on the UCF-101 over three splits, which is even comparable with some Two-Stream based methods such as (Carreira and Zisserman 2017; Wang et al. 2016). Besides, our methods is still very efficient under this kind of settings. The whole network could run over 200 fps on GPU, while other methods listed here are either inefficient or not so effective as the Two-Stream based approaches.

Effectiveness Evaluation. In this part, we try to investigate the robustness of OFF when applying to different kinds of input. According to the definition in equation (4), we can replace the image I from RGB image to optical flow or RGB difference image to extract OFF on feature level for further experiments. Based on the scores predicted by different modalities, we can further improve the classification performance by fusing them together (Simonyan and Zisserman 2014a; Diba et al. 2016a; Wang et al. 2016; Zhang et al. 2016). We carry out the experimental results with various score fusing schemes on UCF-101 split 1, and summarize them in Table 2. Table 2 shows the results when different kinds of modalities are introduced as the network input. From each block separated by a horizon line, we can find that the OFF is complementary to other kinds of modalities, e.g. RGB and optical flow, and could get a remarkable gain every time the OFF is introduced. Besides, interestingly, the OFF is still working when the input modality is already describing the motion information. This phenomenon indicates that the *acceleration information* between frames might also make a difference in describing the temporal patterns.

Component Analysis for OFF. Briefly speaking, the OFF could be divided into two components, the spatial gradients and the temporal gradients. To analyze the effect of each single component, here we report it in the Table 3 on the dataset UCF-101 split1. The notation OFF_T and OFF_S represents the temporal and spatial gradients of the OFF respectively. Here in Table 3 we can conclude that the OFF_T (4.3% gain) is more effective than the OFF_S (0.7% gain) in the OFF module. This phenomenon is not hard to understand, as the CNN for the RGB stream is able to extract high-frequency spatial information, which is quite similar or even better than the simple Sobel operator, the improvement for the spatial gradient may thereby be suppressed by its counterpart. But the 0.7% gain is still remarkable, and could be regarded as an evidence supporting the theoretical analysis in the Section 3.

Comparison with the Hypercolumns CNN. As our network extracts intermediate deep features from a pre-trained CNN, such *hypercolumn* based network structure may lead to additional gain on specific datasets (Hariharan et al. 2015). Experiment and analysis are conducted to investigate whether the OFF is playing a key role for the improvement. The network architecture and all training strategies for the hypercolumn CNN are the same as that in OFF except for the removal of OFF unit, in other words, the hypercolumn network here is constructed as the same structure of OFF sub-network without OFF unit. In this case, the features from feature generation sub-networks are directly fed into the OFF sub-networks without the calculation of OFF.

From the experimental results shown in Table 4, it is clear that, despite the hypercolumn network could get a slight 0.5% improvement on UCF-101 split 1, its final accuracy is still apparently less than the one obtained by OFF(RGB). Therefore, a conclusion could be drawn that it is the OFF calculation rather than the hypercolumn structure that plays the key role in achieving the significant gain.

Comparison with other video classification methods. Above all, after the exploration and analysis of the OFF, we show our final result. As what has been done in TSN, we also assemble the classification scores obtained by different kinds of modalities. We sum the scores produced by each modality together, and get the final version output in Table 6. All the results are evaluated in the UCF-101 and HMDB-51 over 3 splits. Our results are obtained by assembling the scores from RGB, OFF(RGB), optical flow and their corresponding

Table 1 Experimental results of accuracy and efficiency for different real-time video action recognition methods on *UCF-101 over three splits*. Here the notation *Flow* represents the motion modality Optical Flow. Note that our OFF based algorithm could achieve the state-of-the-art performance among real-time algorithms.

Method	Speed (fps)	Acc.
TSN(RGB) Wang et al. (2016)	680	85.5%
TSN(RGB+RGB Diff) Wang et al. (2016)	340	91.0%
TSN(Flow) Wang et al. (2016)	14	87.9%
TSN(RGB+Flow) Wang et al. (2016)	14	94.0%
RGB+EMV-CNN Zhang et al. (2016)	390	86.4%
MDI+RGB Bilen et al. (2016)	<131	76.9%
Two-Stream I3D (RGB+Flow) Carreira and Zisserman (2017)	<14	93.4%
RGB+OFF(RGB)+RGB Diff+ OFF(RGB Diff)	206	**93.3%**

Table 2 Experimental results for different modalities using the OFF on *UCF-101 Split1*. Here Flow denotes the optical flow. OFF(*) denotes the use of OFF for the input *. For example, OFF(RGB) denotes the use of OFF for RGB input. The speed here illustrates the time cost for network forward. The results for RGB and RGB + Flow are from Wang et al. (2016). The OFF(RGB) provides a strong 4.5% improvement when fusing with RGB.

RGB	OFF (RGB)	RGB Diff	OFF (RGB Diff)	Flow	OFF (Flow)	Speed (fps)	Acc.
✓						680	85.5%
✓	✓					450	90.5%
✓		✓				340	90.7%
✓	✓	✓				257	92.0%
✓	✓	✓	✓			206	93.2%
✓				✓		14	93.5%
✓	✓			✓		14	95.1%
✓	✓			✓	✓	14	95.5%

Table 3 Accuracy while different components are applied into the OFF block. The notation OFF_T denotes the temporal difference.

RGB	OFF_T	OFF_S	Acc.
✓			85.5%
✓	✓		89.8%
✓	✓	✓	90.5%

Table 4 Experimental results of accuracy for hypercolumn network and the comparison with OFF on UCF-101 Split1. The denotation "Hyp-Net" indicates the output of hypercolumn network.

