This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Learning Spatio-Temporal Representation with Local and Global Diffusion*

Zhaofan Qiu[†], Ting Yao[‡], Chong-Wah Ngo[§], Xinmei Tian[†], and Tao Mei[‡]
[†] University of Science and Technology of China, Hefei, China
[‡] JD AI Research, Beijing, China [§] City University of Hong Kong, Kowloon, Hong Kong
{zhaofanqiu, tingyao.ustc}@gmail.com, cscwngo@cityu.edu.hk, xinmei@ustc.edu.cn, tmei@live.com

Abstract

Convolutional Neural Networks (CNN) have been regarded as a powerful class of models for visual recognition problems. Nevertheless, the convolutional filters in these networks are local operations while ignoring the large-range dependency. Such drawback becomes even worse particularly for video recognition, since video is an information-intensive media with complex temporal variations. In this paper, we present a novel framework to boost the spatio-temporal representation learning by Local and Global Diffusion (LGD). Specifically, we construct a novel neural network architecture that learns the local and global representations in parallel. The architecture is composed of LGD blocks, where each block updates local and global features by modeling the diffusions between these two representations. Diffusions effectively interact two aspects of information, i.e., localized and holistic, for more powerful way of representation learning. Furthermore, a kernelized classifier is introduced to combine the representations from two aspects for video recognition. Our LGD networks achieve clear improvements on the large-scale Kinetics-400 and Kinetics-600 video classification datasets against the best competitors by 3.5% and 0.7%. We further examine the generalization of both the global and local representations produced by our pre-trained LGD networks on four different benchmarks for video action recognition and spatio-temporal action detection tasks. Superior performances over several state-of-the-art techniques on these benchmarks are reported.

1. Introduction

Today's digital contents are inherently multimedia. Particularly, with the proliferation of sensor-rich mobile devices, images and videos become media of everyday communication. Therefore, understanding of multimedia content becomes highly demanded, which accelerates the development of various techniques in visual annotation. Among them, a fundamental breakthrough underlining the success of these techniques is representation learning. This can be evidenced by the success of Convolutional Neural Networks (CNN), which demonstrates high capability of learning and generalization in visual representation. For example, an ensemble of residual nets [11] achieves 3.57% top-5 error on ImageNet test set, which is even lower than 5.1% of the reported human-level performance. Despite these impressive progresses, learning powerful and generic spatio-temporal representation remains challenging, due to larger variations and complexities of video content.

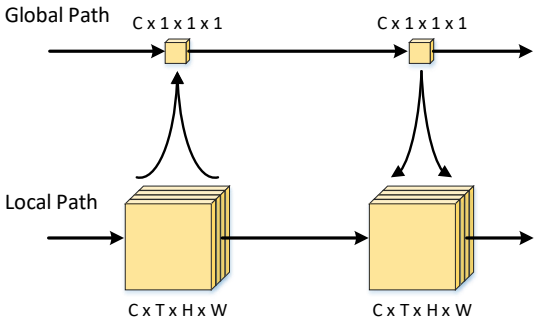


Figure 1. The schematic illustration of the Local and Global Diffusion block. The diffusion between local and global paths enrich the representation learnt on each path.

A natural extension of CNN from image to video domain is by direct exploitation of 2D CNN on video frames [18, 34, 41] or 3D CNN on video clips [15, 28, 29, 38]. An inherent limitation of this extension, however, is that each convolution operation, either 2D or 3D, processes only a local window of neighboring pixels. As window size is normally set to a small value, the holistic view of field cannot be adequately captured. This problem is engineered by performing repeated convolution and pooling operations to capture long-range visual dependencies. In this way, receptive fields can be increased through progressive propagation of signal responses over local operations. When a network is deep, the repeated operations, however, post difficulty to parameter optimization. Concretely, the connection be-

*This work was performed at JD AI Research.

tween two distant pixels are only established after a large number of local operations, resulting in vanishing gradient.

In this paper, we present Local and Global Diffusion (LGD) networks – a novel architecture to learn spatio-temporal representations capturing large-range dependencies, as shown in Figure 1. In LGD networks, the feature maps are divided into local and global paths, respectively describing local variation and holistic appearance at each spatio-temporal location. The networks are composed of several staked LGD blocks of each couples with mutually inferring local and global paths. Specifically, the inference takes place by attaching the residual value of global path to the output of local feature map, while the feature of global path is produced by linear embedding of itself with the global average pooling of local feature map. The diffusion is constructed at every level from bottom to top such that the learnt representations encapsulate a holistic view of content evolution. Furthermore, the final representations from both paths are combined by a novel kernel-based classifier proposed in this paper.

The main contribution of this work is the proposal of the Local and Global Diffusion networks, which is a two-path network aiming to model local and global video information. The diffusion between two paths enables the capturing of large-range dependency by the learnt video representations economically and effectively. Through an extensive set of experiments, we demonstrate that our LGD network outperforms several state-of-the-art models on six benchmarks, including Kinetics-400, Kinetics-600, UCF101, HMDB51 for video action recognition and J-HMDB, UCF101D for spatio-temporal action detection.

2. Related Work

We broadly categorize the existing research in video representation learning into hand-crafted and deep learning based methods.

Hand-crafted representation starts by detecting spatio-temporal interest points and then describing them with local representations. Examples of representations include Space-Time Interest Points (STIP) [21], Histogram of Gradient and Histogram of Optical Flow [22], 3D Histogram of Gradient [19], SIFT-3D [33] and Extended SURF [45]. These representations are extended from image domain to model temporal variation of 3D volumes. One particularly popular representation is the dense trajectory feature proposed by Wang *et al.*, which densely samples local patches from each frame at different scales and then tracks them in a dense optical flow field [40]. These hand-crafted descriptors, however, are not optimized and hardly to be generalized across different tasks of video analysis.

