Video Imprint Segmentation for Temporal Action Detection in Untrimmed Videos

Zhanning Gao,[1,2] Le Wang,[1*] Qilin Zhang,[3] Zhenxing Niu,[2] Nanning Zheng,[1] Gang Hua[4]
[1]Xi'an Jiaotong University, [2]Alibaba Group, [3]HERE Technologies, [4]Microsoft Cloud and AI

Abstract

We propose a temporal action detection by spatial segmentation framework, which simultaneously categorize actions and temporally localize action instances in untrimmed videos. The core idea is the conversion of temporal detection task into a spatial semantic segmentation task. Firstly, the video imprint representation is employed to capture the spatial/temporal interdependences within/among frames and represent them as spatial proximity in a feature space. Subsequently, the obtained imprint representation is spatially segmented by a fully convolutional network. With such segmentation labels projected back to the video space, both temporal action boundary localization and per-frame spatial annotation can be obtained simultaneously. The proposed framework is robust to variable lengths of untrimmed videos, due to the underlying fixed-size imprint representations. The efficacy of the framework is validated in two public action detection datasets.

Introduction

The prevalence of camera phones and video sharing social media has contributed to the dramatic increase of videos on the Internet. Majority of such videos may contain multiple action instances with cluttered background. Such videos are referred to as untrimmed videos, and temporal action detection on untrimmed videos has drawn significant attention recently (Oneata, Verbeek, and Schmid 2014a; Shou, Wang, and Chang 2016; Zhao et al. 2017).

Temporal action detection requires simultaneous action classification and localization of temporal boundaries, *i.e.*, the start and end frame of each action instance. Many recent methods (Shou, Wang, and Chang 2016; Singh and Cuzzolin 2016; Wang, Qiao, and Tang 2014; Dai et al. 2017; Gao, Yang, and Nevatia 2017) leverage image based object detection such as the R-CNN variants (Girshick 2015; Girshick et al. 2014; Ren et al. 2017; Xu, Das, and Saenko 2017) with the "detection by classification" scheme. The major limitation of such methods is their incapability of providing dense per-frame predictions.

To overcome such limitation, several alternative methods have been proposed, including joint prediction of the temporal boundaries and action categories without proposal generation (Lin, Zhao, and Shou 2017; Buch et al. 2017), dense per-frame labeling to refine the temporal boundaries from action proposals (Shou et al. 2017; Yeung et al. 2016). However, they all represent videos as a sequence of frames or snippets, which inevitably incurs the variable lengths input difficulty, especially for large-scale datasets.

A simpler alternative is proposed in this paper as the "temporal Detection By spatial Segmentation" (DBS) framework, which circumvents the variable lengths input difficulty. As illustrated in Figure 1, temporal action detection is recast into spatial semantic segmentation with the video imprint representation (Gao et al. 2017b; 2018), by aligning video frames into a fixed-size tensor feature. Such representation captures statistical characteristics while suppressing redundancies. In addition, video imprint representation preserves the local spatial layout across multiple frames, which justifies the spatial segmentation step with a fully convolutional network (FCN). Each segmented area corresponds to a certain action category. Finally, the segmentation score maps are reversely projected back to the video space, and converted to dense temporal prediction labels, which refine the temporal action boundaries proposals.

The video imprint representation directly assembles and aligns the convolutional neural network (CNN) features to the same fixed-size tensor maps, which allows the direct application of the same FCN segmentation network without modification. In addition, with precisely captured temporal correlations, the imprint representation keeps multiple action instances of the same category spatially proximate in the imprint representation feature map, which makes the action detection process more effective and efficient.

Another major advantage of the DBS framework is the simultaneous detection of relevant spatial regions inside each frame and per-frame temporal predictions. This is made possible due to preserved local spatial layout in the video imprint. We conduct extensive experiments to evaluate our DBS framework on two challenging datasets, *i.e.*, the THUMOS'14 (Jiang et al. 2014) dataset and the ActivityNet dataset (Heilbron et al. 2015). Experimental results show that the DBS method achieves state-of-the-art performance.

The remainder of the paper is organized as follows. We first discuss related work about temporal action detection, semantic segmentation and video imprint generation. Then,

[*]Corresponding author.

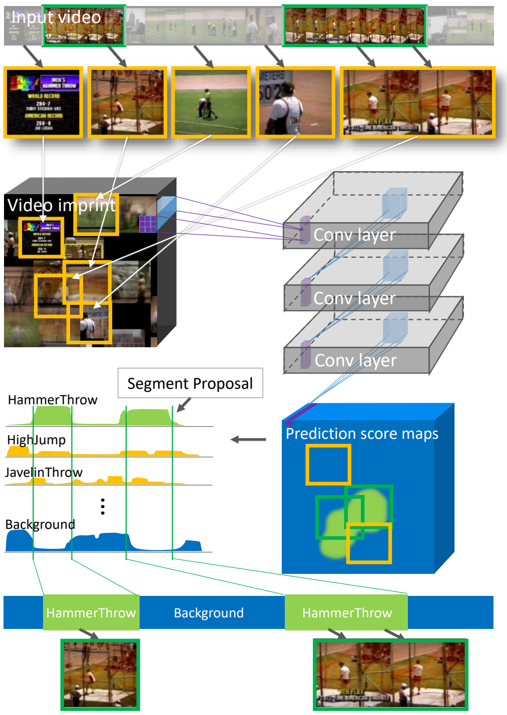

Figure 1: Overview of the temporal detection by spatial segmentation framework for temporal action detection. Based on the video imprint (Gao et al. 2017b; 2018), video frames are nonlinearly projected into the video imprint, as visualized at the bottom left "black cube". FCN-based network is utilized to segment the imprint representation and the corresponding prediction score maps are visualized as a "blue cube" at the bottom center. With the obtained segmentation prediction scores reverse projected back to the video space, the temporal action boundaries (the start and end frames) can be determined.

we present the details of our "temporal detection by spatial segmentation" framework. The evaluation results and discussions are presented in the experiment section. Finally, we conclude the paper at the last section.

