A New Temporal Deconvolutional Pyramid Network for Action Detection

Xiangli Ji, Guibo Luo, and Yuesheng Zhu(✉)

Communication and Information Security Laboratory, Shenzhen Graduate School,
Peking University, Shenzhen 518055, China
{Jxiangli,luoguibo,zhuys}@pku.edu.cn

Abstract. Temporal action detection is a challenging task for detecting various action instances in untrimmed videos. Existing detection approaches are unable to localize the start and end time of action instances precisely. To address this issue, we propose a novel Temporal Deconvolutional Pyramid Network (TDPN), in which a Temporal Deconvolution Fusion (TDF) module in each pyramidal hierarchy is developed to construct strong semantic features of multiple temporal scales for detecting action instances with various durations. In the TDF module, the temporal resolution of high-level feature is expanded by a temporal deconvolution. The expanded high-level features and low-level features are fused by a fusion strategy to form strong semantic features. The fused semantic features with multiple temporal scales are used to predict action categories and boundary offsets simultaneously, which significantly improves the detection performance. Besides, a strict strategy for assigning label is proposed during training to improve the precision of temporal boundaries learned by model. We evaluate our detection approach on two public datasets, *i.e.*, THUMOS14 and MEXaction2. The experimental results have demonstrated that our TDPN model can achieve competitive performance on THUMOS14 and best performance on MEXaction2 compared with the other approaches.

Keywords: Action detection · Untrimmed videos · TDPN network

1 Introduction

Temporal action detection has numerous potential applications in video surveillance, video content analysis and intelligent home care. This task is to detect action instances in untrimmed videos, which needs to output the action categories and the precise start and end time. Since there is high variability in

Y. Zhu—This work was supported in part by the Shenzhen Municipal Development and Reform Commission (Disciplinary Development Program for Data Science and Intelligent Computing); and by Shenzhen International cooperative research projects GJHZ20170313150021171.

C. V. Jawahar et al. (Eds.): ACCV 2018, LNCS 11364, pp. 696–711, 2019.
https://doi.org/10.1007/978-3-030-20870-7_43

the duration of action from arbitrarily long video, temporal action detection is substantially challenging.

In recent years, some progress has been made in temporal action detection [3,16,22,31,33,34]. Many works regard this task as a classification problem, which contains candidate generation stage and classification stage. Earlier attempts [4,33] in temporal action detection adopt sliding windows as candidates and design hand-crafted features for classification, which is computationally expensive. Inspired by progress in image object detection [20], many approaches [22,31,34] adopt the "Proposal + Classification" framework, where a classifier is used to classify action instances generated by proposal methods [6,9]. However, there are some major drawbacks in these approaches. First, the process of proposal generation requires additional time and space costs. Second, deep convolutional features with fixed temporal resolution are used to detect action instances with various temporal scales, which limits the detection performance of these methods. Inspired by unified models [19,29] in object detection, SSAD network [16] and SS-TAD [12] network completely eliminate action proposal generation stage and predict temporal boundaries and specific action categories simultaneously. Although these approaches have a fast speed for detecting actions, the accuracy of detected temporal boundaries is still unsatisfied. For the SS-TAD [12] network, the used recurrent memory modules have a limit span of temporal attention leading to imprecise temporal boundaries. For the SSAD network [16], multiscale features are extracted by temporal convolution to detect actions, yet these features are temporally coarser, so that it cannot localize the start and end of action instances precisely. Besides, the detection performance of SSAD network is dependent heavily on feature extractor since final action classes are obtained by fusing predicted class scores and snippet-level action scores from feature extractor.

To address these issues, we propose a new Temporal Deconvolutional Pyramid Network (TDPN), which adopts Temporal Deconvolution Fusion (TDF) modules in various pyramidal hierarchies to integrate strong semantic information into feature maps with multiple temporal scales. Inspired by FPN [15] network in object detection, we introduce a top-down pathway with lateral connection into SSAD [16] network to extract temporal feature pyramid. Note that it is non-trivial to adapt the top-down architecture from object detection to temporal action detection, which needs to be designed to efficiently deal with temporal features. Different from FPN [15] network, our TDF module in top-down pathway adopts temporal deconvolution to expand temporal resolution of feature maps. To further improve detection performance, we investigate different fusion strategies for features from different pyramid hierarchies in the top-down pathway. Compared to SSAD [16] network, the fused feature sequences with same temporal resolution contains stronger semantics and more context information, which can significantly improve detection performance. The fused semantic features with multiple temporal scales are used to predict action class scores and boundary offsets simultaneously. The post-processing step of TDPN model is simple and only Non-Maximum Suppression (NMS) is used to remove repeatable detection results.

Our main contribution is the proposal of a new Temporal Deconvolutional Pyramid Network for temporal action detection that is eminent in the following aspects: (1) The TDPN model constructs strong semantic features with multiple temporal resolution by using TDF modules in various pyramidal hierarchies to detect action instances, which significantly improves detection performance. (2) Our TDPN model can learn precise temporal boundaries of action instances by using a fusion strategy for features from different pyramid hierarchies and a strict strategy for assigning label during training. (3) Our TDPN model achieves competitive performance (mAP@tIoU = 0.5 of 40.7%) on THUMOS14 dataset and outperforms other approaches on MEXaction2 dataset by increasing mAP@tIoU = 0.5 from 11.0% to 22.1%.