The second category is **deep learning based video representation**. The early works are mostly extended from image representation by applying 2D CNN on video frames. Karparthy *et al.* stack CNN-based frame-level representations in a fixed size of windows and then leverage spatio-temporal convolutions for learning video representation [18]. In [34], the famous two-stream architecture is devised by applying two 2D CNN architectures separately on visual frames and staked optical flows. This two-stream architecture is further extended by exploiting convolutional fusion [5], spatio-temporal attention [24], temporal segment networks [41, 42] and convolutional encoding [4, 27] for video representation learning. Ng *e*t al. [49] highlight the drawback of performing 2D CNN on video frames, in which long-term dependencies cannot be captured by two-stream network. To overcome this limitation, LSTM-RNN is proposed by [49] to model long-range temporal dynamics in videos. Srivastava *et al.* [37] further formulate the video representation learning task as an autoencoder model based on the encoder and decoder LSTMs.

The aforementioned approaches are limited by treating video as a sequence of frames and optical flows for representation learning. More concretely, pixel-level temporal evolution across consecutive frames are not explored. The problem is addressed by 3D CNN proposed by Ji *et al.* [15], which directly learns spatio-temporal representation from a short video clip. Later in [38], Tran *et al.* devise a widely adopted 3D CNN, namely C3D, for learning video representation over 16-frame video clips in the context of large-scale supervised video dataset. Furthermore, performance of the 3D CNN is further boosted by inflating 2D convolutional kernels [3], decomposing 3D convolutional kernels [28, 39] and aggregated residual transformation [9].

Despite these progresses, long-range temporal dependency beyond local operation remains not fully exploited, which is the main theme of this paper. The most closely related work to this paper is [43], which investigates the non-local mean operation proposed in [2]. The work captures long-range dependency by iterative utilization of local and non-local operations. Our method is different from [43] in that local and global representations are learnt simultaneously and the interaction between them encapsulates a holistic view for the local representation. In addition, we combine the final representations from both paths for more accurate prediction.

3. Local and Global Diffusion

We start by introducing the Local and Global Diffusion (LGD) blocks for representation learning. LGD is a cell with local and global paths interacting each other. A classifier is proposed to combine local and global representations. With these, two LGD networks, namely LGD-2D and LGD-3D deriving from temporal segment networks [41] and pseudo-3D convolutional networks [28], respectively, are further detailed.

3.1. Local and Global Diffusion Blocks

Unlike the existing methods which stack the local operations to learn spatio-temporal representations, our proposed Local and Global Diffusion (LGD) model additionally integrates the global aspect into video representation learning. Specifically, we propose the novel neural networks that learn the discriminative local representation and global representation in parallel while combining them to synthesize new information. To achieve this, the feature maps in neural networks are splitted into local path and global path. Then, we define a LGD block to model the interaction between two paths as:

$$\{\mathbf{x}_l, \mathbf{g}_l\} = \mathcal{B}(\{\mathbf{x}_{l-1}, \mathbf{g}_{l-1}\}) \ , \tag{1}$$

where $\{\mathbf{x}_{l-1}, \mathbf{g}_{l-1}\}$ and $\{\mathbf{x}_l, \mathbf{g}_l\}$ denote the input pair and output pair of the l-th block. The local-global pair consists of local feature map $\mathbf{x}_l \in \mathbb{R}^{C\times T\times H\times W}$ and global feature vector $\mathbf{g}_l \in \mathbb{R}^C$, where C, T, H and W are the number of channels, temporal length, height and width of the 4D volume data, respectively.

The detailed operations inside each block $\mathcal{B}$ are shown in Figure 2 and can be decomposed into two diffusion directions as following.

(1) Global-to-local diffusion. The first direction is to learn the transformation from $\mathbf{x}_{l-1}$ to the updated local feature $\mathbf{x}_l$ with the priority of global vector $\mathbf{g}_{l-1}$. Taking the inspiration from the recent successes of Residual Learning [11], we aim to formulate the global priority as the global residual value, which can be broadcasted to each location as

$$\mathbf{x}_l = \text{ReLU}(\mathcal{F}(\mathbf{x}_{l-1}) + \mathcal{US}(\mathbf{W}^{x,g}\mathbf{g}_{l-1})) \ , \tag{2}$$

where $\mathbf{W}^{x,g} \in \mathbb{R}^{C\times C}$ is the projection matrix, $\mathcal{US}$ is the upsampling operation duplicating the residual vector to each location and $\mathcal{F}$ is a local transformation function (i.e., 3D convolutions). The choice of function $\mathcal{F}$ is dependent on the network architecture and will be discussed in Section 4.

(2) Local-to-global diffusion. The second direction is to update the global vector with current local feature $\mathbf{x}_l$. Here, we simply linearly embed the input global feature $\mathbf{g}_{l-1}$ and Global Average Pooling (GAP) of local feature $\mathcal{P}(\mathbf{x}_l)$ by

$$\mathbf{g}_l = \text{ReLU}(\mathbf{W}^{g,x}\mathcal{P}(\mathbf{x}_l) + \mathbf{W}^{g,g}\mathbf{g}_{l-1}) \ , \tag{3}$$

where $\mathbf{W}^{g,x} \in \mathbb{R}^{C\times C}$ and $\mathbf{W}^{g,g} \in \mathbb{R}^{C\times C}$ are the projection matrices combining local and global features.

Compared with the traditional convolutional block which directly apply the transformation $\mathcal{F}$ to local feature, the LGD block introduced in Eq.(2) and Eq.(3) only requires three more projection matrices to produce the output pair. In order to reduce the additional parameters for LGD block, we exploit the low-rank approximation of each projection matrix as $\mathbf{W} = \mathbf{W}_1\mathbf{W}_2$, in which $\mathbf{W}_1 \in \mathbb{R}^{C\times\hat{C}}$ and $\mathbf{W}_2 \in \mathbb{R}^{\hat{C}\times C}$. When $\hat{C} \ll C$, the parameters as well as computational cost can be sharply reduced. Through cross-validation, we empirically set $\hat{C} = \frac{C}{16}$ which is found not to impact the performance negatively. By this approximation, the number of additional parameters is reduced from $3C^2$ to $\frac{3}{8}C^2$ for each block.