Related work

Temporal action detection aims to detect the temporal boundaries and categories of the action instances in untrimmed videos. Many existing work regard the detection problem as a classification problem combined with a action proposal generation process (Jiang et al. 2014; Oneata, Verbeek, and Schmid 2014b; Singh and Cuzzolin 2016), *i.e.*, the "detection by classification" framework.

Under the "detection by classification" framework, a plenty of CNN structures have been explored to enhance the action classification, such as two-stream architectures (Simonyan and Zisserman 2014) and 3D convolutional networks (C3D) (Tran et al. 2015). S-CNN (Shou, Wang, and Chang 2016) is a multi-stage CNN based on C3D, which improves the performance by adding a localization network. Structured Segment Networks (SSN) (Zhao et al. 2017) aims to model the temporal structure by a structured temporal pyramid. These methods all need an action proposals generation step to generate action candidates. To provide more accurate proposals, there are also abundant work in the literature on generating temporal action proposals (Escorcia et al. 2016; Zhao et al. 2017; Gao et al. 2017a).

The drawbacks of the "detection by classification" framework are that the detected action boundaries are predetermined by the action proposals and only the segment-level predictions are obtainable. To overcome these limitations, Lin *et al.* (Lin, Zhao, and Shou 2017) propose a single shot action detector (SSAD) with 1D temporal convolutional layers to skip the proposal generation step. (Xu, Das, and Saenko 2017) combine the activity proposal and classification stages with a Region Convolutional 3D Network (R-C3D). Shou *et al.* (Shou et al. 2017) propose a Convolutional-De-Convolutional (CDC) network to preform dense action prediction which can generate more flexible temporal boundaries. However, most of the existing methods need to feed the videos or segmented proposals into the classifier frame by frame or snippet by snippet, which makes the inference process to be quite complicated due to the variable lengths of the videos.

Video imprint representation. To build compact video representation, the video imprint model is proposed for event analysis (Gao et al. 2017b; 2018). As a generative model, the video imprint can automatically capture the interdependences in image/frame features. Instead of combining with memory network (Sukhbaatar et al. 2015) for weakly supervised recounting task, we propose to convert the task of temporal action detection to a spatial semantic segmentation task based on the video imprint. In this way, we can handle the videos of variable lengths with a fixed-size video imprint tensor to efficiently localize and recognize the temporal action instance.

Semantic segmentation. Recently, FCN has become a standard pipeline for the image semantic segmentation task. By constructing the deep neural network with fully convolutional layers, FCN can provide pixel-wise predictions for the input image. For the semantic segmentation task, FCN and its variants have demonstrated state-of-the-art performance (Chen et al. 2016; Zhao et al. 2016; Zheng et al. 2015). Most of further improvements on FCN mainly focused on multi-scale feature learning (Zhao et al. 2016) and using CRF to refine the segmentation results (Zheng et al. 2015). We exploit the FCN to segment the video imprint so as to identify the temporal action instances. Considering the structure of the video imprint, our FCN only consists of standard convolutional layers. FCNs with more complicated structure can certainly be employed, but this is not the focus of this work.

Methods

In this section, we present the "temporal detection by spatial segmentation" framework as shown in Figure 1.

Video imprint generation

The video imprint has shown enormous potential in representing videos which contain complex events and human activities (Gao et al. 2017b; 2018). By capturing the spatial interdependence among image/frame features, it can remove the redundancy across the images/frames and preserve the local spatial layout among frames. We employ the video imprint for temporal action detection for two reasons. First, the video imprint can handle variable video lengths with a fixed-size tensor structure which can simplify the input format of the action detection system. Second, with precisely captured temporal correlations across video frames, the content consistencies are captured on the imprint map and the similar action instances are simultaneously detected by identifying the corresponding region at once, which makes the inference process more effective and efficient.