2 Related Work

Action Recognition. Over the past several years, great progress has been made in action recognition task. Earlier work mainly focuses on hand-crafted features, such as space-time interest points (STIP) [14] and Improved Dense Trajectory (iDT) [26]. With the remarkable performance obtained from deep neural network in image analysis, many methods adopt features extracted from deep neural networks [23,24]. Two-stream architecture [23] is proposed, which adopts two convolutional neural networks (CNNs) to extract appearance and motion features from single frame and stacked optical flow field respectively. 3D-CNN [24] learns appearance and motion features directly from frame volumes using 3D convolutional filters, which is more efficient than two-stream network. As a upgraded task, temporal action detection usually adopts action recognition models to extract spatiotemporal features. In our TDPN model, a deep two-steam network is used as feature extractor.

Object Detection. According to the used detection framework, object detection methods can be broadly divided into two categories. One is “detect by classifying object region proposals” framework, including region proposal generation stage and classification stage, such as Faster R-CNN [20] and Feature Pyramid Network (FPN) [15]. The region proposals are generated by some methods, such as SelectiveSearch [25] and RPN [20], and then classification network predicts object categories and location offsets of region proposals. The other one is unified detection framework, which skips proposal generation step and encapsulates all computation in a single network. Typical networks of this framework are YOLO [19] and SSD [29]. Object detection focuses on regions of interest in images, yet temporal action detection requires to combine temporal information to detect actions of interest in videos. Different from the FPN [15] network, our TDPN network adopts temporal deconvolution to deal with multiscale temporal features.

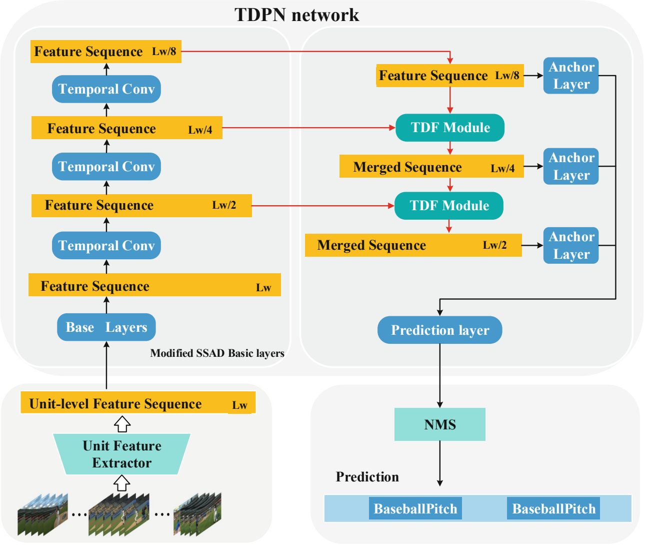


Fig. 1. Framework of our approach: the whole framework includes feature extractor, the TDPN network and the post-processing. Our TDPN network consists of modified SSAD basic layers, Temporal Deconvolution Fusion (TDF) modules, anchor layers and prediction layers. NMS is used to filter out duplicate detection results

Temporal Action Detection. Affected by the object detection methods, temporal action detection approaches are also broadly divided into "proposals by classification" framework and unified framework. For the previous framework, many methods for generating action proposals have been proposed, such as Sparse-prop [9] and TURN [6]. In the classification process, earlier works [11,17,33] mainly use hand-crafted features to classify action proposals. Recently, many approaches extract appearance and motion features using deep neural network [31,34,35] to detect action instances, which improves detection performance. Segment-CNN [35] proposes three segment-based 3D ConNets for detecting action: proposal network, classification network and localization network. Structured segment network (SSN) [34] models temporal structure of action instances via a structured temporal pyramid and a decomposed discriminative model. Based on Faster R-CNN framework, recently proposed TAL-Net [3] network improves receptive field alignment by a multi-tower network and exploits temporal context by extending receptive fields. Different from TAL-Net network, our TDPN model constructs temporal feature pyramid by using temporal convolution and deconvolution for detecting action instances directly without proposal generation process. UntrimmedNet [27] is a weakly supervised architecture, in which classification module and selection module are developed to learn action models and detect action instances respectively. Recurrent neural networks (RNNs) are also used to learn temporal information in many action detection methods [32,33]. The unified framework eliminates the proposal generation

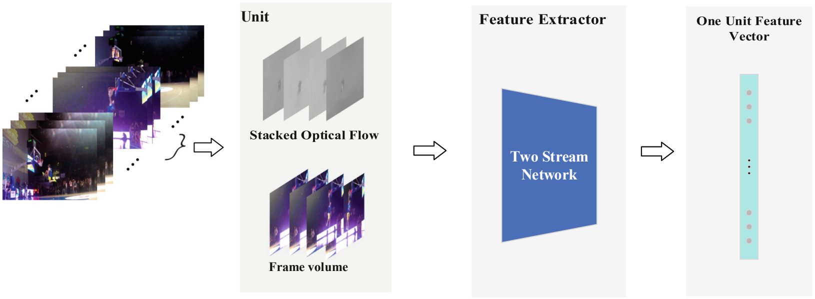


Fig. 2. Unit-level feature extraction. A untrimmed video is divided into units which consists of frame volume and stacked optical flow and unit-level feature sequences are extracted by two-steam network as the input of our TDPN model

stage and predict action categories and location offsets simultaneously, such as SSAD [16] and SS-TAD [12]. Our TDPN model also adopts the unified framework to detect action instances in untrimmed videos.