	RGB	Hyp-Net + RGB	OFF(RGB) + RGB
Acc.	85.5%	86.0%	90.5%

version of OFF(optical flow) together. When we add one more score from OFF(RGB Diff), a slight 0.3% gain is obtained compared to the version without it, and finally results in 96.0% on UCF-101 and 74.2% on HMDB-51. Note that we do not introduce improved Dense Trajectories (iDT) (Wang and Schmid 2013) into our network as the input. The components of inputs we need to prepare in advance for our final version result only consist of RGB and optical flow.

We compare our result with both the traditional approaches and deep learning based approaches. We obtain 2.0%/5.7% gain compared with the baseline Two-Stream TSN (Wang et al. 2016) on UCF-101 (Soomro et al. 2012) and HMDB-51 (Kuehne et al. 2011) respectively. Note that the final version TSN takes 3 modalities (RGB, Optical Flow and iDT) as network input. The other compared methods listed in Table 6 include iDT (Wang and Schmid 2013), Two-Stream ConvNet (Simonyan and Zisserman 2014a), Two-Stream + LSTM (Yue-Hei Ng et al. 2015), Temporal Deep-convolutional Descriptors (TDD) (Wang et al. 2015a), Long-term Temporal Convolutions (LTC) (Varol et al. 2016), Key Volume Mining Deep Framework (KVMDF) (Zhu et al. 2016), and also the current state-of-the-art methods such as Spatio-Temporal Pyramid (STP) (Wang et al. 2017), Saptio-Temporal Multiplier Network (STMN) (Feichtenhofer et al. 2017a), Spatio-Temporal Vector (Cosmin Duta et al. 2017), Lattice LSTM (L^2STM) (Sun et al. 2017), and I3D (Carreira and Zisserman 2017). The method I3D could achieve spectacular performance (98.0% on UCF-101, 80.7% on HMDB-51, over 3 splits) when proposing a new large dataset *Kinetics* for pretrain. While without the pre-training, the method I3D could achieve 93.4% on UCF-101 Split1. From the comparison with all the listed methods, we conclude that our OFF based method allow for state-of-the-art performance in video action recognition.

Experimental results on Kinetics. To further validate the effectiveness of our method, we conduct extensive experiments on another large-scale video dataset, *Kinetics*, which contains about 240k videos for training and 20k videos for validation. Videos in this dataset are categorized into 400 classes. We select another 2D CNN based method TSM (Lin et al. 2019) instead of TSN (Wang et al. 2016), as our baseline method on Kinetics. As shown in Table 5, our method could surpass all listed state-of-the-art methods. Note that on Kinetics, we no longer use optical flow and RGB difference as separate streams to extract the motion guid-

Table 5 We follow the widely-used evaluation protocol Li et al. (2020) to verity the effectiveness of our proposed method. As is shown, our method outperforms most methods.

Method	Backbone	Frame, Size	Acc	GFlops × Clips
I3D Carreira and Zisserman (2017)	Inception	64, 224	71.7%	$108 \times N/A$
R(2+1)D Tran et al. (2018)	ResNet-34	8, 112	74.3%	$152 \times N/A$
A^2-Net Chen et al. (2018)	ResNet-50	8, 224	74.6%	$41 \times N/A$
S3D-G Xie et al. (2018)	Inception	64, 224	74.7%	$71.4 \times N/A$
NL + I3D Wang et al. (2018b)	ResNet-50	32, 224	74.9%	70.5×30
GloRe Chen et al. (2019)	ResNet-50	8, 224	75.1%	28.9×30
SlowOnly Feichtenhofer et al. (2019)	ResNet-50	8, 224	74.8%	41.9×30
SlowFast Feichtenhofer et al. (2019)	ResNet-50	(4+32), 224	75.6%	36.1×30
STM Jiang et al. (2019)	ResNet-50	8, 224	73.7%	$N/A \times 30$
TSM Lin et al. (2019)	ResNet-50	8, 224	74.1%	33×30
TEI Liu et al. (2020)	ResNet-50	8, 224	74.7%	65×30
TEA Li et al. (2020)	ResNet-50	8, 224	75.0%	35×30
Our method	ResNet-50	8, 224	75.8%	59.5×30

ance, which indicates the RGB is the only ingredient in inputs for the entire network. To train on Kinetics, we apply a half-cosine learning rate schedule with the initial learning rate 0.01. The whole training process lasts 110 epochs to the final convergence. Other details about the hyper-parameters and training techniques are followed up with Lin et al. (2019). For fair comparison to all types of framework, we include the 3D-based state-of-the-art networks, *e.g.* I3D (Carreira and Zisserman 2017), S3D (Xie et al. 2018), R(2+1)D (Tran et al. 2017) SlowFast Net (Feichtenhofer et al. 2019) and Non-local Net (Wang et al. 2018b). These 3D networks usually requires more frames (*e.g.* 32, 64) to extract the temporal evidence, and will have higher computational costs due to the convolutions on temporal dimension. Compared with these methods, our approach could be more computationally economic and also shows great capability in terms of accuracy. Apart from these 3D based methods, we also compare our methods with 2D based methods, including A^2-Net (Chen et al. 2018), GloRe (Chen et al. 2019), STM(Jiang et al. 2019), TSM (Lin et al. 2019), TEI (Liu et al. 2020) and TEA (Li et al. 2020). Our method is also able to perform better than all these methods.