Both the tessellated counting grid (TCG) model (Perina and Jojic 2015) and epitome model (Jojic, Frey, and Kannan 2003) is adopted to generate the video imprint for event analysis(Gao et al. 2017b; 2018). The epitome has different location distributions compared with the TCG (discrete categorical versus Gaussian), and the input features of the epitome can be more flexible and the training stage is more efficient. Therefore, we employ the epitome instead of the TCG to generate the video imprint. Formally, the epitome $\mathfrak{E}$ is a set of dependent Gaussian distributions $\left\{\mathcal{N}\left(\boldsymbol{f}_{(i,j)}; \boldsymbol{\mu}_{(x,y)}, \boldsymbol{\Sigma}_{(x,y)}\right)\right\}_{(x,y)\in\mathbf{E}}$ aligned on the grid $\mathbf{E} = [0,1,\ldots,X-1] \times [0,1,\ldots,Y-1]$, where $\boldsymbol{\mu}_{(x,y)} = [\mu_{(x,y)}(0),\ldots,\mu_{(x,y)}(D-1)]^\top$ and $\boldsymbol{\Sigma}_{(x,y)} = \operatorname{diag}\left(\sigma_{(x,y)}(0),\ldots,\sigma_{(x,y)}(D-1)\right)$. Given a tensor $\mathbf{F}$, *e.g.*, CNN feature map, with dimension $W \times H \times D$, $\boldsymbol{f}_{(i,j)} = [f_{(i,j,0)},\ldots,f_{(i,j,D-1)}]^\top$ is the feature vector extracted from $\mathbf{F}$ along the spatial dimensions, *i.e.*, $(i,j) \in \mathbf{W} = [0,1,\ldots,W-1] \times [0,1,\ldots,H-1]$.

Given the epitome $\mathfrak{E} = \left\{\mathcal{N}\left(\boldsymbol{\mu}_{(x,y)}, \boldsymbol{\Sigma}_{(x,y)}\right)\right\}_{(x,y)\in\mathbf{E}}$, as a generative model, the joint distribution over the feature maps set $\{\mathbf{F}(t)\}_{t\in[0,\ldots,T]}$ and the latent window locations $\left\{\mathbf{W}_{(m,n)}(t)\right\}_{t\in[0,\ldots,T]}$ on the epitome can be derived as

$$P\left(\{\mathbf{F}(t)\}, \{\mathbf{W}_{(m,n)}(t)\}\right) \propto \prod_t \sum_{(m,n)\in\mathbf{E}} \prod_{(x,y)\in\mathbf{W}_{(m,n)}} \mathcal{N}\left(\boldsymbol{f}_{(x-m,y-n)}(t); \boldsymbol{\mu}_{(x,y)}, \boldsymbol{\Sigma}_{(x,y)}\right). \quad (1)$$

The parameters $\boldsymbol{\mu}_{(x,y)}$ and $\boldsymbol{\Sigma}_{(x,y)}$ are estimated by marginalizing the joint distribution, *i.e.*, optimizing the log likelihood of the data with an iterative EM algorithm. The E step is

$$q_{(m,n)}(t) \propto \prod_{(x,y)\in\mathbf{W}_{(m,n)}} \mathcal{N}\left(\boldsymbol{f}_{(x-m,y-n)}(t); \boldsymbol{\mu}_{(x,y)}, \boldsymbol{\Sigma}_{(x,y)}\right), \quad (2)$$

and the M step is

$$\mu_{(x,y)}(d) = \frac{\sum_t \sum_{(m,n)\in\mathbf{W}_{(x-W,y-H)}} q_{(m,n)}(t) f_{(x-m,y-n,d)}(t)}{\sum_t \sum_{(m,n)\in\mathbf{W}_{(x-W,y-H)}} q_{(m,n)}}, \quad (3)$$

$$\sigma_{(x,y)}(d) = \frac{\sum_t \sum_{(m,n)\in\mathbf{W}_{(x-W,y-H)}} q_{(m,n)}(t) h_{(x-m,y-n,d)}(t)^2}{\sum_t \sum_{(m,n)\in\mathbf{W}_{(x-W,y-H)}} q_{(m,n)}}, \quad (4)$$

where $h_{(x-m,y-n,d)}(t) = f_{(x-m,y-n,d)}(t) - \mu_{(x,y)}(d)$, $d \in [0,\ldots,D-1]$, $q_{(m,n)}(t)$ is the posterior probability $p(\mathbf{W}_{(m,n)}(t)|\mathbf{F}(t))$. In practice, we also adopt the efficient two-step scheme (Gao et al. 2018) to accelerate the learning process, and employ $\left\{\boldsymbol{\mu}_{(x,y)}\right\}_{(x,y)\in\mathbf{E}}$ as the descriptors of the video imprint.

Fully convolutional networks

The video imprint contains the local spatial layout among frames and each location may correspond to a certain part of the action instance. It is possible to map the location category to the frame-level action prediction via the posterior probability $q_{(m,n)}(t)$. This goal is quite similar to semantic segmentation, which predicts the category of each pixel of the input image. Inspired by current methods for semantic segmentation (Long, Shelhamer, and Darrell 2015), we propose to use the fully convolutional networks (FCN) to perform the temporal action detection task by segmenting the video imprint, *i.e.*, temporal detection by spatial segmentation.

A standard FCN structure is employed in our framework (see Figure 1). The grid size of video imprint is usually set to a small number ($X = Y = 24, 32, 48, 64$ in our experiment) because the CNN features have been pooled to compact high-level feature maps. Hence, the pooling layer is omitted and only the standard convolutional layer is adopted to construct the FCN structure. The influence of the FCN's architectures and parameters will be discussed in the experiment section.