3 Approach

In this section, we introduce our Temporal Deconvolution Pyramid Network (TDPN) and the training procedure. As shown in Fig. 1, the whole architecture of system consists of feature extractor, the TDPN network and the post-processing. Initial feature pyramid is obtained by temporal convolution in modified SSAD basic layers, and a new Temporal Deconvolution Fusion (TDF) module is developed to extract strong semantic features with multiple temporal resolution. In the post-processing, NMS is used to remove repeatable results to obtain final detection results. We will describe each component and training procedure in details.

3.1 Video Unit Feature Extraction

We adopt deep two-stream network [30] to extract feature sequences as the input of TDPN model, where spatial CNN network uses ResNet [8] model and temporal CNN network adopts BN-Inception model [10]. A untrimmed video V is divided into T_v/m_u consecutive units where m_u is the number of frames in a unit and T_v is the number of frames in V. We pad the video in tail with last frame so that each unit has the same length m_u. Note that units are not overlapped with each other. As shown in Fig. 2, a frame volume with 8 frames is sampled uniformly from a unit, which is fed into the spatial network to extract features of "Flatten_673" layer. We compute the mean of these 8 feature vectors as unit-level appearance feature. Stacked optical flow field is calculated from 6 consecutive frames at the center of a unit, which is fed into the temporal network to extract motion feature of "global pool" layer. The appearance feature vector and the motion feature vector are concatenated as the feature vector of a unit.

Given a untrimmed video V, we can extract a unit-level feature sequence. Since the length of video is various, we use a sliding window with fixed length to divide the feature sequence into segments as the input of the TDPN model.

3.2 TDPN Network

We propose the TDPN model, which consists of modified SSAD basic layers, Temporal Deconvolution Fusion (TDF) modules, anchor layers and prediction layers. A key innovation is to strengthen semantics of multiscale features by using TDF modules in the top-down pathway, which improves significantly detection performance. The post-processing step of TDPN model is simple and only NMS is used to filter out duplicate detection results.

Modified SSAD Basic Layers. In the bottom-up pathway of our TDPN model, the modified SSAD basic layers is used, which consists of base layers and three temporal convolutional layers, as shown in Fig. 1.

The original base layers of SSAD [16] network contains two temporal convolutional layers for increasing the size of receptive fields and two temporal max pooling layers for shortening feature maps. Note that the max pooling layers are removed in the base layers of TDPN model since the temporal length of input features needs to remain unchanged in these layers. Then three temporal convolutional layers are stacked to extract initial feature maps with multiple temporal resolution. We denote the feature map of l-th temporal convolutional layer in the bottom-up pathway as $F_E^l \in R^{L_l \times T_l}$. The output feature maps of these convolutional layers are F_E^1, F_E^2 and F_E^3 with size $L_w/2 \times 512$, $L_w/4 \times 512$ and $L_w/8 \times 512$ respectively.

Temporal Deconvolution Fusion Module. As shown in Fig. 3, Temporal Deconvolution Fusion (TDF) module in each pyramid hierarchy of the top-down pathway is comprised of temporal deconvolutional layers, fusion sub-module, temporal convolutional layers and batch normalization layers, which is used to fuse semantically strong, temporally coarser features from top-down pathway and temporally fine features from bottom-up pathway.

The temporal deconvolution is actually the transpose of temporal convolution rather than an actual deconvolution, which is a convolutional form of sparse coding. Filters of the temporal deconvolutional layers can be parameterized by weights $W_l \in R^{T_l \times d \times T_{l-1}}$ and biases $b_l \in R^{T_l}$, where d is the duration of filters, T_l and T_{l-1} respectively indicate the number of filters in the l-th and $(l-1)$-th deconvolutional layer. The output vector E_t of the l-th deconvolutional layer at time step t is

$$E_t^l = f(\sum_{t'=1}^{d} \left\langle W_{t'}^l, E_{t+d-t'}^{l-1} \right\rangle + b^l) \ , \tag{1}$$

Where $f(\bullet)$ is the activation function, $E_{t+d-t'}^{l-1}$ is the activation vector of the $(l-1)$-th deconvolutional layer at the time step $(t+d-t')$.