6 OFF on video object detection

In this section, we explore the possibility of transplanting OFF into the task of video object detection. Different from single image object detection task (Girshick 2015; Ren et al. 2015; He et al. 2017), video object detection is still a challenging problem due to problems such as motion blur, video defocus etc. To tackle these problems, a lot of researches (Kang et al. 2016, 2017b; Lee et al. 2016; Han et al. 2016; Kang et al. 2017a; Feichtenhofer et al. 2017b; Zhu et al. 2017a) have been proposed. Not only in methods that intend to enhance the algorithm on bounding-box level (Han et al. 2016; Kang et al. 2017a; Feichtenhofer et al. 2017b) or methods proposing to learn better temporal features (Zhu et al. 2017b,a), the motion representation always plays an important role. To achieve better performance, these methods calculate the optical flow by using conputationally expensive methods like FlowNet (Dosovitskiy et al. 2015; Ilg et al. 2016). However, our OFF as a fast and robust inter-frame motion representation, could essentially help to alleviate these problems with only a little computational cost.

6.1 Framework Introduction

Following the FGFA method (Zhu et al. 2017a), we choose R-FCN (Dai et al. 2016) as the detection framework with the ResNet-101 (He et al. 2016b) as the feature extraction network. In the feature extraction stage, we first remove the last global average pooling layer and its following fully-connected layer of the ResNet-101 and then apply another 3 × 3 convolution layer to reduce the output dimension into 1024. The OFF module is applied onto the output feature of the network to get the motion representation and several residual blocks with SE block (Hu et al. 2018) are stacked to refine features with these motion information.

In the detection stage, we follow the settings in (Dai et al. 2016). We split the output of the feature extraction network (with 1024 channels) into two independent parts, which are then fed into the RPN network and the R-FCN network respectively. The parameter k for position-sensitive score maps is set to 7.

Specific details about the network are shown in the Figure 5. Based on the above setting, we also find that applying OFF units to multiple scales of feature maps can further improve the performance. Several residual blocks with SE module (Hu et al. 2018) are used to

Table 6 Performance comparison to the state-of-the-art methods on UCF-101 and HMDB-51 over 3 splits.

Method	UCF-101	HMDB-51
iDT Wang and Schmid (2013)	86.4%	61.7%
Two-Stream Simonyan and Zisserman (2014a)	88.0%	59.4%
Two-Stream TSN Wang et al. (2016)	94.0%	68.5%
Three-Stream TSN Wang et al. (2016)	94.2%	69.4%
Two-Stream+LSTM Yue-Hei Ng et al. (2015)	88.6%	-%
TDD+iDT Wang et al. (2015a)	91.5%	65.9%
LTC+iDT Varol et al. (2016)	91.7%	64.8%
KVMDF Zhu et al. (2016)	93.1%	63.3%
STP Wang et al. (2017)	94.6%	68.9%
STMN+iDT Feichtenhofer et al. (2017a)	94.9%	72.2%
ST-VLMPF+iDT Cosmin Duta et al. (2017)	94.3%	73.1%
L^2STM Sun et al. (2017)	93.6%	66.2%
Two-Stream I3D Carreira and Zisserman (2017)	93.4%	66.4%
Two-Stream I3D (with Kinetics 300k) Carreira and Zisserman (2017)	98.0%	80.7%
Ours	**96.0%**	**74.2%**

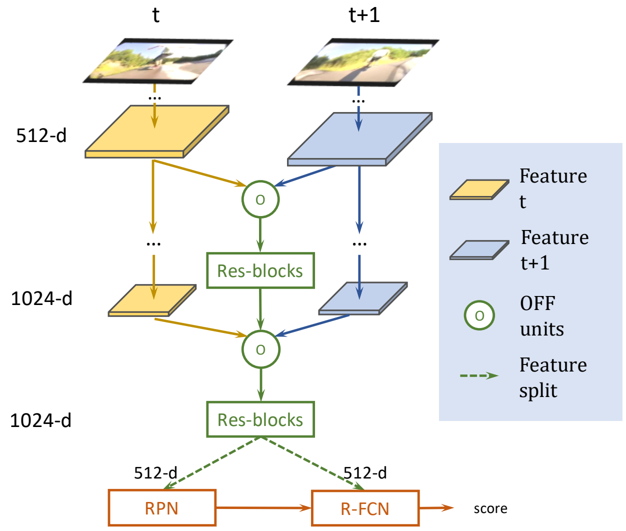

Fig. 5 Illustration of applying OFF on the video object detection task. Here the notation $* - d$ represents the feature is of $*$ channels. We reduce the output feature dimension to 1024 at the end of ResNet-101. Within the ResNet-101, OFF is applied on the 512-d feature maps and 1024-d output feature maps separately. Several residual blocks with SE follow the OFF units to further blend the feature maps with motion information. In the detection stage, we firstly split the 1024-d feature maps to the former 512-d and the latter 512-d feature maps, and feed them to the RPN network and R-FCN network separately.

connect between OFF units from different levels of resolution.

6.2 Experimental Setup

Datasets. Following the setting used in Zhu et al. (2017a), ImageNet VID dataset (Russakovsky et al. 2015) is used for the video object detection task. 3862 video snippets from the training set are used in training stage and 555 snippets from the validation set are used for model testing. There are overall 30 object categories, which are a subset of ImageNet DET dataset.

Implementation Details. In the training phase, both the data in the ImageNet DET training set and the ImageNet VID training set were used. Stochastic Gradient Descent (SGD) is adopted as the optimizer. We perform 220k iterations on 4 GPUs, with mini-batch size 1 on each. The learning rate is set to be 0.00025 and 0.000025 respectively for the first 73k iterations and the rest.