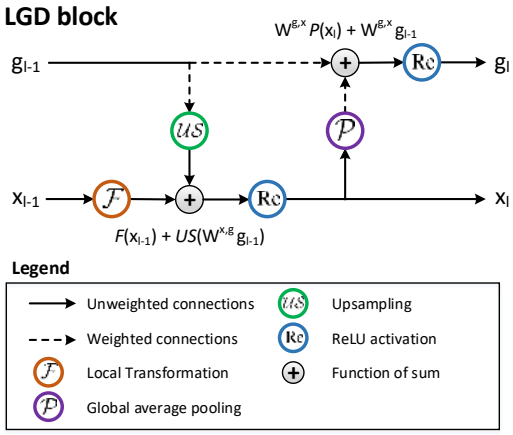


Figure 2. A diagram of a LGD block.

3.2. Local and Global Combination Classifier

With the proposed LGD block, the network can learn local and global representations in parallel. The next question is how to make the final prediction by combining the two representations. Here, we consider the kernelized view of similarity measurement between two videos. Formally, denote $\{\mathbf{x}_L, \mathbf{g}_L\}$ and $\{\mathbf{x'}_L, \mathbf{g'}_L\}$ as the last output pair of two videos, we choose the bilinear kernel [25] on both the local and global features, which can be trained end-to-end in neural network. Thus, the kernel function can be given by

$$\begin{aligned} &k(\{\mathbf{x}_L, \mathbf{g}_L\}, \{\mathbf{x'}_L, \mathbf{g'}_L\}) \\ &= \langle \mathbf{x}_L, \mathbf{x'}_L \rangle_2 + \langle \mathbf{g}_L, \mathbf{g'}_L \rangle_2 \\ &= \frac{1}{N^2}\sum_i\sum_j \langle \mathbf{x}_L^i, \mathbf{x'}_L^j \rangle_2 + \langle \mathbf{g}_L, \mathbf{g'}_L \rangle_2 \\ &\approx \frac{1}{N^2}\sum_i\sum_j \langle \varphi(\mathbf{x}_L^i), \varphi(\mathbf{x'}_L^j) \rangle + \langle \varphi(\mathbf{g}_L), \varphi(\mathbf{g'}_L) \rangle \end{aligned} \ , \tag{4}$$

in which $N = L \times H \times W$ is the number of spatio-temporal locations, $\langle \cdot, \cdot \rangle_2$ is the bilinear kernel and $\mathbf{x}_L^i \in \mathbb{R}^C$ denotes the feature vector of i-th position in $\mathbf{x}_L$. In the last line of Eq (4), we approximate the bilinear kernel by Tensor Sketch Projection φ in [6], which can effectively reduce the dimension of feature space. By decomposing the kernel function in Eq (4), the feature mapping is formulated as

$$\phi(\{\mathbf{x}_L, \mathbf{g}_L\}) = [\frac{1}{N}\sum_i \varphi(\mathbf{x}_L^i), \varphi(\mathbf{g}_L)] \ , \tag{5}$$

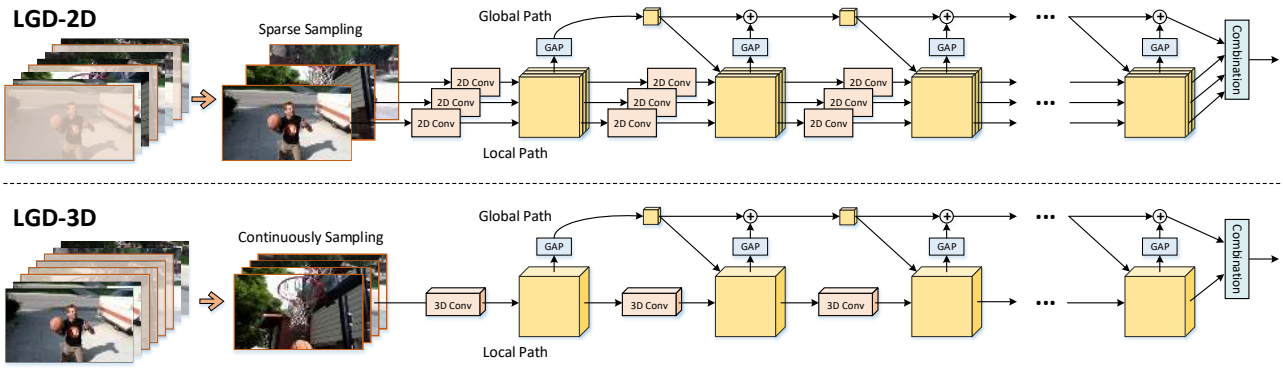


Figure 3. The overview of two different Local and Global Diffusion networks. The upper one, called LGD-2D, applies the LGD block on the temporal segment network [41], which sparsely samples several frames and exploits 2D convolution as the local transformation. The lower one, called LGD-3D, continuously samples a short video clip and exploits pseudo-3D convolution [28] as the local transformation. For both LGD networks, the learnt local and global features are combined to achieve the final representation.

where $[\cdot,\cdot]$ denotes concatenation of two vectors. The $\phi(\{\mathbf{x}_L, \mathbf{g}_L\})$ combines the pair into a high dimensional representation. The whole process can be trained end-to-end in the neural networks. Finally, the resulting representation is fed into a fully connected layer for class labels prediction.

4. Local and Global Diffusion Networks

The proposed LGD block and the classifier can be easily integrated with most of the existing video representation learning frameworks. Figure 3 shows two different constructions of LGD blocks, called LGD-2D and LGD-3D, with different transformation $\mathcal{F}$ and training strategies.