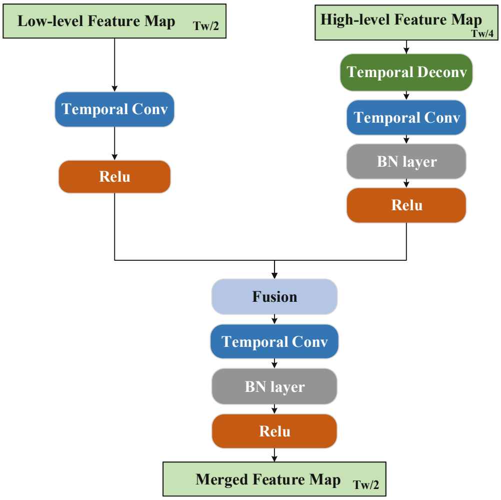


Fig. 3. TDF module. Temporal deconvolution is used to expand temporal dimension of high-level, strong semantic feature maps from top-down pathway. Low-level feature maps with high temporal resolution come from the bottom-up pathway. Fusion sub-module is adopted to fuse high-level features and low-level features

Then a temporal convolutional layer is added to further expand the temporal receptive field and a batch normalization layer is adopted to speed up training and further improve detection accuracy. With these layers, the temporal dimension of high-level feature map from top-down pathway is doubled. We adopt 1×1 convolution to match the number of channels in low-level features from bottom-up pathway. The fusion sub-module is used to fuse the high-level feature maps with expanded temporal dimension and low-level feature maps from the bottom-up pathway. To further improve detection performance, we explore different fusion methods, including element-wise sum, element-wise mean and channel concatenation.

Two TDF modules are stacked in the top-down pathway, which output feature maps F_D^1 and F_D^2 with size $L_w/2 \times 512$ and $L_w/4 \times 512$ respectively. Together with the output of the last layer $F_E^3 \in R^{L_w/8 \times 512}$ in the bottom-up pathway, the fused feature maps are used to predict action class scores and boundary offsets simultaneously.

Anchor and Prediction Layers. We use three anchor layers composed of temporal convolution to process the fused feature maps with multiple temporal scales from the top-down pathway. In each feature map of anchor layers, each temporal location is associated with K anchor instances with different scales. The scale ratios of anchors are the same as the ones used in SSAD [16] network, as shown in Table 1. When the length of video unit m_u is 16, the strides of these feature sequences are 32, 64, 128 frames respectively. The temporal scale ranges

Table 1. The anchor settings for feature sequences with different temporal resolutions in the proposed TDPN model.

Feature maps	Strides	Anchor scale ratios	Temporal scale ranges
F_D^1	32	$(1, 1.5, 2)$	32–64
F_D^2	64	$(0.5, 0.75, 1, 1.5, 2)$	32–128
F_D^3	128	$(0.5, 0.75, 1, 1.5, 2)$	64–256

of these anchor instances are 32–64, 32–128 and 64–256 frames, respectively. An important reason why the performance of our TDPN model is better than SSAD model is that the anchor instances with same temporal scales contains context information and strong semantics.

In the prediction layer, temporal convolution is adopted to predict action categories probabilities, boundaries offsets and overlap scores simultaneously. Each level of the temporal feature pyramid corresponds to a prediction layer and parameters are not shared in these layers. Similar to SSAD [16], classification scores are obtained by softmax layer and overlap scores are normalized by sigmoid function. Finally, we use NMS to remove duplicate results to obtain final detection results. The threshold in NMS is set to 0.1 by empirical validation.

3.3 Training

Label Assignment. We propose a new strategy for assigning action label to the detected action instances during training. Given a window w_i, g_i is the corresponding ground truth instances, including action categories, the starting and ending frames of action instances. Based on the time intersection-over-union (tIoU), we assign an activity label to a predicted anchor instance (1) if the highest one among the tIoUs with all ground truth instances g_i is higher than a threshold σ; (2) if it has the highest tIoU for a given ground truth instance. Note that the priority of the first case is higher than the second to avoid a predicted anchor instance being assigned multiple activity labels. When these conditions are not satisfied, it will be assigned a background label. The threshold σ is set to 0.7 by empirical validation. Our TDPN model can learn precise temporal boundaries of action instances by using this strict strategy for assigning label during training.

Optimization. Temporal action detection is a multi-task problem, including regression and classification tasks. To train the TDPN model, we need to optimize both regression and classification tasks jointly. The objective loss function is a weighted sum of the softmax loss and the smooth L_1 loss, which is given by:

$$Loss = L_{soft\max} + \lambda L_{reg} + L_2(\Theta) \ , \tag{2}$$

where $L_{soft\max}$ is a standard multi-class softmax loss function; L_{reg} is smooth L_1 loss; $L_2(\Theta)$ is the L_2 regularization loss; Θ represents the parameters of the TDPN model; λ is a loss trade-off parameter and is set to 2.

L_{reg} is defined as

$$L_{reg} = \frac{1}{N} \sum_{i=1}^{N} \sum_{z=1}^{C} l_i^z [R(\varDelta \hat{c}_i^z - \varDelta c_i^z) + R(\varDelta \hat{w}_i^z - \varDelta w_i^z)], \tag{3}$$

where $\varDelta c$ and $\varDelta w$ are location transformations of ground truth instances; $\varDelta \hat{c}$ and $\varDelta \hat{w}$ are the same for predicted action instances; N is the number of training samples in the mini-batch; C denotes the number of classes; R is the L_1 distance; $l_i^z = 1$ when the true class label of the i-th instance is z, otherwise, $l_i^z = 0$.

4 Experiments

In this section, the experiments are conducted to evaluate the performance of our proposed TDPN model on two detection benchmark datasets: THUMOS14 [13] and MEXaction2 [1].