6.3 Experimental results

Experimental results of mean Average Precision (mAP) on the ImageNet VID validation set are shown in Table 7. The corresponding runtime is in the last column of the table. The result of the baseline method, ResNet-101+RFCN, is shown in the first row, and the experiment results of applying OFF on single and multiple scales are shown in the last two rows respectively. As we can see from the table, applying OFF on single and multiple scales can achieve 2.1 and 2.6 mAP(%) gain respectively.

Besides, the comparison between the proposed method and the FGFA is shown in the Table 8. Both the reimplementation result and the official result (in the parentheses) of the FGFA are shown in the first row. We exhibit that the proposed method is about 4.32 times

Table 7 Experimental results of mAP for video object detection methods on ImageNet VID validation. Note that SS represents Single Scale, which means only applying OFF units on the 1024-d feature maps, MS represents Multi Scales, which means applying OFF units on the both 512-d feature maps and 1024-d feature maps. The MS version is our final experiment setting. The corresponding runtime is shown in the final column.

Method	mAP(%)	runtime(ms)
RFCN Dai et al. (2016)	73.12	80.7
RFCN+OFF(SS) w/o SE	74.96	121.8
RFCN+OFF(SS)	75.23	128.3
RFCN+OFF(MS)	75.78	137.7

Table 8 Comparison between the proposed method and the FGFA method (Zhu et al. 2017a) for video object detection task on ImageNet VID validation. The ours refers to applying OFF on multiple scales.

Method	mAP(%)	runtime(ms)
FGFA Zhu et al. (2017a)	76.12 (76.3)	733
Ours	75.78	137.7

faster than the FGFA while the performances (mAP) of these two methods are comparable.

7 Apply OFF in Video Compression Artifact Removal

In this part, in order to explore the robustness and the effectiveness of the OFF on low-level vision tasks, we further apply the OFF module to the task called video compression artifact removal.

7.1 Introduction of video compression artifact removal

Video compression artifact removal (Maggioni et al. 2012; Xue et al. 2017; Lu et al. 2018) aims to reduce the compression artifact made by lossy compression algorithms. Different from image artifact removal tasks, the temporal information is an additional data dimension in videos, which plays a very important role in recovering video quality. In other words, how to effectively use inter-frame information is critical for removing video compression artifact.

To utilize the temporal information, there are usually two common solutions. The first is to estimate the motion information and warp features based on the motion before post-processing (Xue et al. 2017; Bao et al. 2018; Caballero et al. 2017; Tao et al. 2017; Liu et al. 2017; Kappeler et al. 2016). This solution heavily relies on the quality of the estimated motion representation, while the existing algorithms for motion estimation are usually either too slow or inaccurate. Another solution is to fuse features or restored images of multiple frames (*e.g.* by concatenation or 3D convolution) (Lu et al. 2018; Shi et al. 2016; Kim et al. 2018). Here we follow the first scheme and utilize the fast and robust OFF unit as the inter-frame motion representation to alleviate the inconsistency between the robustness and efficiency. However, how to apply the OFF module to the video compression artifact removal task?

Following the DnCNN (Zhang et al. 2017), we take the residual learning strategy and formulate the task as learning the artifact residual of the input video frame. Formally, we will have:

$$\begin{aligned} N_t &= I(x,y,t) - D(x,y,t), \\ N_{t+\Delta t} &= I(x+\Delta x, y+\Delta y, t+\Delta t) \\ &\quad - D(x+\Delta x, y+\Delta y, t+\Delta t), \end{aligned} \tag{10}$$

where $I(x,y,t)$ and $D(x,y,t)$ denote the pixel at the location (x,y) at time t for the noisy (input) image and denoised (target) image respectively, and N_t represents the learned artifact residual at time t. For frames t and $(t+\Delta t)$, Δx and Δy are the spatial pixel displacement in x and y axes respectively.

As formulated in the Equation (1), the proposed OFF is motivated by the brightness constant constraint. Under this assumption, ideally for any images without noise, the brightness for the same location of an given object is usually kept unchanged over time, so there is:

$$D(x,y,t) = D(x+\Delta x, y+\Delta y, t+\Delta t), \tag{11}$$

Generally we will have a mapping function that could remove the noise and transform the noisy image into the noiseless image. Hereby, we denote g as the mapping function for transforming the noisy frame I with parameters W_g. For short representation, we define $p = (x,y,t)$ as the spatio-temporal location of a specific pixel, then as for the feature-level for any differentialable function g, we can get:

$$N_{t+\Delta t} = g(I;W_g)(x+\Delta x, y+\Delta y, t+\Delta t) - D(x,y,t), \tag{12}$$

Similar to the derivation from Equation 2 to Equation (4), we can further derive:

$$\begin{aligned} \frac{N_{t+\Delta t}}{\Delta t} &= \frac{\partial g(I;W_g)(p)}{\partial x} v_x + \frac{\partial g(I;W_g)(p)}{\partial y} v_y \\ &\quad + \frac{\partial g(I;W_g)(p)}{\partial t} + \frac{I(x,y,t) - D(x,y,t)}{\Delta t}, \end{aligned} \tag{13}$$