Training FCN

The FCN cannot be trained directly with temporal action instance annotations. Instead, the action category should be annotated per location on the grid of training video imprints, *i.e.*, the annotation map $\mathbf{A} = [a_{(0:X-1,0:Y-1)}(0),\ldots,a_{(0:X-1,0:Y-1)}(C-1)]$, where C is number of the action categories. We utilize the posterior probability $q_{(m,n)}(t)$, which capture the distribution of each frame on the imprint grid, to convert temporal frame-level annotations of the action instances to spatial imprint-level annotations. First, the frames $\{\mathbf{I}(t)\}_{t\in[0,1,\ldots,T]}$ of the training video are grouped into distinctive sets according the frame-level annotation. Each set consists of the frames with the same action category or the background, *i.e.*, $\mathbf{T}_c = \{t|\operatorname{ann}(\mathbf{I}(t)) = c\}$ where c is the action identifier of the frame $\mathbf{I}(t)$. Then, we construct an active map for each set $\mathbf{T}_c$ to filter out the locations on the imprint where no frames with current action category are assigned, the active map is computed as

$$a_{(x,y)}(c) = \begin{cases} 1 & \text{if } (x,y) \in \mathbf{W}_{(m,n)} \text{ and } \sum_{t\in\mathbf{T}_c} q_{(m,n)} > \tau \\ 0 & \text{otherwise} \end{cases}, \quad (5)$$

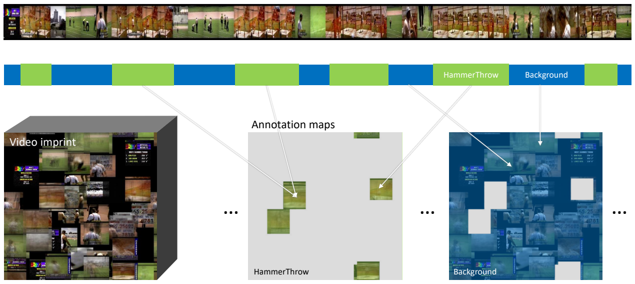

Figure 2: Illustration of the annotation maps converted from the temporal action boundaries and categories.

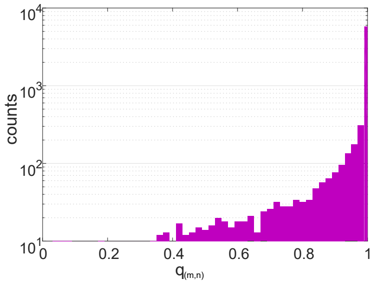

Figure 3: The statistical result of the value $q_{(m,n)}(t)$ from random selected videos. The maximum $q_{(m,n)}(t)$ from ten thousand video frames are collected. The vertical axis is shown in log-scale.

where $(x, y) \in \mathbf{E}$ and c is the identifier of the action categories. We set $\tau = 4$ according to (Gao et al. 2018).

At last, the annotation map for the video imprint is produced by concatenating the active map of each category. Figure 2 shows an example of annotation maps converted from the temporal action boundaries and categories[1]. Supervised with the annotation maps of the training video imprints, the FCN can be trained end-to-end from scratch.

Inference stage

The whole inference stage consists of three steps: First, construct the video imprint representation of the input video. Then, feed the video imprint into the FCN and output the location-level prediction score maps $\mathbf{P}$. At last, convert the prediction score maps $\mathbf{P}$ into frame-level prediction scores according to the distribution q of the input video imprint. We adopt the same strategy with (Shou et al. 2017) to generate the final temporal boundaries of the action instances. In addition, different from previous work, we can also predict the category related areas inside each frame since the local spatial layout is preserved in the video imprint.

Converting location-level prediction score maps into frame-level prediction score is quite similar to the event recounting step of (Gao et al. 2017b; 2018). The event recounting task computes the recounting map of each frame by weighted sum of the sub-windows on the weights map. The output is a heat map which indicates the importance score of each area inside the frame that related to a certain event category. In the action detection task, this spatial to temporal conversion is based on the prediction score maps $\mathbf{P}$ with multiple channels corresponding to action categories. Formally, the prediction score maps of each frame $\mathbf{p}(t)$ can be computed with

$$\mathbf{p}(t) = \sum_{(x,y)\in\mathbf{E}} q_{(x,y)}(t)\mathbf{P}_{(x,y)}, \tag{6}$$

where $\mathbf{P}_{(x,y)}$ is a tensor with size $W \times H \times C$ (C is the category number), which denotes the prediction score maps cropped from $\mathbf{P}$ in the window $\mathbf{W}_{(x,y)}$. The prediction scores for each frame are then obtained with the sum over the spatial grid of $\mathbf{p}(t)$. In practice, the computation of Equation (6) is time consuming with large t and C. However, as shown in Figure 3, we have observed that the distribution q is quite sparse, and each video frame is assigned to one of the locations on the video imprint with high probability (close to 1). Hence, instead of summing over the sub-windows, we can simply compute the $\mathbf{p}(t)$ by cropping $\mathbf{P}_{(x,y)}$ in the window $\mathbf{W}_{(x,y)}$ with highest $q_{(x,y)}$, *i.e.*, the $\mathbf{p}(t)$ is computed as

$$\mathbf{p}(t) = \mathbf{P}_{(x,y)}, \quad \text{where } (x, y) = \arg\max_{(x,y)} q_{(x,y)}(t). \tag{7}$$

Experiments

Dataset and evaluation protocol

We evaluate our method on both the per-frame labeling task and the temporal action detection task with the THUMOS'14 (Jiang et al. 2014) and ActivityNet (Heilbron et al. 2015) datasets. We also employ parts of UCF101 (Soomro, Zamir, and Shah 2012) to augment the training data.