4.1 Evaluation Datasets

THUMOS14 [13]**.** The whole dataset contains 1010 untrimmed videos for validation and 1574 untrimmed videos for testing. In the temporal action detection task, 200 validation set videos and 213 test set videos have temporal annotations in 20 action categories. This dataset is particularly challenging as many action instances have very short duration in pretty long videos. This dataset dose not provide the training set by itself, so the UCF-101 dataset including 13320 trimmed videos is appointed as the official training set. Following the practices, we only use its validation set for training and the test set for evaluating the TDPN network. Note that we remove two falsely annotated videos ("270", "1496") in the test set.

MEXaction2 [1]**.** This dataset contains three subsets: INA videos, YouTube clips and UCF101 Horse Riding clips. In the Mexaction2 dataset, only two action classes are annotated temporally: "BullChargeCape" and "HorseRiding". INA videos consist of 117 untrimmed videos with approximately 77 h in total, which are divided into three parts: training, parameter validation and testing. YouTube clips and UCF101 Horse Riding clips are trimmed and each clip contains one action instance, which are only used for training. Although this dataset only contains two action categories, it is particularly challenging in temporal action detection task. There are high imbalance between background and action instance of interest and high variability in point of view, image quality and action duration for "HorseRiding" in the MEXaction2 dataset.

Evaluation Metrics. We follow the convention metrics used in the THUMOS14, which use mean Average Precision (mAP) at different tIoU thresholds for evaluation and calculate Average Precision (AP) for each activity categories.

Table 2. Ablation study of TDF module in the proposed TDPN model. "SP" denotes that the network has single pathway and "TP" denotes that the network has two contrary pathway.

tIoU	0.1	0.2	0.3	0.4	0.5
SSAD [16]	50.1	47.8	43.0	35.0	24.6
re-trained SSAD (SP)	54.6	52.1	47.7	41.1	31.1
Our TDPN (TP)	58.6	56.3	51.8	44.4	35.0

On THUMOS14, the mAP with tIoU thresholds $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ is used for comparing the performance of different methods and the AP at 0.5 tIoU is used for each activity category. On the MEXaction2 dataset, the mAP at 0.5 tIoU is used to compare performance of different approaches.

4.2 Implementation Details

During training, the length of sliding window is L_w and the stride size is $\frac{1}{4}L_w$. Note that the stride size is $\frac{3}{4}L_w$ during prediction. The windows of training data should contain at least one ground truth instance and the windows without ground truth instances will be held out from training. L_w is set to 32 by empirical validation. The batch size is 32 and each mini-batch is constructed from one window. To make the TDPN model converge fast, we randomly shuffle the order of training data. Similar to SSAD [16], the hard negative mining strategy is adopted to balance the proportion of positive and negative samples. Since there is no suitable pre-trained temporal deconvolutional network, the parameters of whole TDPN model are randomly initialized by the Xavier method [7]. The learning rate is initially set to 10^{-4} and then reduced by a factor of 10 after every 30 epochs. Training is terminated after a maximum of 60 epochs. We implement our system using Tensorflow [2], with training executed on a machine with 32G memory, NVIDIA Titan Xp GPU and Intel i7-6700K processor.

4.3 Ablation Study

Two Pathways vs Single Pathway. Our TDPN model contains two contrary pathways by introducing a top-down pathway into SSAD [16] network. The main strength of our TDPN model is that the features constructed by TDF modules in the top-down pathway contain more context information and stronger semantics than the ones with same temporal resolution in SSAD network. To compare fairly with SSAD network, we re-train the SSAD network using the unit-level feature sequences as our baseline model. The used strategy for label assignment is same as the one proposed in SSAD [16]. The element-wise sum is chosen as the fusion method in TDF module, which provides the best performance (See Table 3 bottom sections). Table 2 lists the detection results of our baseline model and TDPN model on THUMOS14, which shows that the mAP at 0.5

Table 3. Ablation study of methods for fusing feature sequences.

Fusion methods	mAP ($\theta = 0.5$)
Eltw-mean	37.9
Channel concatenate	38.8
Eltw-sum	40.7

Table 4. Ablation study of strategy for label assignment. "LA1" denotes the strategy for assigning label used in SSAD [16]. "LA2" denotes the strategy proposed in this paper.

tIoU	0.1	0.2	0.3	0.4	0.5
TDPN (LA1)	58.6	56.3	51.8	44.4	35.0
TDPN (LA2)	63.1	61.1	56.7	49.9	40.7

tIoU of the re-trained SSAD model increases from 24.6% to 31.1%. Compared to the SSAD model with single pathway, the mAP at 0.5 tIoU of our TDPN model increases from 31.1% to 35.0% (about 3.9% improvement). From these results, we can get two conclusions: (1) Strong semantic features with multiple temporal resolution constructed by TDF modules in the top-down pathway can significantly improve detection performance. (2) The unit-level feature sequences are effective to represent the spatiotemporal characteristics of actions.