Based on Equation 10, there is:

$$\begin{aligned}
\frac{N_{t+\Delta t}}{\Delta t} &= \frac{\partial g(I;W_g)(p)}{\partial x}v_x + \frac{\partial g(I;W_g)(p)}{\partial y}v_y \\
&\quad + \frac{\partial g(I;W_g)(p)}{\partial t} + \frac{N_t}{\Delta t}, \\
\frac{N_{t+\Delta t}-N_t}{\Delta t} &= \frac{\partial g(I;W_g)(p)}{\partial x}v_x + \frac{\partial g(I;W_g)(p)}{\partial y}v_y \\
&\quad + \frac{\partial g(I;W_g)(p)}{\partial t}, \\
\frac{\Delta N}{\Delta t} &= \frac{\partial g(I;W_g)(p)}{\partial x}v_x + \frac{\partial g(I;W_g)(p)}{\partial y}v_y \\
&\quad + \frac{\partial g(I;W_g)(p)}{\partial t},
\end{aligned} \tag{14}$$

where the ΔN represents the differences between noise information of frames, and Δt is a constant number. Thereby we can find that the vector $[\frac{\partial g(I;W_g)(p)}{\partial x}, \frac{\partial g(I;W_g)(p)}{\partial y}, \frac{\partial g(I;W_g)(p)}{\partial t}]$, which is defined as the OFF in this paper, is encoded with the optical flow v_x, v_y in one single equation. When noise applied in the two consecutive frames are same (a.k.a. $\Delta N = 0$), the brightness constant constraint here for the noisy videos is equivalent to the one defined for the ordinary noiseless videos. Details of the mapping function g will be discussed in next part.

7.2 Framework Introduction

For the framework, we choose DnCNN (Zhang et al. 2017) as our baseline to validate the effectiveness of OFF on video compression artifact removal task. To build the baseline, We follow the basic settings of DnCNN and retrain the model on Vimeo-90K (Xue et al. 2017) dataset. The basic structure consists of 20 convolutional layers and a big identity mapping from input to the output. Every convolutional layer in the network is followed by ReLU activation function without Batch Normalization (BN) (Ioffe and Szegedy 2015a). The input size of the network is 448 × 256, and the whole process does not evolve any up-sampling or down-sampling process.

We first apply one OFF unit between two adjacent frames on different stages of this structure. Experiments show that OFF performs better on the early stage (after the first 3 convolutional layers) of DnCNN. Extra 5 convolutional layers follow the OFF unit to better refine the mixture information of frame feature and inter-frame motion representations. We also try to apply multiple OFF units on different stages of DnCNN, but that only exhibits trivial improvement. We think the reason is that local features (extracted by the shallow layers) are more important than global features for reducing compression artifacts. Besides, increasing the number of input frames can also help improve performance. Considering the amount of model parameters, three consecutive frames(t-1, t, t+1) are proposed as the final setting.

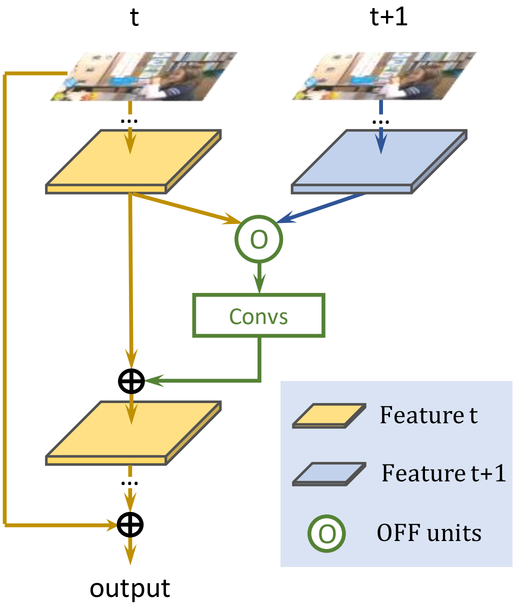

Fig. 6 How to apply the OFF for video compression artifact removal. Take two input frames as an example, The OFF unit followed by several convolutional layers are applied to extract motion representation from the output features of the feature extraction network. Then the motion representations optimized from the OFF are merged into the basic feature via a sum operation. In the end, following the residual learning strategy of DnCNN, an identity mapping from the input to the output is applied.

Specific details about the network are shown in the Figure 6. Two input frames are taken as an example for simplicity. At the very beginning of the whole network, each input frame is fed into three convolutional layers with kernel size 3 × 3 independently. Then OFF units are applied on these features to get the motion representation. In the following, 5 convolutional layers and an identity mapping from input feature of OFF units are utilized to merge the feature and the motion representation. In the final stage, the rest 18 convolutional layers are proposed for further refinement. Finally, a big identity mapping from input to the output is applied.

Discussion: The difference between the high level tasks and the low level tasks. Generally, the video action recognition and the video object detection belong to high-level vision tasks, and video compression artifact removal belongs to low-level tasks. Therefore, there are some differences for adapting the OFF mod-

Table 9 Average PSNR results for video compression artifact removal methods on Vimeo-90K dataset Xue et al. (2017). Note that q20 and q40 represents the quantization factor of video compression algorithm (JPEG2000). The higher the quantization factor, the lower the quality of video. The (2) and (3) represents the num of frames utilized in the experiments.

Method	q40(dB)	q20(dB)
DnCNN Zhang et al. (2017)	35.22	37.26
DnCNN_OFF(2)	35.78	37.71
DnCNN_OFF(3)	36.07	38.00

ule between the high-level tasks and low-level tasks. As we mentioned before, the OFF module is applied to the 512-d feature maps in the later stage for video object detection, but for video compression artifact removal, it is applied just after the first 3 convolutional layers in the early stage. This is because that the low-level details are more important in the video compression artifact removal. Besides, the formulation of DnCNN is to learn the residual noise information. Thereby, instead of directly feeding the output feature map of OFF module to the next stage, the output feature of OFF module need to be added back for video compression artifact removal.