4.1. LGD-2D

The straightforward way to learn video representation directly employs 2D convolution as the transformation function $\mathcal{F}$. Thus, in the local path of LGD-2D, a shared 2D CNN is performed as backbone network on each frame independently, as shown in the upper part in Figure 3. To enable efficient end-to-end learning, we uniformly split a video into T snippets and select only one frame per snippet for processing. The idea is inspired by Temporal Segment Network (TSN) [41, 42], which overcomes computational issue by selecting a subset of frames for long-term temporal modeling. Thus, the input of LGD-2D consists of T non-continuous frames, and the global path learns a holistic representation of all these frames. Please note that the initial local representation $\mathbf{x}_1$ is achieved by a single local operation $\mathcal{F}$ applied on the input frames, and the initial global representation $\mathbf{g}_1 = \mathcal{P}(\mathbf{x}_1)$ is the global average of $\mathbf{x}_1$. At the end of the networks, the local and global combination classifier is employed to achieve a hybrid prediction.

4.2. LGD-3D

Another major branch of video representation learning is 3D CNN [15, 28, 38]. Following the common settings of 3D CNN, we feed T consecutive frames into the LGD-3D network and exploit 3D convolution as local transformation $\mathcal{F}$, as shown in the lower part in Figure 3. Nevertheless, the training of 3D CNN is computationally expensive and the model size also has a quadratic growth compared with 2D CNN. Therefore, we choose the pseudo-3D convolution proposed in [28] that decomposes 3D learning into 2D convolutions in spatial space and 1D operations in temporal dimension. To simplify the decomposition, in this paper, we only choose P3D-A block with the highest performance in [28], which cascades the the spatial convolution and temporal convolution in turn.

Here, we show the exampler architecture of LGD-3D based on the ResNet-50 [11] backbone in Table 1. The LGD-3D firstly replaces each 3×3 convolutional kernel in original ResNet-50 with one $1 \times 3 \times 3$ spatial convolution and $3 \times 1 \times 1$ temporal convolution, and then builds a LGD block based on each residual unit. All the weights of spatial convolutions can be initialized from the pre-trained ResNet-50 model as done in [28]. The dimension of input video clip is set as $16 \times 112 \times 112$ consisting of 16 consecutive frames with resolution 112×112. The clip length will be reduced twice by two max pooling layers with temporal stride of 2. The computational cost and training time thus can be effectively reduced by the small input resolution and temporal pooling. The final local representation with dimension $4 \times 7 \times 7$ is combined with global representation by the kernelized classifier. This architecture can be easily extended to ResNet-101 or deeper networks by repeating more LGD blocks.

4.3. Optimization

Next, we present the optimization of LGD networks. Considering the difficulty in training the whole network from scratch by kernelized classifier [6, 25], we propose a two-stage strategy to train the LGD networks. At the beginning of the training, we optimize the basic network without

Table 1. The detailed architecture of LGD-3D with the ResNet-50 backbone network. The LGD blocks are shown in brackets and the kernel size for each convolution is presented followed by the number of output channels.

Layer	Operation	Local path size
conv1	$1 \times 7 \times 7, 64$ $3 \times 1 \times 1, 64$, stride 1, 2, 2	$16 \times 56 \times 56$
pool1	$2 \times 1 \times 1$, max, stride 2, 1, 1	$8 \times 56 \times 56$
res2	$\begin{bmatrix} 1 \times 1 \times 1, & 64 \\ 1 \times 3 \times 3, & 64 \\ 3 \times 1 \times 1, & 64 \\ 1 \times 1 \times 1, & 256 \end{bmatrix}_{\text{LGD}} \times 3$	$8 \times 56 \times 56$
pool2	$2 \times 1 \times 1$, max, stride 2, 1, 1	$4 \times 56 \times 56$
res3	$\begin{bmatrix} 1 \times 1 \times 1, & 128 \\ 1 \times 3 \times 3, & 128 \\ 3 \times 1 \times 1, & 128 \\ 1 \times 1 \times 1, & 512 \end{bmatrix}_{\text{LGD}} \times 4$	$4 \times 28 \times 28$
res4	$\begin{bmatrix} 1 \times 1 \times 1, & 256 \\ 1 \times 3 \times 3, & 256 \\ 3 \times 1 \times 1, & 256 \\ 1 \times 1 \times 1, & 1024 \end{bmatrix}_{\text{LGD}} \times 6$	$4 \times 14 \times 14$
res5	$\begin{bmatrix} 1 \times 1 \times 1, & 512 \\ 1 \times 3 \times 3, & 512 \\ 3 \times 1 \times 1, & 512 \\ 1 \times 1 \times 1, & 2048 \end{bmatrix}_{\text{LGD}} \times 3$	$4 \times 7 \times 7$

the combination classifier, and adjust local and global representations separately. Denote $\{\mathbf{x}_L, \mathbf{g}_L\}$ and $\mathbf{y}$ as the last output pair and corresponding category of the input video, the optimization function is given as

$$\mathcal{L}_{\mathbf{W}_g}(\mathbf{g}_L,\ \mathbf{y}) + \mathcal{L}_{\mathbf{W}_x}(\mathcal{P}(\mathbf{x}_L),\ \mathbf{y})\ , \tag{6}$$

where $\mathcal{L}_{\mathbf{W}}$ denotes the softmax cross-entropy loss with projection matrix $\mathbf{W}$. The overall loss consists of the classification errors from both global representation and local representation after global average pooling. After the training of basic network, we then tune the whole network with the following loss:

$$\mathcal{L}_{\mathbf{W}_c}(\phi(\{\mathbf{x}_L, \mathbf{g}_L\}),\ \mathbf{y})\ , \tag{7}$$

where $\phi(\cdot)$ is the feature mapping proposed in Section 3.2.