THUMOS'14 has 1010 videos for validation and 1574 videos for testing. This dataset does not provide the training set. Following the standard practice, our method is trained on the validation set and evaluated on the testing set.

ActivityNet v1.2 contains 9682 videos in 100 classes, and ActivityNet v1.3 contains 19994 videos in 200 classes. Those videos are divided in three subsets, *i.e.*, training, validation and testing, with $2 : 1 : 1$. Since the labels of the testing set are unreleased, we present the evaluation results on the validation set for comparison.

Per-frame labeling task aims to predicting accurate labels for every frame. Following conventional metrics (Yeung et al. 2015), we treat the per-frame labeling task as a retrieval problem. All frames in the test set are ranked by their confidence scores for a certain category and the Average Precision (AP) is computed with the confidence scores for the category. Then, the mean AP (mAP) are computed by averaging the APs over all action categories.

Temporal action detection task is evaluated, following the conventions, by mean Average Precision (mAP) at different IoU thresholds. For the THUMOS'14 dataset, the IoU thresholds are $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$. The mAP at

[1] Since the video imprint is generated based on high-level CNN features, it cannot be directly visualized. For ease of illustration, we accumulate the frames on the location with the maximum $q_{(m,n)}(t)$ and draw the mean image.

0.5 IoU threshold is used for comparing results from different methods. For the ActivityNet, the IoU thresholds are $\{0.5, 0.75, 0.95\}$. The average mAP with IoU $[0.5 : 0.05 : 0.95]$ is used for comparing results from different methods.

Implementation details

CNN feature extraction. Given an input video, we sample 5 frames per second (5 fps) to extract the CNN features. Similar to (Zhao et al. 2017), we employ the two-stream CNNs models (Simonyan and Zisserman 2014) to extract both appearance and motion features of each video frame. We adopt the output from the last convolutional layer of the CNN models as the frame-level descriptors to generate the video imprint representation. Both appearance and motion based CNN features are employed to evaluate our "temporal detection by spatial segmentation" framework.

Data augmentation. Existing methods based on "detection by classification" framework (Zhao et al. 2017; Lin, Zhao, and Shou 2017) require the annotations of temporal boundaries in the training stage to compute the location loss. Only video data with action temporal boundaries can be used as training data. However, after converting the temporal boundaries to the annotation maps which implicitly encode the temporal annotation information, the FCN can be simply supervised by the annotation maps. Hence, we can use the trimmed action video data to enhance our model.

We propose two data augmentation strategies to enhance our FCN model: (a) Leverage extra trimmed action video data, *e.g.*, UCF101 (Soomro, Zamir, and Shah 2012). Since the UCF101 contains the same 20 categories with THUMOS'14, we can treat the UCF101 video as a action instance with temporal boundary covering the whole video. Thus, we can obtain the annotation map without background label [2]. (b) To emphasize the effects of the positive action instances during the training process, we crop the action instances into separate videos by their temporal annotated boundaries and treat them as individual training videos to generate the video imprints. The gains from the data augmentations are significant (see Table 3).

Training details with FCN. The FCN consists of several convolutional layers, each layer uses 3×3 kernel size and the same channels with the input video imprint. The activation units are ReLU for the mid convolutional layers and softmax for the last convilutional layer. The categorical crossentropy loss function is adopted for the training process, and the FCN is trained with the adaptive moment estimation (Adam) algorithm (Kingma and Ba 2014). The FCN can be easily trained with quick convergence due to small amount of parameters. In our experiments, the number of epoch is set to 10, and the batchsize is set to 16.

Post-processing. For the imprint descriptors in the video imprint, we apply the same post-processing method with (Gao et al. 2018), *i.e.*, first power normalized and then PCA-whitened (The feature dimension is reduced to 512 from 1024) and l_2-normalized. We employ the TAG (Zhao et al. 2017) method to generate the action proposals. Then, the action proposals are filtered according to the frame-level prediction scores and the temporal boundaries of each proposal are refined with the same strategy proposed in (Shou et al. 2017), which first performs Gaussian kernel density estimation for the frame-level prediction scores of each proposal, *i.e.*, obtain the mean $\hat{\mu}$ and standard deviation $\hat{\sigma}$, and then shrink the temporal boundaries of the proposal until reach a frame with the prediction score greater than $\hat{\mu} - \hat{\sigma}$.

Grid size (X,Y)	(24,24)	(32,32)	(48,48)	(64,64)
Imprint generation (s)	0.51	1.18	1.85	3.01
Inference (ms)	25.7	33.5	55.1	97.2
mAP	27.8	29.2	31.5	31.7

Table 1: Temporal action detection performance and the running time for video imprint generation and the FCN inference process with different size of video imprint. IoU threshold of mAP is set to 0.5.

C1:	$\mathrm{C21}$
C2:	$\mathrm{C512} - \mathrm{C21}$
C3:	$\mathrm{C512} - \mathrm{C512} - \mathrm{C21}$
C4:	$\mathrm{C512} - \mathrm{C512} - \mathrm{C512} - \mathrm{C21}$
F3:	$\mathrm{F512} - \mathrm{F512} - \mathrm{F21}$

Architecture	**C1**	**C2**	**C3**	**C4**	**F3**
mAP	27.4	29.8	31.5	30.7	29.6

Table 2: Temporal action detection performance with different architectures of FCN. "C512" denotes the convolutional layer with 3×3 kernel size and 512 output channels. "F512" denotes the convolutional layer with 1×1 kernel size.