7.3 Experimental setup

Datasets. All experiments are conducted on the benchmark dataset Vimeo-90K (Xue et al. 2017), which is built for evaluating various different video restoration tasks, such as video artifact removal, video denoising, video super-resolution *etc.* There are 89,800 independent clips from 4,278 videos with resolution of 448×256 . Each clip includes 7 frames. 64,612 clips are used for training and 7,824 clips for evaluation. No data augmentation is applied. We use FFmpeg to generate the encoded images. Experiments are performed on codec JPEG2000 with quantization parameter q=20 and q=40 separately.

Implementation Details. In the model training process, we use the Adam optimizer (Kingma and Ba 2014) with the initial learning rate as 5×10^{-4}. Cosine learning rate scheduler is adopted. Mini-batch size is set to 28 on 4 GPUs, and Xavier uniform is applied for weight initialization. For all experiments, we train the network for 50 epochs. The model is able to be trained in an end-to-end manner and do not need any pre-training or finetuning process. In the testing process, only the 4th frame of each clip in the Vimeo-90K dataset is evaluated. Peak Signal-to-Noise Ratio (PSNR) is used as the evaluation metrics.

7.4 Experimental results

The experimental results of average PSNR for video compression (JPEG2000) artifact removal task are shown in Table 9 where the baseline result is shown in the first line. In the rest two lines, experimental results of applying one OFF unit on two adjacent frames and three adjacent frames are shown respectively. As we can see from the table, The OFF unit can significantly help to improve the performance with around 0.8dB.

8 Conclusion

In this paper, we have presented *Optical Flow guided Feature (OFF)*, a novel motion representation derived from and guided by the optical flow. OFF is both fast and robust. By plugging the OFF into CNN framework, the result with only RGB as input on UCF-101 is even comparable to the result obtained by Two-Stream (RGB+Optical Flow) approaches, and at the same time, the OFF plugged network is still very efficient with the speed over 200 frames per second. Besides, it has been proven that the OFF is still complementary to other motion representations like optical flow. By applying the OFF unit into CNN architectures for both the high-level vision tasks, *e.g.* video action recognition and video object detection, and low-level vision tasks like video compression artifact removal, we could boost the performance by a large margin and simultaneously maintain the efficiency.

References

Bao W, Lai WS, Zhang X, Gao Z, Yang MH (2018) Memcnet: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. arXiv preprint arXiv:181008768

Barron JL, Fleet DJ, Beauchemin SS (1994) Performance of optical flow techniques. IJCV 12(1):43–77

Bigun J, Granlund GH, Wiklund J (1991) Multidimensional orientation estimation with applications to texture analysis and optical flow. T-PAMI 13(8):775–790

Bilen H, Fernando B, Gavves E, Vedaldi A, Gould S (2016) Dynamic image networks for action recognition. In: CVPR, pp 3034–3042

Brox T, Bruhn A, Papenberg N, Weickert J (2004) High accuracy optical flow estimation based on a theory for warping. In: ECCV, Springer, pp 25–36

Caballero J, Ledig C, Aitken A, Acosta A, Totz J, Wang Z, Shi W (2017) Real-time video super-resolution with spatio-temporal networks and motion compensation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4778–4787

Carreira J, Zisserman A (2017) Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. arXiv preprint arXiv:170507750

Chen Y, Kalantidis Y, Li J, Yan S, Feng J (2018) Aˆ 2-nets: Double attention networks. In: Advances in neural information processing systems, pp 352–361

Chen Y, Rohrbach M, Yan Z, Shuicheng Y, Feng J, Kalantidis Y (2019) Graph-based global reasoning networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 433–442

Chu X, Yang W, Ouyang W, Ma C, Yuille AL, Wang X (2017) Multi-context attention for human pose estimation. In: CVPR

Cosmin Duta I, Ionescu B, Aizawa K, Sebe N (2017) Spatio-Temporal Vector of Locally Max Pooled Features for Action Recognition in Videos. In: CVPR, pp 3097–3106

Dai J, Li Y, He K, Sun J (2016) R-fcn: Object detection via region-based fully convolutional networks. In: Advances in neural information processing systems, pp 379–387

Dalal N, Triggs B, Schmid C, Dalal N, Triggs B, Schmid C, Detection H, Oriented U (2006) Human Detection Using Oriented Histograms of Flow and Appearance. In: ECCV, pp 428–441

Diba A, Pazandeh AM, Van Gool L (2016a) Efficient two-stream motion and appearance 3d cnns for video classification. arXiv preprint arXiv:160808851

Diba A, Sharma V, Van Gool L (2016b) Deep temporal linear encoding networks. arXiv preprint arXiv:161106678

Dosovitskiy A, Fischer P, Ilg E, Hausser P, Hazirbas C, Golkov V, van der Smagt P, Cremers D, Brox T (2015) Flownet: Learning optical flow with convolutional networks. In: ICCV, pp 2758–2766

Feichtenhofer C, Pinz A, Wildes R (2016) Spatiotemporal residual networks for video action recognition. In: NIPS, pp 3468–3476

Feichtenhofer C, Pinz A, Wildes RP (2017a) Spatiotemporal multiplier networks for video action recognition. In: CVPR

Feichtenhofer C, Pinz A, Zisserman A (2017b) Detect to track and track to detect. In: Proceedings of the IEEE International Conference on Computer Vision, pp 3038–3046

Feichtenhofer C, Fan H, Malik J, He K (2019) Slowfast networks for video recognition. In: Proceedings of the IEEE international conference on computer vision, pp 6202–6211