5. Experiments

5.1. Datasets

We empirically evaluate LGD networks on the Kinetcis-400 [3] and Kinetcis-600 [7] datasets. The Kinetics-400 dataset is one of the large-scale action recognition benchmarks. It consists of around 300K videos from 400 action categories. The 300K videos are divided into 240K, 20K, 40K for training, validation and test sets, respectively. Each video in this dataset is 10-second short clip cropped from the raw YouTube video. Note that the labels for test set are not publicly available and the performances on Kinetics-400 dataset are all reported on the validation set. The Kinetics-600 is an extended version of Kinetics-400 dataset, firstly made public in ActivityNet Challenge 2018 [7]. It consists of around 480K videos from 600 action categories. The 480K videos are divided into 390K, 30K, 60K for training, validation and test sets, respectively. Since the labels for Kinetics-600 test set are available, we report the final performance on both the validation and test sets.

5.2. Training and Inference Strategy

Our proposal is implemented on Caffe [16] framework and the mini-batch Stochastic Gradient Descent (SGD) algorithm is employed to optimize the model. In the **training stage**, for LGD-2D, we set the input as 224×224 image which is randomly cropped from the resized 240×320 video frame. For LGD-3D, the dimension of input video clips is set as $16 \times 112 \times 112$, which is randomly cropped from the resized non-overlapping 16-frame clip with the size of $16 \times 120 \times 160$. Each frame/clip is randomly flipped along horizontal direction for data augmentation. We set each mini-batch as 128 triple frames for LGD-2D, and 64 clips for LGD-3D, which are implemented with multiple GPUs in parallel. The network parameters are optimized by standard SGD. For each stage in Section 4.3, the initial learning rate is set as 0.01, which is divided by 10 after every 20 epochs. The training is stopped after 50 epoches.

There are two **weights initialization** strategies for LGD networks. The first one is to train the whole networks from scratch. In this way, all the convolutional kernels and the projection matrices $\mathbf{W}$ in LGD block are initialized by Xavier initialization [8], and all the biases are set as zero. The second one initializes the spatial convolutions with the existing 2D CNN pre-trained on ImageNet dataset [31]. In order to keep the semantic information for these pre-trained convolutions, we set the projection matrix $\mathbf{W}^{x,g}$ as zero, making the global residual value vanishes when the training begins. Especially, the temporal convolutions in LGD-3D are initialized as an identity mapping in this case.

In the **inference stage**, we resize the video frames with the shorter side 240/120 for LGD-2D/LGD-3D, and perform spatially fully convolutional inference on the whole frame. Thus, the LGD-2D will predict one score for each triple frames and the video-level prediction score is calculated by averaging all scores from 10 uniformly sampled triple frames. Similarly, the video-level prediction score from LGD-3D is achieved by averaging all scores from 15 uniformly sampled 16-frame clips.

5.3. Evaluation of LGD block

We firstly verify the effectiveness of our proposed LGD block for spatio-temporal representation learning and compare with two diffusion block variants, i.e., block_{v1} and block_{v2} by different diffusion functions. Specifically, compared with LGD block, the $\textbf{block}_{v1}$ ignores the global representation from lower layers, making the output function

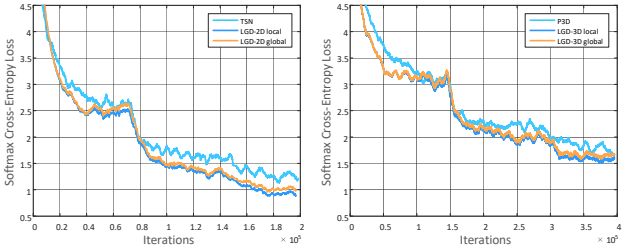
Table 2. Performance comparisons between baseline and LGD block variants on Kinetics-600 validation set. All the backbone networks are ResNet-50 trained from scratch. The local and global combination classifier is not used for fair comparison.

(a) LGD-2D

Method	Top-1	Top-5
TSN baseline	71.0	90.0
block$_{v1}$	71.6	90.2
block$_{v2}$	72.2	90.5
LGD block	**72.5**	**90.7**

(b) LGD-3D

Method	Top-1	Top-5
P3D baseline	71.2	90.5
block$_{v1}$	72.7	91.1
block$_{v2}$	73.6	91.6
LGD block	**74.2**	**92.0**



(a) LGD-2D (b) LGD-3D

Figure 4. The training loss on Kinetics-600 datasets. All the backbone networks in this figure are ResNet-50 trained from scratch.

of global path as

$$\mathbf{g}_l = \mathcal{P}(\mathbf{x}_l) \ . \tag{8}$$

Motivated by the channel-wise scaling proposed in [13], the **block**$_{v2}$ utilizes the global priority as channel-wise multiplication. Thus, the output of local path in block$_{v2}$ can be formulated as

$$\mathbf{x}_l = \text{ReLU}(\mathcal{F}(\mathbf{x}_{l-1}) \odot \mathcal{US}(sigmoid(\mathbf{W}^{x,g}\mathbf{g}_{l-1}))) \ , \tag{9}$$

where $\odot$ denotes the element-wise multiplication.

Table 2 summarizes the performance comparisons on Kinetics-600 dataset. The backbone architectures are all ResNet-50 trained from scratch. Overall, all the three diffusion blocks (i.e., LGD block, block$_{v1}$ and block$_{v2}$) exhibit better performance than baseline networks for both 2D and 3D CNNs. The results basically indicate the advantage of exploring large-scale dependency by the diffusion between local path and global path. In particular, as indicated by our results, utilizing the proposed LGD block which embeds both input local and global representations and explores the global priority as residual value, can constantly lead to better performance than block$_{v1}$ and block$_{v2}$.

The loss curves of baseline networks and LGD networks are shown in Figure 4. The training losses of local and global paths in Eq. (9) are given separately. Generally, the LGD networks produce lower losses than baseline networks, and converge faster and stably. Another observation is that the loss on local path is consistently lower than the loss on global path. We speculate that this may be due to information lost by low-rank approximation of projection matrices in Eq. (3).

