Neurocomputing 454 (2021) 373–381

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Efficient Two-Step Networks for Temporal Action Segmentation

Yunheng Li [a], Zhuben Dong [b], Kaiyuan Liu [b], Lin Feng [b,*], Lianyu Hu [a], Jie Zhu [c], Li Xu [c], Yuhan wang [a], Shenglan Liu [b,*]

[a] School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China
[b] School of Innovation and Enterpreneurship, Dalian University of Technology, Dalian 116024, China
[c] Alibaba Group, China

ARTICLE INFO

Article history:
Received 18 December 2020
Revised 24 April 2021
Accepted 30 April 2021
Available online 4 May 2021
Communicated by Zidong Wang

Keywords:
Temporal action segmentation
Temporal series pyramid networks
Local Burr suppression
Two-step method

ABSTRACT

Due to boundary ambiguity and over-segmentation issues, identifying all the frames in long untrimmed videos is still challenging. To address these problems, we present the Efficient Two-Step Network (ETSN) with two components. The first step of ETSN is Efficient Temporal Series Pyramid Networks (ETSPNet) that capture both local and global frame-level features and provide accurate predictions of segmentation boundaries. The second step is a novel unsupervised approach called Local Burr Suppression (LBS), which significantly reduces the over-segmentation errors. Our empirical evaluations on the benchmarks including 50Salads, GTEA and Breakfast dataset demonstrate that ETSN outperforms the current state-of-the-art methods by a large margin.

1. Introduction

Analyzing human actions in videos is of vital importance for many applications such as human action analysis [8,35,34,36,33] and surveillance [18]. It is crucial to temporally segment activities in long untrimmed videos for many application scenes, like surveillance [31,32] and video processing. Therefore, approaches for temporal action segmentation have received more attention. While recognizing actions from short trimmed videos has achieved great performance [26,23,29,1,27,28], labeling all the frames in a long untrimmed video is still challenging.

One main challenge is the problem of inaccurate action boundaries [30], which is caused by sudden label changes but gradual transitions of actions. The state-of-the-art methods mainly focus on improving receptive fields for modeling long-term dependency in order to identify hard-to-recognize frames of action boundaries. For example, the baseline model MS-TCN [4] expands the temporal receptive fields by increasing the number of network layers. However, the higher layers lead to the loss of local information and the lack of correlation of long-range dependency. In addition, simply relying only on the features of long-term dependency tends to output low confident or even incorrect predictions for ambiguous frames of action boundaries.

Another common challenge is that over-segmentation errors [4,17,30] always occur in temporal action segmentation tasks for single frame classification. The current state-of-the-art methods tend to predict incorrect segments because of the inevitable noise in videos. We define the incorrect frames produced by over-segmentation errors as burrs. Previous methods [4,17] alleviate this problem by additional temporal smoothing loss functions. Recent method [30] trains an extra network to smooth the burrs of predictions but suffers high computational cost. However, these methods are not satisfactory for solving the problem of over-segmentation errors.

To tackle the challenges above, we design the Efficient Two-Step Network termed as ETSN, as shown in Fig. 1. The first step of ETSN is the Efficient Temporal Series Pyramid Networks (ETSPNet) that captures both local and global temporal dependency. Moreover, it also recognizes ambiguous frames to provide accurate predictions of segmentation boundaries. The second step is a novel unsupervised approach called Local Burr Suppression (LBS), which addresses the problem of over-segmentation.

Different from the previous works [4,17] which enlarge modeling capacity by stacking multiple dilated convolutions, ETSPNet provides a new perspective to operate on the full temporal resolution. An efficient temporal series pyramid of dilated convolutions is used to enlarge temporal modeling capacity and receptive fields, which is beneficial to produce more confident predictions for ambiguous segmentation boundaries.