Parameter analysis

Grid size of the video imprint. Since the video imprint is computed based on the epitome model, the window size should be the same with the input CNN feature maps (In our case, $W = H = 7$). We evaluate the performance of the action detection task with different video imprint size, *i.e.*, with $X = Y = 24, 32, 48, 64$. Table 1 shows the detection performance with different (X, Y). We also evaluate the running time for video imprint generation and the FCN inference process. We report the average GPU (Titan Xp with 12GB memory) running time with the THUMOS'14 test set (average video duration time is 230s). Compared with video imprint generation, the inference time is negligible. As a trade off between the computation efficiency and performance, we set $|\mathbf{E}| = 48 \times 48$ in the following experiments.

Architectures of FCN. Table 2 shows the action detection results with different network architectures of FCN. "C512" denotes the convolutional layer with 3×3 kernel size and 512 output channels. To evaluate the influence of the kernel size, we also explore the FCN with 1×1 kernel size (Architecture **F3**: $\mathrm{F512} - \mathrm{F512} - \mathrm{F21}$). As shown in

[2] The locations without frames assigned in the video imprint are filtered out during the training and inference process.

TH14-val	√	√	√	×	√
TH14-gt	×	√	×	√	√
UCF20	×	×	√	√	√
No. of samples	200	2.8k	2.9k	5.3k	5.5k
mAP	24.6	28.2	27.1	27.5	31.5

Table 3: The influence of data augmentation in the training stage. "TH14-val" denotes the original training data, *i.e.*, the validation set of THUMOS'14. "TH14-gt" denotes the training data generated with groundtruth action instances of THUMOS'14. "UCF20" denotes the video data extracted form UCF101 which have the same action categories with THUMOS'14. the video imprint is generated with appearance based CNN features and $X = Y = 48$. IoU threshold of mAP is set to 0.5.

IoU	0.1	0.2	0.3	0.4	0.5
RGB	50.7	48.7	44.4	38.8	31.5
Flow	50.3	48.5	45.1	38.5	31.2
R+F	56.7	54.7	50.6	43.1	34.3

Table 4: The action detection performance with different feature modalities. "RGB" denotes the appearance CNN feature. "Flow" denotes the motion CNN feature. "R+F" shows the fusion results.

Table 2, the FCN with 3 convolutional layers (**C3**) outperforms the other architectures. The convolutional operation can leverage the local spatial information to improve the action detection performance. Hence, we adopt the **C3** architecture in the following experiments.

Data augmentation strategies. We conduct an ablation study to evaluate the influence of our two data augmentation strategies. Table 3 demonstrates the action detection performance with different data augmentation strategies. "TH14-gt" and "UCF20" denote the two datasets generated with our proposed data augmentation strategies, and both can enhance the model and improve the action detection performance. An interesting finding is that only with the trimmed video data can also train our "temporal detection by spatial segmentation" framework. This result verifies the capacity of the video imprint to capture and model the temporal correlations into local spatial layout on the imprint map.

Two-stream CNN features. We employ the two-stream CNN model to extract both appearance and motion features. Table 4 shows the action detection performance with different feature modalities. We also combine these two features with later fusion strategy, *i.e.*, averaging the prediction scores from different features. It shows that the appearance and motion features both can be represented with video imprint. In addition, combining two-stream feature can further boost the action detection performance.

Methods	mAP
Single-frame CNN (Yeung et al. 2015)	34.7
Two-steam(Simonyan and Zisserman 2014)	36.2
LSTM (Donahue et al. 2015)	39.3
MultiLSTM (Yeung et al. 2015)	41.3
CDC (Shou et al. 2017)	44.4
DBS	**47.8**

Table 5: The per-frame labeling performance on THUMOS'14.

Comparison with state-of-the-art

Per-frame labeling. The per-frame labeling task aims to predict accurate labels for every frame of the input video. In Table 5, we compare our method with some state-of-the-art methods. Our method is denoted as DBS ("temporal detection by spatial segmentation"). The DBS outperforms the models based either on CNN (Simonyan and Zisserman 2014) or LSTM (Donahue et al. 2015; Yeung et al. 2015), and also the CDC (Shou et al. 2017) model which can operate on spatial and temporal dimensions simultaneously.

Temporal action detection. We compare our method with other methods on the THUMOS'14 and ActivityNet datasets. As shown in Table 6, our method achieves superior action detection results and outperforms previous work when $\mathrm{IoU} \geqslant 0.4$ on THUMOS'14 ($\mathrm{mAP} = 34.3$ with $\mathrm{IoU} = 0.5$). Table 7 and Table 8 show the temporal action detection performance on ActivityNet v1.2 and v1.3, respectively. Despite the simple and efficient inference process, our method also achieves significant improvement compared with other methods. In addition, the gain of our method is gradually increased with larger IoU threshold, *i.e.*, our method tends to produce more accurate temporal boundaries for action instances. We achieves $\mathrm{mAP@0.5IoU} = 27.8$ on ActivityNet v1.2 and $\mathrm{mAP@average} = 26.1$ on ActivityNet v1.3 which outperform existing methods. In addition, although the ActivityNet dataset holds more training videos, the annotated action instances per category is instead less than the THUMOS'14 (about 70 instances per category for ActivityNet and 300 instances for THUMOS'14). Hence, more annotated action instances per category may further boost the performance on the ActivityNet dataset.