Methods for Fusing Feature Sequences. We explore the different fusion methods in the TDF module of our TDPN model, including element-wise sum, element-wise mean and channel concatenation. During training, we use the strategy for label assignment proposed in Sect. 3.3 to train our TDPN model. The mAP at 0.5 tIoU is adopted to compare different methods on THUMOS14. As shown in Table 3, the element-wise sum method achieves best performance among these methods (the 40.7% mAP at 0.5 tIoU). Therefore, element-wise sum can effectively combine high-level feature maps with coarser temporal resolution and low-level feature maps with fine temporal resolution, which improves the performance of the TDPN detector.

Strategies for Label Assignment. Here, we evaluate that the impact of label assignment on detection performance on THUMOS14. "LA1" denotes the strategy for label assignment proposed in SSAD [16], where a predicted action instance is assigned the corresponding activity label if its highest tIoU with all ground truth instances is higher than 0.5. "LA2" denotes that the strategy for label assignment discussed in the Sect. 3.3. We use these strategies to train our TDPN model respectively. Action detection results are measured by mAP of different tIoU thresholds. As shown in Table 4, the strategy proposed in this paper achieves the best performance (about 5.7% improvement of the mAP at

Table 5. Action detection results on THUMOS14 test dataset, measured by mAP at different tIoU thresholds.

tIoU	0.1	0.2	0.3	0.4	0.5
Wang *et al.* [28]	18.2	17.0	14.0	11.7	8.3
Oneata *et al.* [18]	36.6	33.6	27.0	20.7	14.4
SLM [21]	39.7	35.7	30.0	23.2	15.2
FG [32]	48.9	44.0	36.0	26.4	17.1
PSDF [33]	51.4	42.6	33.6	26.1	18.8
S-CNN [35]	47.7	43.5	36.3	28.7	19.0
CDC [22]	-	-	40.1	29.4	23.3
SSAD [16]	50.1	47.8	43.0	35.0	24.6
TURN [6]	54.0	50.9	44.1	34.9	25.6
RC3D [31]	54.5	51.5	44.8	35.6	28.9
CBR [5]	60.1	56.7	50.1	41.3	31.0
TAL-Net [3]	59.8	57.1	53.2	48.5	**42.8**
SSAD (re-trained)	54.6	52.1	47.7	41.1	31.1
TDPN (ours)	**63.1**	**61.1**	**56.7**	**49.9**	40.7

0.5 tIoU) compared with the strategy used in SSAD [16]. Our proposed strategy increases the number of positive samples with short duration that are detected hard during training. These results demonstrate that our strict strategy for label assignment can improve the precision of temporal boundaries learned by model.

From the above comparisons, we adopt the unit-level feature sequences as the input of our TDPN model, the fused features with multiple temporal resolutions for detecting action instances and strict label assignment for training. Next, the TDPN model will be compared with other state-of-the-art approaches.

4.4 Comparison with the State of the Art

On the THUMOS14 and MEXaction2 datasets, we compare our TDPN model with existing state-of-the-art approaches, and using the matrix mentioned above reports detection performance. In our experiments, we set the unit length and window length to 16 and 32 respectively and use element-wise sum method to fuse features in the TDF module.

THUMOS14. In the last row of Table 5, our TDPN model shows about 9.6% improvement at the mAP@0.5 over our re-trained SSAD network, which indicates the importance of exploiting the feature maps with the high temporal resolution and strong semantics. Moreover, we compare the TDPN model with state-of-the-art approaches and the results during challenge [18, 28]. As shown in Table 5, when the tIoU threshold is less than 0.5, our TDPN model outperforms the prior state-of-the-art approaches, including Cascaded Boundary Regression