* Corresponding authors.
E-mail addresses: fenglin@dlut.edu.cn (L. Feng), liusl@dlut.edu.cn (S. Liu).

https://doi.org/10.1016/j.neucom.2021.04.121
0925-2312/

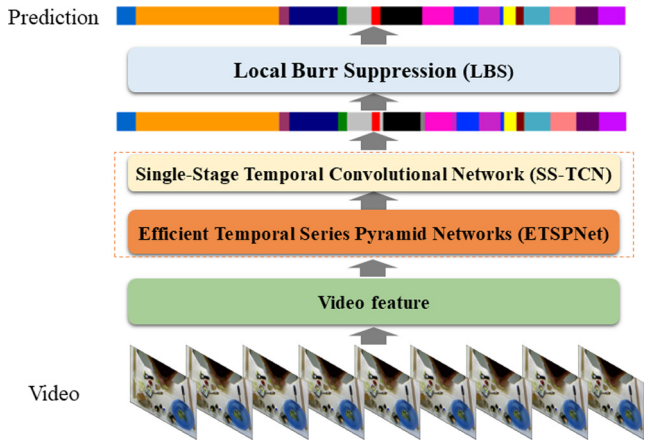


Fig. 1. Overview of the Efficient Two-Step Network (ETSN) comprised of Efficient Temporal Series Pyramid Networks (ETSPNet) and Local Burr Suppression (LBS).

Although ETSPNet has the ability of modeling various temporal dependency, burrs still exist in temporal action segmentation task. Different from smoothing loss function [4,17] and smoothing global predictions [30], we argue that burrs occur local of predictions due to over-segmentation. So, we propose a post-processing approach called LBS to locate and remove burrs. In this sense, ETSN, the union of ETSPNet which provides more accurate boundaries for burrs and LBS, is able to greatly reduce over-segmentation errors (See Section 3.2 for details).

Finally, we evaluate ETSN on three datasets with high spatio-temporal variations: including GTEA [5], 50Salads [25] and Breakfast dataset [9]. Our method achieves an improvement of 2.3% for GTEA and 5.3% for Breakfast in F1@10 score and 4.5% gain in segmental edit distance for 50salads. In summary, our paper makes three main contributions:

1.**Efficient Temporal Series Pyramid Networks (ETSPNet)**: We design a novel network structure that efficiently captures both local and global frame-level features to boost the performance of frame-level classification.
2.**Local Burr Suppression (LBS)**: We propose a post-processing unsupervised method to greatly reduce over-segmentation errors. LBS is a general solution that F1 scores have improved by combining state-of-the-art methods.
3.**Action Segmentation with ETSN**: Our two-step approach achieves state-of-the-art results on three challenging benchmarks for action segmentation: including 50Salads [25], GTEA [5] and Breakfast dataset [9].

2. Related work

2.1. Temporal action segmentation

Temporal convolution networks (TCN) are used by Action Segmentation methods to capture frame-level features for frame labeling. For example, Lea et al. [13] used an encoder-decoder architecture to capture long-range dependency for action segmentation and detection. Lei et al. [15] introduced deformable convolutions to replace the normal convolution of [13] and added a residual stream with high temporal resolution. In contrast to above approaches, Farha et al. [4] used dilated convolutions to increase the temporal receptive fields instead of the commonly used temporal pooling. Li et al. [17] introduced a dual dilated layer on top of [4] that combines both large and small receptive fields. However, it greatly increases the amount of model parameters. Chen et al. [1] [2] proposed Self-Supervised Temporal Domain Adaptation (SSTDA) to jointly align cross-domain feature spaces embedded with local and global temporal dynamics.

The above methods [13,4,17] enhance the temporal receptive fields to capture long-term dependency by increasing the complexity of the model. But only capturing the features of long-term dependency might result in losing fine-grained information that is necessary for frame recognition. Different from the previous methods, our framework uses an efficient pyramid model to capture both local and global features of frames. And it tackles the problems of inaccurate boundaries and incorrect classifications of short actions.

2.2. Reducing over-segmentation errors

For over-segmentation errors, the state-of-the-art methods generally adopted smooth label strategy. The previous methods [4,17] usually added temporal smoothing loss functions to suppress over-segmentation errors. For over-segmentation problem, the two-stream segmentation methods achieve better performance than the existing single-stream segmentation methods. Li et al. [30] designed the Local Barrier Pooling (LBP) as the second-stream network to generate smoother predictions from the output of temporal model. Compared with LBP [30], we propose an unsupervised method called LBS which only modify local over-segmentation frames of the video. The union of ETSPNet and LBS can better alleviate the problem of over-segmentation.