Efficiency analysis. Due to the compact video imprint and the simple FCN architecture, our method is more memory-efficient (less than 200M) compared with CDC (Shou et al. 2017) (around 1GB) which also provide dense labeling results. As shown in Table 1, the inference is quite efficient as the CDC even combine both RGB and flow stream. For the RGB stream, we can process a 50s long video in one second, including the feature extraction step and the video imprint generation step. Since the prediction results are generated by one-time inference with the whole video, our method shall be more efficient for long untrimmed videos.

Visualization of temporal action detection and spatial prediction results. Figure 4 shows some temporal action

IoU threshold	0.1	0.2	0.3	0.4	0.5	0.6	0.7
(Richard and Gall 2016)	39.7	35.7	30.0	23.2	15.2	—	—
(Yeung et al. 2016)	48.9	44.0	36.0	26.4	17.1	—	—
(Yuan et al. 2016)	51.4	42.6	33.6	26.1	18.8	—	—
S-CNN (Shou, Wang, and Chang 2016)	47.7	43.5	36.3	28.7	19.0	10.3	5.3
CDC (Shou et al. 2017)	49.1	46.1	40.1	29.4	23.3	13.1	7.9
SSAD (Lin, Zhao, and Shou 2017)	50.1	47.8	43.0	35.0	24.6	—	—
R-C3D (Xu, Das, and Saenko 2017)	54.5	51.5	44.8	35.6	28.9	—	—
SS-TAD (Buch et al. 2017)	—	—	45.7	—	29.2	—	9.6
SSN (Zhao et al. 2017)	**66.0**	**59.4**	**51.9**	41.0	29.8	—	—
CBR-TS (Gao, Yang, and Nevatia 2017)	60.1	56.7	50.1	41.3	31.0	19.1	9.9
DBS	56.7	54.7	50.6	**43.1**	**34.3**	**24.4**	**14.7**

Table 6: The action detection performance on THUMOS’14, measured by mAP at different IoU threshold.

IoU threshold	0.5	0.75	0.95	Avg
SSN-SW (Zhao et al. 2017)	—	—	—	18.2
(Xiong et al. 2017)	41.1	24.1	5.0	24.9
SSN-TAG (Zhao et al. 2017)	—	—	—	25.9
DBS	44.0	27.5	7.4	**27.8**

Table 7: The action detection performance on validation set of ActivityNet v1.2.

IoU threshold	0.5	0.75	0.95	Avg
(Dai et al. 2017)	36.2	21.1	3.9	—
(Heilbron et al. 2017)	40.0	17.9	4.7	21.7
(Xiong et al. 2017)	39.1	23.5	5.5	24.0
DBS	43.2	25.8	6.1	**26.1**

Table 8: The action detection performance on validation set of ActivityNet v1.3.

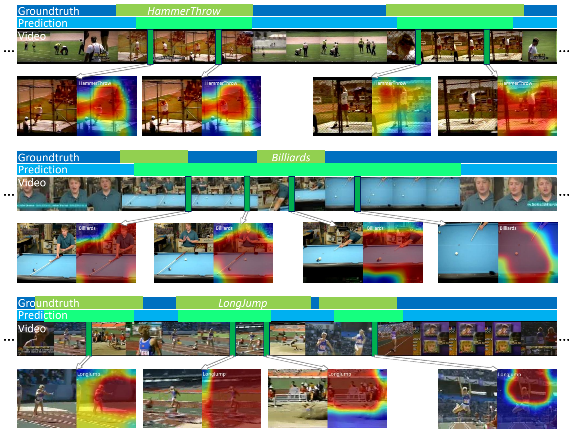

Figure 4: Visualization of temporal action detection and spatial prediction results. The heat map is used to denote the spatial prediction results which is the score map of the predicted action category.

detection results by our methods. In addition, some spatial prediction results, *i.e.*, the score map obtained with Equation (7) are also presented in Figure 4. We use heat map to denote the score map related to the predicted action category. We can see that the spatial prediction map can coarsely infer the spatial areas inside the detected frames that correlates to the predicted action category.

Conclusion and future work

We propose a temporal detection by spatial segmentation (DBS) framework for the temporal action detection task. In contrast to previous work, the DBS framework reformulate the temporal action detection problem into a semantic segmentation task on the video imprint representation. By converting all video frames into the fixed-size video imprint, long-term content consistencies and spatial interdependences among frame-level features can be captured and reflected by spatial proximity in such feature space. Subsequently, we employ an FCN to conduct efficient and effective segmentation on the video imprint representation to generate accurate temporal action detection results. The experiments show that our method achieves state-of-the-art performance for the action detection task. In addition, the specific areas inside each frame relevant to a specific action category can also be illustrated, thanks to the video imprint representation. As the potential future research, we plan to develop an end-to-end training algorithm to assistant the video imprint generation with temporal instance annotations.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0700800, in part by the National Natural Science Foundation of China under Grants 61629301, 61773312, 61503296, and 91748208.

References

Buch, S.; Escorcia, V.; Ghanem, B.; Fei-Fei, L.; and Niebles, J. 2017. End-to-end, single-stream temporal action detection in untrimmed videos. In *BMVC*.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2016. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*.