Table 3. Performance contribution of each design in LGD networks. Top-1 accuracies are shown on Kinetics-600 validation set.

Method	R50	R101	Img	Com	Long	Top-1
LGD-2D	√					72.5
	√		√			74.4
	√		√	√		74.8
		√				74.5
		√	√			76.4
		√	√	√		76.7
LGD-3D	√					74.2
	√		√			75.8
	√		√	√		76.3
	√		√	√	√	79.4
		√				76.0
		√	√			77.7
		√	√	√		78.3
		√	√	√	√	**81.5**

5.4. An Ablation Study of LGD networks

Next, we study how each design in LGD networks influences the overall performance. Here, we choose ResNet-50 (**R50**) or ResNet-101 (**R101**) as backbone network. This backbone network is either trained from scratch or pre-trained by ImageNet (**Img**). The local and global combination classifier (**Com**) uses the kernelized classifier for prediction. In order to capture long-term temporal information, we further extend the LGD-3D network with 128-frame input (**Long**). Following the settings in [43], we firstly train the networks with 16-frame clips in the first stage in Section 4.3 and then with 128-frame clips in the second stage. When training with 128-frame clips, we increase the stride of pool1 layer to 4, and set each mini-batch as 16 clips to meet the requirements of GPU memory. The training is stopped after 12.5 epoches.

Table 3 details the accuracy improvement on Kinetics-600 dataset by different designs of LGD networks. When exploiting ResNet-50 as backbone network, the pre-training on ImageNet dataset successfully boosts up the top-1 accuracy from 72.5% to 74.4% for LGD-2D and from 74.2% to 75.8% for LGD-3D. This demonstrates the effectiveness of pre-training on large-scale image recognition dataset. The local and global combination classifier which combines the representations from two paths leads to the performance boost of 0.4% and 0.5% for LGD-2D and LGD-3D, respectively. Especially for LGD-3D, the training on 128-frame clips contributes a large performance increase of 3.1% by involving long-term temporal information in the network. Moreover, compared with ResNet-50, both the LGD-2D and LGD-3D based on ResNet-101 exhibit significantly better performance, with the top-1 accuracy of 76.7% and 81.5% for LGD-2D and LGD-3D, respectively. The results verify that deeper networks have larger learning capacity for spatio-temporal representation learning.

Table 4. Performance comparisons with the state-of-the-art methods on Kinetics-400 validation set.

Method	Backbone	Top-1	Top-5
I3D RGB [3]	Inception	72.1	90.3
I3D Flow [3]	Inception	65.3	86.2
I3D Two-stream [3]	Inception	75.7	92.0
ResNeXt-101 RGB [9]	custom	65.1	85.7
R(2+1)D RGB [39]	custom	74.3	91.4
R(2+1)D Flow [39]	custom	68.5	88.1
R(2+1)D Two-stream [39]	custom	75.4	91.9
NL I3D RGB [43]	ResNet-101	77.7	93.3
S3D-G RGB [46]	Inception	74.7	93.4
S3D-G Flow [46]	Inception	68.0	87.6
S3D-G Two-stream [46]	Inception	77.2	93.0
From Anet17 winner report [1]			
2D CNN RGB	Inception-ResNet-v2	73.0	90.9
Three-stream late fusion	Inception-ResNet-v2	74.9	91.6
Three-stream SATT	Inception-ResNet-v2	77.7	93.2
LGD-3D RGB	ResNet-101	79.4	94.4
LGD-3D Flow	ResNet-101	72.3	90.9
LGD-3D Two-stream	ResNet-101	**81.2**	**95.2**

5.5. Comparisons with State-of-the-Art

We compare with several state-of-the-art techniques on Kinetics-400 and Kinetics-600 datasets. The performance comparisons are summarized in tables 4 and 5, respectively. Please note that most recent works employ fusion of two or three modalities on these two datasets. Broadly, we can categorize the most common modalities into four categories, i.e., RGB, Flow, Two-stream and Three-stream. The **RGB/Flow** feeds the video frames/optical flow images into the networks. The optical flow image in this paper consists of two-direction optical flow extracted by TV-L1 algorithm [50]. The predictions from RGB and Flow modalities are fused by **Two-stream** methods. The **Three-stream** approaches further merge the prediction from audio input.

As shown in Table 4, with only RGB input, the LGD-3D achieves 79.4% top-1 accuracy, which makes the relative improvement over the recent approaches I3D [3], R(2+1)D [39], NL I3D [43] and S3D-G [46] by 10.1%, 6.8%, 2.1% and 6.2%, respectively. This accuracy is also higher than 2D CNN with a deeper backbone reported by the ActivityNet 2017 challenge winner [1]. Note that the LGD-3D with RGB input can obtain higher performance even compared with the Two-stream or Three-stream methods. When fusing the prediction from both RGB and Flow modalities, the accuracy of LGD-3D will be further improved to 81.2%, which is to-date the best published performance on Kinetics-400.

Similar results are also observed on Kinetics-600, as summarized in Table 5. Since this dataset is recently made available for ActivityNet 2018 challenge, we show the performance of different approaches reported by the challenge winner [10] and challenge runner-up [48]. With the RGB inputs, LGD-3D achieves 81.5% top-1 accuracy on Kinetics-600 validation set, which obtains 3.4% relative improvement than P3D with the deeper backbone of ResNet-152. The performance is higher than that of NL I3D which also explores large-range dependency. This result basically indicates that LGD network is an effective way to learn video representation with a global aspect. By combining the RGB and Flow modalities, the top-1 accuracy of LGD-3D achieves 83.1%, which is even higher than three-stream method proposed by ActivityNet 2018 challenge winner.

Table 5. Performance comparisons with the state-of-the-art methods on Kinetics-600. Most of the performances are reported on validation set except the performance of LGD-3D Two-stream* are on the test set.