3. Technical approach

In this section, we introduce our approach ETSN for action segmentation. Section 3.1 illustrates how to utilize an efficient temporal series pyramid network to enlarge temporal modeling. Section 3.2 shows how our proposed novel Local Burr Suppression removes the burrs of predictions. Section 3.3 introduces the training details of our framework. Let $X = [x_1, \cdots, x_T] \in R^{D_s \times T}$ be the input to ETSN, our goal is to predict the class label for frames $C = [c_1, \cdots, c_T]$, where T is the number of frames in a video and D_s is the dimension of the feature.

3.1. Efficient temporal series pyramid networks

It is of vital importance to establish models for temporal structures to recognize label of frames in temporal action segmentation task. To get better performance, recent methods [4,17] mainly focus on capturing long-term dependency by increasing the complexity of the model. But simply capturing the features of long-term dependency might result in losing fine-grained information and predicting inaccurate action boundary. In the field of image segmentation, the spatial pyramid structure is introduced in Efficient Spatial Pyramid Network (ESPNet) [20] to increase the accuracy of determining image segmentation boundaries. Inspired by this, we build a pyramid structure in the time series to capture multi-scale features with a large temporal receptive field, so that the ETSPNet can predict more accurate temporal sequence segmentation boundary.

3.1.1. ETSPNet

The first layer of ETSPNet is a 1×1 convolutional layer that adjusts the dimension of the input features to match the number of feature maps in the network. Moreover, the output of this layer is used as the starting feature of the network. To generate the temporal features of multi-scale receptive fields, this layer is followed by several layers of dilated 1D convolution, whose input is the starting feature. In addition, we use a dilation factor that is doubled

[1] Since they use the target label for cross-domains action segmentation during training model, we will not compare with SSTDA in this paper.

at each layer. All these layers have the same number of convolutional filters with kernel size 3. Each layer applies a dilated convolution with ReLU activation and 1×1 convolutional layer. The dilated 1D convolution of layer k generates the feature vector H_k, which can be formulated as:

$$H_k = W_2 * (ReLU(W_1 * X_{start} + b_1)) + b_2, \quad (1)$$

where X_{start} is the starting feature, $*$ denotes the convolution operator, $W_1 \in R^{3\times D\times D}$ is the weight matrix of the dilated convolution filters with kernel size 3 and D is the number of convolutional filters, $W_2 \in R^{1\times D\times D}$ is the weight matrix of the 1×1 convolution, and $b_1, b_2 \in R^D$ indicate the bias vectors. Since the receptive fields grows 2^k with the number of layers k, we achieve a very large receptive fields with a few layers by using dilated convolution, which helps in preventing the model from over-fitting during training. In addition, we connect the output features of each dilated convolution layer together to obtain the features of different receptive fields, which is significantly benefits to identify frames with different degrees of difficulty. The receptive field of each layer can be formulated as:

$$ReceptiveField(k) = 2^k - 1, \quad (2)$$

where $k \in [1, K]$ is the layer number.

Only using 4 convolution layers to capture frame-level features easily incurs under-fitting problems for hard frames of training data, which result in achieving bad performance in fine-grained action segmentation task. So the output of the previous dilated convolutional layer is input to each dilated convolutional layer on the base of the starting feature. These operations are illustrated in Fig. 2. Not only does ETSPNet have a complex network structure but also the temporal receptive fields of each dilated convolution are doubled. In addition, 1×1 convolution is utilized to keep consistent with the dimension of the starting feature. We further use residual connections upon the output of the last convolution to facilitate gradients. The output vector G_k of the improved dilated 1D convolution can be formulated as:

$$G_k = W_2 * (ReLU(W_1 * (X_{start} + G_{k-1}) + b_1)) + b_2, \quad (3)$$

where G_{k-1} indicates the output of the previous layer. To obtain the probabilities for the output class M_t, we apply a 1×1 convolution over the output of the last convolution layer and then add a softmax activation layer as:

$$M_t = Softmax(Wh_{E,t} + b), \quad (4)$$

where $h_{E,t}$ is the output of the last convolution layer at time t, $W \in R^{C\times D}$ and $b \in R^C$ are the weights and bias for the 1×1 convolution layer. C is the number of classes and D is the number of convolutional filters.