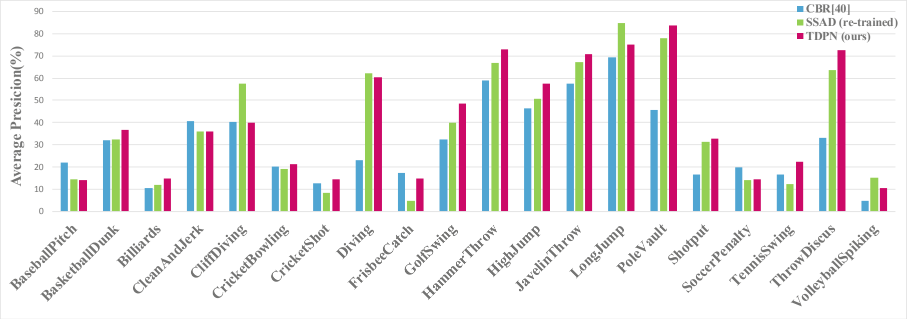


Fig. 4. Per-class AP at 0.5 tIoU threshold on THUMOS14

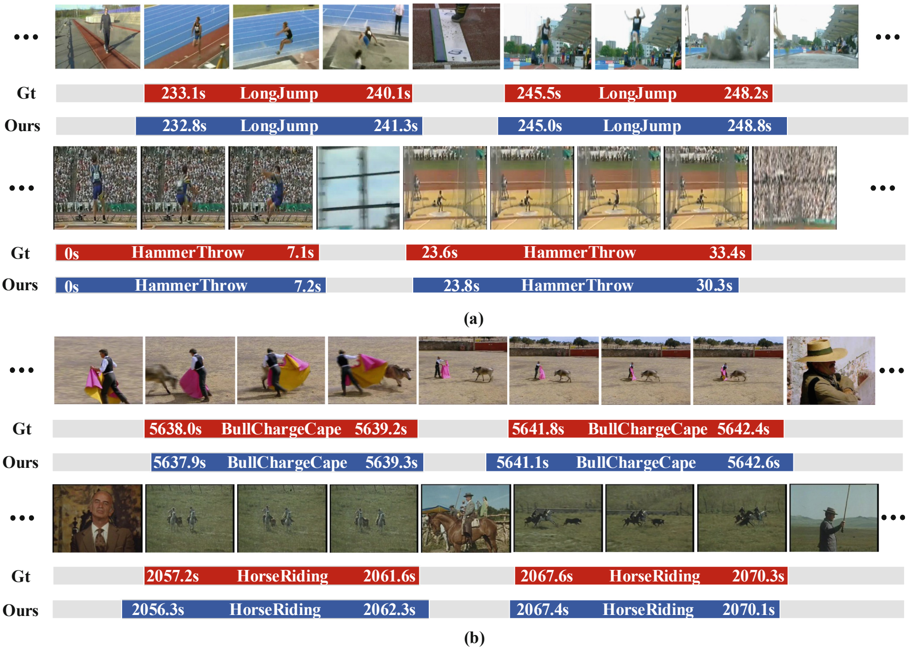
Table 6. Average precision on MEXaction2 dataset. The tIoU threshold is set to 0.5.

Methods	BullChargeCape	HorseRiding	mAP (%)
DTF [1]	0.3	3.1	1.7
S-CNN [35]	11.6	3.1	7.4
SSAD [16]	16.5	5.5	11.0
Our TDPN	**35.2**	**9.1**	**22.1**

(CBR) [5] and recently proposed TAL-Net [3] network. When the tIoU threshold is 0.5, our TDPN model achieves competitive performance. These results indicate that the TDPN model can detect the temporal boundaries in untrimmed video precisely. Figure 4 shows the AP at 0.5 tIoU for each action category of different methods including CBR [5], re-trained SSAD and our TDPN model. Our approach performs the best for 12 action categories compared with other methods, specially for “GolfSwing”, “HammerThrow” and “ThrowDiscus”. These results clearly demonstrates the superiority of our method. Qualitative detection results on THUMOS14 are shown in Fig. 5.

MEXaction2. We use all 38 untrimmed videos in MEXaction2 training dataset to train our TDPN model. The anchor scale ratios are the same as ones used in THUMOS14 dataset since the duration distribution of action instances on MEXaction2 is similar to the THUMOS14 dataset.

We compare TDPN model with other existing methods, including typical dense trajectory features (DTF) [1], Segment-CNN (SCNN) [35] and SSAD [16]. All methods are evaluated using standard criteria mentioned in Sect. 4.1. According to Table 6, our TDPN model outperforms other approaches for both “BullChargeCape” action and “HorseRiding” action, and the mAP at 0.5 tIoU threshold increases from 11.0% to 22.1% (about 11.1% improvement). The major challenge of this dataset is high variability in point of view, action duration for “HorseRiding” and image quality. These results indicate our TDPN model is


Gt
233.1s LongJump 240.1s
245.5s LongJump 248.2s
Ours
232.8s LongJump 241.3s
245.0s LongJump 248.8s
Gt
0s HammerThrow 7.1s
23.6s HammerThrow 33.4s
Ours
0s HammerThrow 7.2s
23.8s HammerThrow 30.3s
Gt
5638.0s BullChargeCape 5639.2s
5641.8s BullChargeCape 5642.4s
Ours
5637.9s BullChargeCape 5639.3s
5641.1s BullChargeCape 5642.6s
Gt
2057.2s HorseRiding 2061.6s
2067.6s HorseRiding 2070.3s
Ours
2056.3s HorseRiding 2062.3s
2067.4s HorseRiding 2070.1s

(a)

(b)

Fig. 5. Qualitative visualization of the actions detected by the TDPN network. Figure (a) and Figure (b) show detection results for two action classes on THUMOS14 dataset and MEXaction2 dataset respectively

capable of handing such problems. Figure 5 displays the visualization of detection results for “BullChargeCape” and “HorseRiding” respectively.

5 Conclusions

In this paper, we propose the Temporal Deconvolutional Pyramid Network (TDPN) for temporal action detection. The temporal convolutions are adopted to construct a initial feature pyramid in the bottom-up pathway. With the strong semantic features formed by Temporal Deconvolution Fusion (TDF) modules in various pyramidal hierarchies, the detection performance can be improved significantly. We explore different fusion strategies and the experiment results have showed that element-wise sum can achieve the excellent performance. To further improve detection accuracy, a strict strategy for label assignment is designed to train the model. Finally, the proposed TDPN model achieves competitive performance on THUMOS14 and outperforms other approaches on MEXaction2, which demonstrates our method is capable of localizing temporal boundaries precisely. For future work, we plan to explore end-to-end system that combines feature extractor and TDPN model to detect action instances from raw videos.