Method	Backbone	Top-1	Top-5
From Anet18 winner report [10]			
TSN RGB	SENet-152	76.2	–
TSN Flow	SENet-152	71.3	–
StNet RGB	Inception-ResNet-v2	78.9	–
NL I3D RGB	ResNet-101	78.6	–
Three-stream Attention	mixed	82.3	96.0
Three-stream iTXN	mixed	82.4	95.8
From Anet18 runner-up report [48]			
P3D RGB	ResNet-152	78.4	93.9
P3D Flow	ResNet-152	71.0	90.0
P3D Two-stream	ResNet-152	80.9	94.9
LGD-3D RGB	ResNet-101	81.5	95.6
LGD-3D Flow	ResNet-101	75.0	92.4
LGD-3D Two-stream	ResNet-101	**83.1**	**96.2**
LGD-3D Two-stream*	ResNet-101	**82.7**	**96.0**

5.6. Evaluation on Video Representation

Here we evaluate video representation learnt by our LGD-3D for two different tasks and on four popular datasets, i.e., UCF101, HMDB51, J-HMDB and UCF101D. UCF101 [36] and HMDB51[20] are two of the most popular video action recognition benchmarks. UCF101 consists of 13K videos from 101 action categories, and HMDB51 consists of 7K videos from 51 action categories. We follow the three training/test splits provided by the dataset organisers. Each split in UCF101 includes about 9.5K training and 3.7K test videos, while a HMDB51 split contains 3.5K training and 1.5K test videos.

J-HMDB and UCF101D are two datasets for spatio-temporal action detection. J-HMDB [14] contains 928 well trimmed video clips of 21 actions. The videos are truncated to actions and the bounding box annotations are available for all frames. It provides three training/test splits for evaluation. UCF101D [36] is a subset of UCF101 for action detection task. It consists of 3K videos from 24 classes with spatio-temporal ground truths.

We first validate the global representations learnt by the pre-trained LGD-3D network. Therefore, we fine-tune the pre-trained LGD-3D on the UCF101 and HMDB51

Table 6. Performance comparisons with the state-of-the-art methods on UCF101 (3 splits) and HMDB51 (3 splits).

Method	Pretraining	U101	H51
IDT [40]	–	86.4	61.7
Two-stream [34]	ImageNet	88.0	59.4
TSN [41]	ImageNet	94.2	69.4
I3D RGB [3]	ImageNet+Kinetics-400	95.4	74.5
I3D Flow [3]	ImageNet+Kinetics-400	95.4	74.6
I3D Two-stream [3]	ImageNet+Kinetics-400	97.9	80.2
ResNeXt-101 RGB [9]	Kinetics-400	94.5	70.2
R(2+1)D RGB [39]	Kinetics-400	96.8	74.5
R(2+1)D Flow [39]	Kinetics-400	95.5	76.4
R(2+1)D Two-stream [39]	Kinetics-400	97.3	78.7
S3D-G RGB [46]	ImageNet+Kinetics-400	96.8	75.9
LGD-3D RGB	ImageNet+Kinetics-600	97.0	75.7
LGD-3D Flow	ImageNet+Kinetics-600	96.8	78.9
LGD-3D Two-stream	ImageNet+Kinetics-600	**98.2**	**80.5**

Table 7. The performance in terms of video-mAP on J-HMDB (3 splits) and UCF101D datasets.

Method	J-HMDB		UCF101D			
	0.2	0.5	0.05	0.1	0.2	0.3
Weinzaepfel *et al.* [44]	63.1	60.7	54.3	51.7	46.8	37.8
Saha *et al.* [32]	72.6	71.5	79.1	76.6	66.8	55.5
Peng *et al.* [26]	74.3	73.1	78.8	77.3	72.9	65.7
Singh *et al.* [35]	73.8	72.0	–	–	73.5	–
Kalogeiton *et al.* [17]	74.2	73.7	–	–	77.2	–
Hou *et al.* [12]	78.4	76.9	78.2	77.9	73.1	69.4
Yang *et al.* [47]	–	–	79.0	77.3	73.5	60.8
Li *et al.* [23]	82.7	81.3	82.1	81.3	77.9	71.4
LGD-3D RGB	77.3	74.2	78.8	77.6	69.3	64.1
LGD-3D Flow	84.5	82.9	86.5	84.2	79.8	74.7
LGD-3D Two-stream	**85.7**	**84.9**	**88.3**	**87.1**	**82.2**	**75.6**

datasets. The performance comparisons are summarized in Table 6. Overall, the two-stream LGD-3D achieves 98.2% on UCF101 and 80.5% on HMDB51, which consistently indicate that video representation produced by our LGD-3D attains a performance boost against baselines on action recognition task. Specifically, the two-stream LGD-3D outperforms three traditional approaches, i.e., IDT, Two-stream and TSN by 11.8%, 10.2% and 4.0% on UCF101, respectively. The results demonstrate the advantage of pre-training on large-scale video recognition dataset. Moreover, compared with recent methods pre-trained on Kinetics-400 dataset, LGD-3D still surpasses the best competitor Two-stream I3D by 0.3% on UCF101.

Next, we turn to evaluate the local representations from pre-trained LGD-3D networks on the task of spatio-temporal action detection. To build the action detection framework based on LGD-3D, we firstly obtain the action proposals in each frame by a region proposal network [30] with ResNet-101. The action tubelet is generated by proposal linking and temporally trimming in [32]. Then the prediction score of each proposal is estimated by the ROI-pooled local feature from LGD-3D network. In Table 7, we summarize the performance comparisons on J-HMDB (3 splits) and UCF101D with different IoU thresholds. Our LGD-3D achieves the best performance at all the cases. Specifically, at the standard threshold (0.5 for J-HMDB, and 0.2 for UCF101D), LGD-3D makes relative improvement of 4.4% and 5.5% than the best competitor [23] on J-HMDB and UCF101D, respectively. Figure 5 showcases four detection examples from J-HMDB and UCF101D.