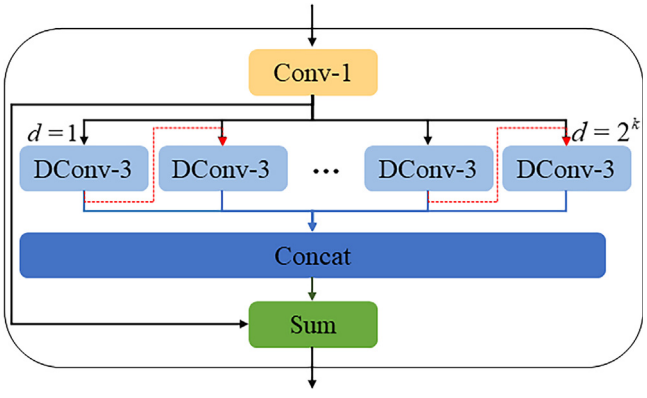


Fig. 2. Overview of the ETSPNet. Conv-1 is a 1×1 convolutional layer. And, DConv-3 is a dilated 1D convolution with ReLU activation and 1×1 convolutional layer. At each layer k, the dilated residual layer uses a convolution with dilated factor 2^k.

In order to refine the initial predictions, we add multi-stage temporal convolutional network [4] based on the previous stages. Different from MS-TCN [4], we use ETSPNet instead of SS-TCN [4] to extract features of the frames and generate initial predictions. It contains multi-scale features and preserves the temporal resolution of the input sequence, which produces more confident predictions of boundaries. More specifically, using long-term dependency of frame-level features, ETSPNet can identify the different classifications of two adjacent video segments while it can recognize more accurate action boundaries by short-term dependency.

3.2. Local Burr suppression

Although ETSPNet improves the performance of action segmentation, the predictions still contain over-segmentation errors. Previous work has shown that temporal smoothing losses of temporal consistency [4,17] still tend to over-segment actions. Extant two-stream method [30] generates smoother predictions from the output of temporal model by averaging predictions among neighborhood with adaptive weights aware of action boundaries. However, it requires an additional complex network, which suffers from expensive computational cost.

Algorithm 1: Local Burr Suppression (LBS)

Input: Initial prediction M_t, Pre-set burrs window size L, Pre-set confidence threshold P
Output: Refinded predictions M_r
1: Remove the short action segments
2: Calculate the length of each action segment to get the action length (n).
3: **if** ($n < L$) and ($Mean(P_i) < P$)**then**
4: Locate the burrs.
5: **if** Even-numbered continuous burrs **then**
6: Perform the even-numbered continuous burrs process.
7: **else** Odd-numbered continuous burrs
8: Ignore the middle burrs and perform the even-numbered continuous burrs process. 9: **end if**
10: **if** Isolated burrs **then**
11: Perform the isolated burrs process.
12: **end if**
13: **end if**

Different from smoothing loss function [4,17] and smoothing global predictions [30] of all the frames, we argue that burrs are caused by over-segmentation occured local of prediction. Inspired by the non-maximum suppression (NMS) [7] removal of redundant detection boxes, we provide a post-processing unsupervised method called LBS, which is effective to correct over-segmentation errors. LBS can be implemented as Algorithm 1, it can be simply decomposed into two parts after the predictions of ETSPNet: First, locating the real burrs with the length and the average confidence of the segments. Second, removing burrs of the predictions according to the type of burrs.