References

1. Mexaction2 (2015). http://mexculture.cnam.fr/xwiki/bin/view/Datasets/Mex+action+dataset
2. Abadi, M., Agarwal, A., et al.: TensorFlow: largescale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)
3. Chao, Y., Vijayanarasimhan, S., Seybold, B., et al.: Rethinking the faster R-CNN architecture for temporal action localization. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, Salt Lake City (2018)
4. Gaidon, A., Harchaoui, Z., Schmid, C.: Temporal localization of actions with actoms. IEEE Trans. Pattern Anal. Mach. Intell. **35**(11), 2782–2795 (2013)
5. Gao, J., Yang, Z., Nevatia, R.: Cascaded boundary regression for temporal action detection. In: British Machine Vision Conference. BMVA Press, London (2017)
6. Gao, J., Yang, Z., Sun, C., Chen, K., Nevatia, R.: TURN TAP: temporal unit regression network for temporal action proposals. In: IEEE International Conference on Computer Vision, pp. 3648–3656. IEEE Computer Society, Venice (2017)
7. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Teh, Y.W., Titterington, M. (eds.) The Thirteenth International Conference on Artificial Intelligence and Statistics, vol. 9, pp. 249–256. PMLR, Sardinia (2010)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778. IEEE, Las Vegas (2016)
9. Heilbron, F.C., Niebles, J.C., Ghanem, B.: Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1914–1923. IEEE, Las Vegas (2016)
10. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
11. Jain, M., van Gemert, J.C., et al.: Action localization by tubelets from motion. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 740–747. IEEE, Columbus (2014)
12. Jain, M., Gemert, J.V., et al.: End-to-end, single-stream temporal action detection in untrimmed videos. In: British Machine Vision Conference. BMVA Press, London (2017)
13. Jiang, Y.G., Liu, J., Roshan Zamir, A., et al.: THUMOS challenge: action recognition with a large number of classes (2014). http://crcv.ucf.edu/THUMOS14/
14. Laptev, I.: On space-time interest points. Int. J. Comput. Vis. **64**(2–3), 107–123 (2005)
15. Lin, T., Dollr, P., Girshick, R., et al.: Feature pyramid networks for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 936–944. IEEE, Honolulu (2017)
16. Lin, T., Zhao, X., Shou, Z.: Single shot temporal action detection. In: Proceedings of the 2017 ACM on Multimedia Conference, pp. 988–996. ACM, New York (2017)
17. Mettes, P., van Gemert, J.C., et al.: Bag-of-fragments: selecting and encoding video fragments for event detection and recounting. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, pp. 427–434. ACM, New York (2015)
18. Oneata, D., Verbeek, J., Schmid, C.: The LEAR submission at Thumos 2014 (2014). https://hal.inria.fr/hal-01074442

19. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: unified, real-time object detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788. IEEE, Las Vegas (2016)
20. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017)
21. Richard, A., Gall, J.: Temporal action detection using a statistical language model. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3131–3140. IEEE, Las Vegas (2016)
22. Shou, Z., Chan, J., Zareian, A., Miyazawa, K., Chang, S.: CDC: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1417–1426. IEEE, Venice (2017)
23. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, pp. 568–576. MIT Press, Cambridge (2014)
24. Tran, D., Bourdev, L., Fergus, R., et al.: Learning spatiotemporal features with 3D convolutional networks. In: 2015 IEEE International Conference on Computer Vision, pp. 4489–4497. IEEE Computer Society, Washington (2015)
25. Uijlings, J.R., Van De Sande, K.E., et al.: Selective search for object recognition. Int. J. Comput. Vis. **104**(2), 154–171 (2013). https://doi.org/10.1007/s11263-013-0620-5
26. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: IEEE International Conference on Computer Vision, pp. 3551–3558. IEEE, Sydney (2013)
27. Wang, L., Xiong, Y., Lin, D., Gool, L.V.: Untrimmednets for weakly supervised action recognition and detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 6402–6411. IEEE, Honolulu (2017)
28. Wang, L., Qiao, Y., Tang, X.: Action recognition and detection by combining motion and appearance features (2014). http://crcv.ucf.edu/THUMOS14/
29. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
30. Xiong, Y., Wang, L., et al.: CUHK & ETHZ & SIAT submission to ActivityNet challenge 2016. arXiv preprint arXiv:1608.00797 (2016)
31. Xu, H., Das, A., Saenko, K.: R-C3D: region convolutional 3D network for temporal activity detection. In: IEEE International Conference on Computer Vision, pp. 5794–5803. IEEE, Venice (2017)
32. Yeung, S., Russakovsky, O., Mori, G., Fei-Fei, L.: End-to-end learning of action detection from frame glimpses in videos. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2678–2687. IEEE, Las Vegas (2016)
33. Yuan, J., Ni, B., Yang, X., Kassim, A.A.: Temporal action localization with pyramid of score distribution features. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3093–3102. IEEE, Las Vegas (2016)
34. Zhao, Y., Xiong, Y., Wang, L., et al.: Temporal action detection with structured segment networks. In: IEEE International Conference on Computer Vision, pp. 2933–2942. IEEE Computer Society, Venice (2017)
35. Zheng, S., Dongang, W., Fu, C.S.: Action temporal localization in untrimmed videos via multi-stage CNNs. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1049–1058. IEEE, Las Vegas (2016)