Dai, X.; Singh, B.; Zhang, G.; Davis, L. S.; and Chen, Y. Q. 2017. Temporal context network for activity localization in videos. In *ICCV*, 5727–5736.

Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; and Darrell, T. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2625–2634.

Escorcia, V.; Heilbron, F. C.; Niebles, J. C.; and Ghanem, B. 2016. Daps: Deep action proposals for action understanding. In *ECCV*, 768–784.

Gao, J.; Yang, Z.; Sun, C.; Chen, K.; and Nevatia, R. 2017a. Turn tap: Temporal unit regression network for temporal action proposals. In *ICCV*, 3648–3656.

Gao, Z.; Hua, G.; Zhang, D.; Jojic, N.; Wang, L.; Xue, J.; and Zheng, N. 2017b. ER3: A unified framework for event retrieval, recognition and recounting. In *CVPR*, 2253–2262.

Gao, Z.; Wang, L.; Jojic, N.; Niu, Z.; Zheng, N.; and Hua, G. 2018. Video imprint. *TPAMI*.

Gao, J.; Yang, Z.; and Nevatia, R. 2017. Cascaded boundary regression for temporal action detection. In *BMVC*.

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 580–587.

Girshick, R. 2015. Fast r-cnn. In *ICCV*, 1440–1448.

Heilbron, F. C.; Escorcia, V.; Ghanem, B.; and Niebles, J. C. 2015. ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*, 961–970.

Heilbron, F. C.; Barrios, W.; Escorcia, V.; and Ghanem, B. 2017. Scc: Semantic context cascade for efficient action detection. In *CVPR*, 3175–3184.

Jiang, Y.-G.; Liu, J.; Roshan Zamir, A.; Toderici, G.; Laptev, I.; Shah, M.; and Sukthankar, R. 2014. THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/.

Jojic, N.; Frey, B. J.; and Kannan, A. 2003. Epitomic analysis of appearance and shape. In *ICCV*, volume 1, 34–41.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lin, T.; Zhao, X.; and Shou, Z. 2017. Single shot temporal action detection. *arXiv preprint arXiv:1710.06236*.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*, 3431–3440.

Oneata, D.; Verbeek, J.; and Schmid, C. 2014a. Efficient action localization with approximately normalized fisher vectors. In *CVPR*, 2545–2552.

Oneata, D.; Verbeek, J.; and Schmid, C. 2014b. The LEAR submission at thumos 2014. In *ECCV Workshop*.

Perina, A., and Jojic, N. 2015. Capturing spatial interdependence in image features: the counting grid, an epitomic representation for bags of features. *TPAMI* 37(12):2374–2387.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2017. Faster r-cnn: Towards real-time object detection with region proposal networks. *TPAMI* 39(6):1137–1149.

Richard, A., and Gall, J. 2016. Temporal action detection using a statistical language model. In *CVPR*, 3131–3140.

Shou, Z.; Chan, J.; Zareian, A.; Miyazawa, K.; and Chang, S.-F. 2017. CDC: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. *arXiv preprint arXiv:1703.01515*.

Shou, Z.; Wang, D.; and Chang, S.-F. 2016. Temporal action localization in untrimmed videos via multi-stage CNNs. In *CVPR*, 1049–1058.

Simonyan, K., and Zisserman, A. 2014. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 568–576.

Singh, G., and Cuzzolin, F. 2016. Untrimmed video classification for activity detection: submission to activitynet challenge. *arXiv preprint arXiv:1607.01979*.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Sukhbaatar, S.; Weston, J.; Fergus, R.; et al. 2015. End-to-end memory networks. In *NIPS*, 2440–2448.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 4489–4497.

Wang, L.; Qiao, Y.; and Tang, X. 2014. Action recognition and detection by combining motion and appearance features. In *ECCV Workshop*.

Xiong, Y.; Zhao, Y.; Wang, L.; Lin, D.; and Tang, X. 2017. A pursuit of temporal accuracy in general activity detection. *arXiv preprint arXiv:1703.02716*.

Xu, H.; Das, A.; and Saenko, K. 2017. R-c3d: region convolutional 3d network for temporal activity detection. In *ICCV*, 5794–5803.

Yeung, S.; Russakovsky, O.; Jin, N.; Andriluka, M.; Mori, G.; and Fei-Fei, L. 2015. Every moment counts: Dense detailed labeling of actions in complex videos. *IJCV* 1–15.

Yeung, S.; Russakovsky, O.; Mori, G.; and Fei-Fei, L. 2016. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, 2678–2687.

Yuan, J.; Ni, B.; Yang, X.; and Kassim, A. A. 2016. Temporal action localization with pyramid of score distribution features. In *CVPR*, 3093–3102.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2016. Pyramid scene parsing network. *arXiv preprint arXiv:1612.01105*.

Zhao, Y.; Xiong, Y.; Wang, L.; Wu, Z.; Tang, X.; and Lin, D. 2017. Temporal action detection with structured segment networks. In *ICCV*.

Zheng, S.; Jayasumana, S.; Romera-Paredes, B.; Vineet, V.; Su, Z.; Du, D.; Huang, C.; and Torr, P. H. 2015. Conditional random fields as recurrent neural networks. In *ICCV*, 1529–1537.