3.2.1. Locating the burrs

For the predicted labels generated by ETSPNet, we calculate the length of each action segment to get the action length (n). Then the short action segments that are impossible to occur are directly removed. Moreover, we judge the action segment smaller than the pre-set burrs window size L as suspicious burrs. In addition, the real burrs are picked up from suspicious burrs by the pre-set

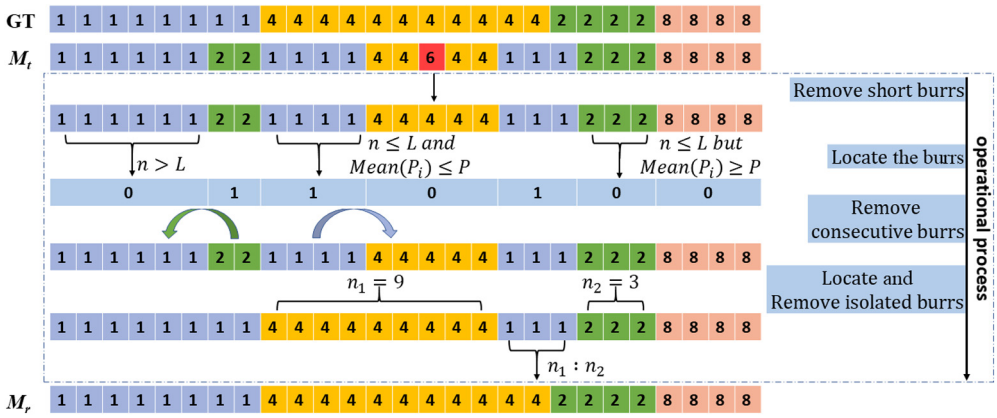


Fig. 3. Overview of how LBS removes the burrs of initial predictions. Different color bands represent different action classes, and the horizontal direction denotes the time axis. The default value is $L = 4$ and $P = 0.5$. Given the initial prediction M_t, we first locate the burrs with the length and the average confidence of action clips. Then, burrs are removed according to the type of them to generate refinded predictions M_r. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

confidence threshold (P) of the action segments, as shown in the above part of Fig. 3. It is difficult for LBS to regard correctly labeled video clips as burrs because the average confidence of real segments is much higher than that of burrs.

3.2.2. Removing the burrs

According to the number of continuous burrs, we divide them into continuous and isolated burrs. For continuous burrs, we take two consecutive burrs as an example to illustrate how LBS removes them, as illustrated in Fig. 3. The segmentation point between two continuous burrs is very possible to be a real segmentation boundary, because the model has identified the correct action boundaries even though the judgment of the segment category is incorrect. For this reason, we use the category of adjacent non-burr to replace that of previous burr and do the same for the subsequent burrs. For odd-numbered continuous burrs, we ignore the middle burrs and convert them to even-numbered burrs for processing. For isolated burrs, it is divided into two consecutive burrs (based on the length of the front and back adjacent non-burr segments of the burr) and performed the same removal work as that of two consecutive burrs.

3.3. Training ETSN

First, ETSPNet is trained to generate initial predictions. It contains ten dilated convolutional layers, in which the dilation factor d is doubled in each layer. For ETSPNet training, we set the number of filters to 64 in all the layers of the model (128 for GTEA) and the kernel size is 3. This is bacause GTEA dataset has only 28 videos but 11 action categories as shown in Table 1, model requires more features to recognize frames of the action segments. The detailed parameter settings of SS-TCN are the same as [4]. Compared with the 50salads dataset, the Breakfast dataset has 1712 videos and 48 action categories, which makes it more difficult to identify all the frames. So we use Adam optimizer with a learning rate 10^{-3} for 50Salads and GTEA dataset and a learning rate 5×10^{-4} for Breakfast dataset. Moreover, we multiply the learning rate by 0.5 every 30 epochs for all datasets. For LBP, we set $L = 40$ and $P = 0.31$ for GTEA, $L = 50$ and $P = 0.36$ for 50Salads, and $L = 100$ and $P = 0.68$ for Breakfast dataset.

Table 1
The statistics of action segmentation datasets.

	GTEA	50Salads	Breakfast
# videos	28	50	1712
# subjects	4	25	52
# classes	11	17	48
# instances/video	20	20	6
cross-validation	4	5	4

4. Experiments

4.1. Datasets and evaluation metrics

We evaluate our proposed ETSN on three challenging action segmentation datasets: GTEA [5], 50salads [25] and Breakfast dataset [9]. The overall statistics of the three datasets are listed in Table 1. Some sample images in GTEA and Breakfast datasets are shown in Fig. 4.