Fernando B, Gavves E, Oramas J, Ghodrati A, Tuytelaars T (2017) Rank pooling for action recognition. T-PAMI 39(4):773–787

Girshick R (2015) Fast R-CNN. In: ICCV, pp 1440–1448

Han W, Khorrami P, Paine TL, Ramachandran P, Babaeizadeh M, Shi H, Li J, Yan S, Huang TS (2016) Seq-nms for video object detection. arXiv preprint arXiv:160208465

Hariharan B, Arbeláez P, Girshick R, Malik J (2015) Hypercolumns for object segmentation and fine-grained localization. In: CVPR, pp 447–456

He K, Zhang X, Ren S, Sun J (2016a) Deep residual learning for image recognition. In: CVPR, pp 770–778

He K, Zhang X, Ren S, Sun J (2016b) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778

He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: Computer Vision (ICCV), 2017 IEEE International Conference on, IEEE, pp 2980–2988

Horn, Berthold KP; Schunck BG (1981) Determining Optical Flow. Artificial Intelligence 17:185–203

Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7132–7141

Ilg E, Mayer N, Saikia T, Keuper M, Dosovitskiy A, Brox T (2016) Flownet 2.0: Evolution of optical flow estimation with deep networks. arXiv preprint arXiv:161201925

Ioffe S, Szegedy C (2015a) Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:150203167

Ioffe S, Szegedy C (2015b) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML, pp 448–456

Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: Convolutional Architecture for Fast Feature Embedding. arXiv preprint arXiv:14085093

Jiang B, Wang M, Gan W, Wu W, Yan J (2019) Stm: Spatiotemporal and motion encoding for action recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2000–2009

Kang K, Ouyang W, Li H, Wang X (2016) Object detection from video tubelets with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 817–825

Kang K, Li H, Xiao T, Ouyang W, Yan J, Liu X, Wang X (2017a) Object detection in videos with tubelet proposal networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 727–735

Kang K, Li H, Yan J, Zeng X, Yang B, Xiao T, Zhang C, Wang Z, Wang R, Wang X, et al. (2017b) T-cnn: Tubelets with convolutional neural networks for object detection from videos. IEEE Transactions on Circuits and Systems for Video Technology

Kappeler A, Yoo S, Dai Q, Katsaggelos AK (2016) Video super-resolution with convolutional neural networks. IEEE Transactions on Computational Imaging 2(2):109–122

Kim SY, Lim J, Na T, Kim M (2018) 3dsrnet: Video super-resolution using 3d convolutional neural networks. arXiv preprint arXiv:181209079

Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980

Klaser A, Marszalek M, Schmid C (2008) A Spatio-Temporal Descriptor Based on 3D-Gradients. In: BMVC, pp 275–1

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: NIPS, pp 1097–1105

Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T (2011) HMDB: A large video database for human motion recognition. In: ICCV, pp 2556–2563

Lee B, Erdenee E, Jin S, Nam MY, Jung YG, Rhee PK (2016) Multi-class multi-object tracking using changing point detection. In: European Conference on Computer Vision, Springer, pp 68–83

Li Y, Fang C, Yang J, Wang Z, Lu X, Yang MH (2018) Flow-grounded spatial-temporal video prediction from still images. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 600–615

Li Y, Ji B, Shi X, Zhang J, Kang B, Wang L (2020) Tea: Temporal excitation and aggregation for action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 909–918

Lin J, Gan C, Han S (2019) Tsm: Temporal shift module for efficient video understanding. In: Proceedings of the IEEE International Conference on Computer Vision, pp 7083–7093

Liu D, Wang Z, Fan Y, Liu X, Wang Z, Chang S, Huang T (2017) Robust video super-resolution with learned temporal dynamics. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2507–2515

Liu Z, Luo D, Wang Y, Wang L, Tai Y, Wang C, Li J, Huang F, Lu T (2020) Teinet: Towards an efficient architecture for video recognition. In: AAAI, pp 11669–11676

Lu G, Ouyang W, Xu D, Zhang X, Gao Z, Sun MT (2018) Deep kalman filtering network for video compression artifact reduction. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 568–584

Maggioni M, Boracchi G, Foi A, Egiazarian K (2012) Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms. IEEE Transactions on image processing 21(9):3952–3966

Newell A, Yang K, Deng J (2016) Stacked Hourglass Networks for Human Pose Estimation. In: ECCV, Springer, pp 483–499, DOI 10.1007/978-3-319-46484-8_29, URL http://dx.doi.org/10.1007/978-3-319-46484-8{_}29

Ng JYH, Choi J, Neumann J, Davis LS (2016) ActionFlowNet: Learning Motion Representation for Action Recognition. arXiv preprint arXiv:161203052

Ouyang W, Zeng X, Wang X (2016) Learning mutual visibility relationship for pedestrian detection with a deep model. IJCV 120(1):14–27

Ouyang W, Wang K, Zhu X, Wang X (2017) Chained cascade network for object detection. In: ICCV

Pan J, Wang C, Jia X, Shao J, Sheng L, Yan J, Wang X (2019) Video generation from single semantic label map. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

Peng X, Wang L, Wang X, Qiao Y (2016) Bag of Visual Words and Fusion Methods for Action Recognition: Comprehensive Study and Good Practice. Computer Vision and Image Understanding 150:109–125, DOI 10.1016/j.cviu.2016.03.013, URL http://arxiv.org/abs/1405.4506, 1405.4506

Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: NIPS, pp 91–99

Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, et al. (2015) Imagenet large scale visual recognition challenge. International journal of computer vision 115(3):211–252