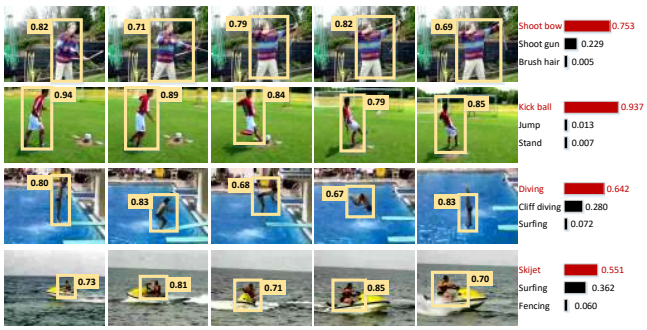


Figure 5. Four detection examples of our method from J-HMDB (upper two rows) and UCF101D (lower two rows). The proposal score is given for each bounding box. Top predicted action classes for each tubelet are on the right.

6. Conclusion

We have presented Local and Global Diffusion (LGD) network architecture which aims to learn local and global representations in an unified fashion. Particularly, we investigate the interaction between localized and holistic representations, by designing LGD block with diffusion operations to model local and global features. A kernelized classifier is also formulated to combine the final prediction from two representations. With the development of the two components, we have proposed two LGD network architectures, i.e., LGD-2D and LGD-3D, based on 2D CNN and 3D CNN, respectively. The results on large-scale Kinetics-400 and Kinetics-600 datasets validate our proposal and analysis. Similar conclusion is also drawn from the other four datasets in the context of video action recognition and spatio-temporal action detection. The spatio-temporal video representation produced by our LGD networks is not only effective but also highly generalized across datasets and tasks. Performance improvements are clearly observed when comparing to other feature learning techniques. More remarkably, we achieve new state-of-the-art performances on all the six datasets.

Our future works are as follows. First, more advanced techniques, such as attention mechanism, will be investigated in the LGD block. Second, more in-depth study of how to combine the local and global representations could be explored. Third, we will extend the LGD network to other types of inputs, e.g., audio information.

References

[1] Yunlong Bian, Chuang Gan, Xiao Liu, Fu Li, Xiang Long, Yandong Li, Heng Qi, Jie Zhou, Shilei Wen, and Yuanqing Lin. Revisiting the effectiveness of off-the-shelf temporal modeling approaches for large-scale video classification. *arXiv preprint arXiv:1708.03805*, 2017.

[2] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *CVPR*, 2005.

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[4] Ali Diba, Vivek Sharma, and Luc Van Gool. Deep temporal linear encoding networks. In *CVPR*, 2017.

[5] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016.

[6] Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. Compact bilinear pooling. In *CVPR*, 2016.

[7] Bernard Ghanem, Juan Carlos Niebles, Cees Snoek, Fabian Caba Heilbron, Humam Alwassel, Victor Escorcia, Ranjay Khrisna, Shyamal Buch, and Cuong Duc Dao. The activitynet large-scale activity recognition challenge 2018 summary. *arXiv preprint arXiv:1808.03766*, 2018.

[8] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[9] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet. In *CVPR*, 2018.

[10] Dongliang He, Fu Li, Qijie Zhao, Xiang Long, Yi Fu, and Shilei Wen. Exploiting spatial-temporal modelling and multi-modal fusion for human action recognition. *arXiv preprint arXiv:1806.10319*, 2018.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[12] Rui Hou, Chen Chen, and Mubarak Shah. Tube convolutional neural network (t-cnn) for action detection in videos. In *ICCV*, 2017.

[13] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[14] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *ICCV*, 2013.

[15] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE Trans. on PAMI*, 35(1):221–231, 2013.

[16] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, 2014.

[17] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *ICCV*, 2017.

[18] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[19] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008.

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

[21] Ivan Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

[22] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[23] Dong Li, Zhaofan Qiu, Qi Dai, Ting Yao, and Tao Mei. Recurrent tubelet proposal and recognition networks for action detection. In *ECCV*, 2018.

[24] Dong Li, Ting Yao, Lingyu Duan, Tao Mei, and Yong Rui. Unified spatio-temporal attention networks for action recognition in videos. *IEEE Trans. on MM*, 21(2):416–428, 2019.

[25] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *ICCV*, 2015.

[26] Xiaojiang Peng and Cordelia Schmid. Multi-region two-stream r-cnn for action detection. In *ECCV*, 2016.

[27] Zhaofan Qiu, Ting Yao, and Tao Mei. Deep quantization: Encoding convolutional activations with deep generative model. In *CVPR*, 2017.

[28] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 2017.

[29] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning deep spatio-temporal dependence for semantic video segmentation. *IEEE Trans. on MM*, 20(4):939–949, 2018.

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[31] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[32] Suman Saha, Gurkirt Singh, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Deep learning for detecting multiple space-time action tubes in videos. In *BMVC*, 2016.

[33] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *ACM MM*, 2007.

[34] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[35] Gurkirt Singh, Suman Saha, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *ICCV*, 2017.

[36] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human action classes from videos in the wild. *CRCV-TR-12-01*, 2012.

[37] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015.

[38] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[39] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018.

[40] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[41] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[42] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE Trans. on PAMI*, 2018.

[43] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[44] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *ICCV*, 2015.

[45] Geert Willems, Tinne Tuytelaars, and Luc Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, 2008.

[46] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018.

[47] Zhenheng Yang, Jiyang Gao, and Ram Nevatia. Spatio-temporal action detection with cascade proposal and location anticipation. In *BMVC*, 2017.

[48] Ting Yao and Xue Li. Yh technologies at activitynet challenge 2018. *arXiv preprint arXiv:1807.00686*, 2018.

[49] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015.

[50] C Zach, T Pock, and H Bischof. A duality based approach for realtime tv-l1 optical flow. *Pattern Recognition*, 2007.