Scovanner P, Ali S, Shah M (2007) A 3-dimensional sift descriptor and its application to action recognition. In: ACM'MM, pp 357–360, DOI 10.1145/1291233.1291311, URL http://dl.acm.org/citation.cfm?id=1291311{%}7B{%}25{%}7D5Cnhttp://portal.acm.org/citation.cfm?doid=1291233.1291311

Shi W, Caballero J, Huszár F, Totz J, Aitken AP, Bishop R, Rueckert D, Wang Z (2016) Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1874–1883

Shi Y, Tian Y, Wang Y, Zeng W, Huang T (2017) Learning long-term dependencies for action recognition with a biologically-inspired deep network. In: CVPR, pp 716–725

Simonyan K, Zisserman A (2014a) Two-stream convolutional networks for action recognition in videos. In: NIPS, pp 568–576

Simonyan K, Zisserman A (2014b) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556

Soomro K, Zamir AR, Shah M (2012) UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. arXiv preprint arXiv:12120402 URL http://arxiv.org/abs/1212.0402

Sun L, Jia K, Yeung DY, Shi BE (2015) Human action recognition using factorized spatio-temporal convolutional networks. In: ICCV, pp 4597–4605

Sun L, Jia K, Chen K, Yeung DY, Shi BE, Savarese S (2017) Lattice Long Short-Term Memory for Human Action Recognition. arXiv preprint arXiv:170803958

Sun S, Kuang Z, Sheng L, Ouyang W, Zhang W (2018) Optical flow guided feature: A fast and robust motion representation for video action recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going Deeper with Convolutions. In: CVPR, pp 1–9

Tao X, Gao H, Liao R, Wang J, Jia J (2017) Detail-revealing deep video super-resolution. In: Proceedings of the IEEE International Conference on Computer Vision, pp 4472–4480

Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3d convolutional networks. In: ICCV, pp 4489–4497

Tran D, Ray J, Shou Z, Chang SF, Paluri M (2017) ConvNet Architecture Search for Spatiotemporal Feature Learning. arXiv preprint arXiv:170805038

Tran D, Wang H, Torresani L, Ray J, LeCun Y, Paluri M (2018) A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp 6450–6459

Varol G, Laptev I, Schmid C (2016) Long-term temporal convolutions for action recognition. arXiv preprint arXiv:160404494

Varol G, Laptev I, Schmid C (2017) Long-term temporal convolutions for action recognition. T-PAMI

Wang H, Schmid C (2013) Action Recognition with Improved Trajectories. In: ICCV, pp 3551–3558

Wang H, Klaser A, Schmid C, Liu CL (2011) Action recognition by dense trajectories. In: CVPR, IEEE, pp 3169–3176

Wang L, Qiao Y, Tang X (2015a) Action recognition with trajectory-pooled deep-convolutional descriptors. In: ICCV, pp 4305–4314

Wang L, Xiong Y, Wang Z, Qiao Y (2015b) Towards good practices for very deep two-stream convnets. arXiv preprint arXiv:150702159

Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Van Gool L (2016) Temporal segment networks: Towards good practices for deep action recognition. ECCV pp 20–36

Wang TC, Liu MY, Zhu JY, Liu G, Tao A, Kautz J, Catanzaro B (2018a) Video-to-video synthesis. In: Advances in Neural Information Processing Systems

Wang X, Girshick R, Gupta A, He K (2018b) Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 7794–7803

Wang Y, Long M, Wang J, Yu PS (2017) Spatiotemporal Pyramid Network for Video Action Recognition. In: CVPR, pp 1529–1538

Wei SE, Ramakrishna V, Kanade T, Sheikh Y (2016) Convolutional pose machines. In: CVPR, pp 4724–4732

Xie S, Sun C, Huang J, Tu Z, Murphy K (2018) Rethinking spatiotemporal feature learning: Speed-accuracy tradeoffs in video classification. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 305–321

Xue T, Chen B, Wu J, Wei D, Freeman WT (2017) Video enhancement with task-oriented flow. arXiv preprint arXiv:171109078

Yang W, Li S, Ouyang W, Li H, Wang X (2017) Learning feature pyramids for human pose estimation. In: ICCV

Yue-Hei Ng J, Hausknecht M, Vijayanarasimhan S, Vinyals O, Monga R, Toderici G (2015) Beyond short snippets: Deep networks for video classification. In: CVPR, pp 4694–4702

Zach C, Pock T, Bischof H (2007) A duality based approach for realtime TV-L1 optical flow. Pattern Recognition pp 214–223

Zeng X, Ouyang W, Yan J, Li H, Xiao T, Wang K, Liu Y, Zhou Y, Yang B, Wang Z, et al. (2017) Crafting gbd-net for object detection. T-PAMI

Zhang B, Wang L, Wang Z, Qiao Y, Wang H (2016) Real-time action recognition with enhanced motion vector CNNs. In: CVPR, pp 2718–2726

Zhang K, Zuo W, Chen Y, Meng D, Zhang L (2017) Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Transactions on Image Processing 26(7):3142–3155

Zhao H, Tian M, Sun S, Shao J, Yan J, Yi S, Wang X, Tang X (2017) Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In: CVPR

Zhu W, Hu J, Sun G, Cao X, Qiao Y (2016) A key volume mining deep framework for action recognition. In: CVPR, pp 1991–1999

Zhu X, Wang Y, Dai J, Yuan L, Wei Y (2017a) Flow-guided feature aggregation for video object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp 408–417

Zhu X, Xiong Y, Dai J, Yuan L, Wei Y (2017b) Deep feature flow for video recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2349–2358