For all the datasets, we report the three evaluation metrics as in [4]: frame-wise accuracy (Acc), segmental edit score and the segmental F1 score at the IOU thresholds 0.10, 0.25 and 0.5, denoted by F1@ {10,25,50}. The segmental F1 score as proposed by [13] is used as a measure of the quality of the predictions. As with mAP detection scores, the precision and recall are computed for true positives, false positives, and false negatives summed over all classes, so the segmental F1 score is defined as $F1 = 2\frac{prect^* recall}{prect+recall}$. Although the accuracy is the most commonly used metric for temporal action segmentation, long action classes have a stronger impact than short action classes on this metric while over-segmentation errors can hardly affect it. The segmental edit score is only related to the actions sequence of predictions. Moreover, the F1 score can penalize over-segmentation errors while it does not penalize minor temporal shifts between the predictions and ground truth. As for this reason, we use the F1 score to evaluate the quality of the prediction.

4.2. Study on ETSPNet and LBS

In this subsection, we validate the ability of our proposed ETSPNet and LBS by comparing to MS-TCN [4] and MS-TCN++ [17]. ETSPNet aims to provide a more precise prediction of frames in videos. Although MS-TCN++ [2] [17] achieves better performance, it costs 100% extra time compared with MS-TCN [4]. The performance of ETSPNet is almost identical to that of MS-TCN++ (e.g., 0.5% on Breakfast and −0.6% on 50salads for Acc) as shown in Tables 4 and

[2] The training epoch is 100 for MS-TCN++ and 50 for MS-TCN and ETSPNet.

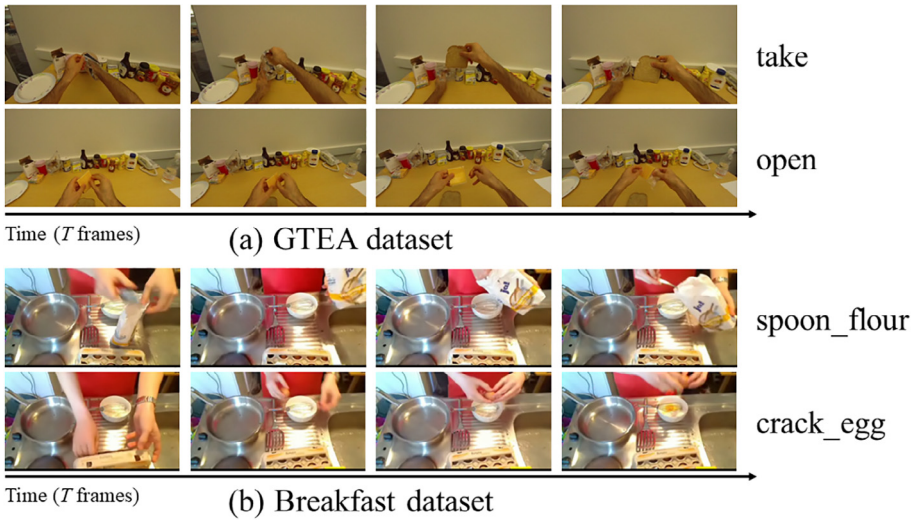


Fig. 4. Sample image categories from the GTEA and Breakfast datasets.

Table 2
FLOPs, speed and GPU memory cost. The FLOPs is for one random sample with frames = 1000. The speed and memory is measured on a single GTX 2080 Ti GPU with batch size = 1 in PyTorch evaluation mode. Network time is time of training the model for 50 epochs on 50Salads dataset.

Model	Params	FLOPs	Network time	Memory
MS-TCN [4]	80.0 K	80.0 M	5.2 min	1.6G
MS-TCN++ [17]	99.8 K	99.8 M	9.0 min	1.7G
ETSPNet	84.1 K	84.1 M	5.2 min	1.7G

5. Moreover, Table 2 shows ETSPNet halves the Network time on the 50salads dataset. In addition, the latency and GPU memory cost of ETSPNet are almost the same as MS-TCN [4] baseline but achieves better performance (e.g., 9.8% on Breakfast for F1@10 score, 9.0% on 50salads and 3.3% on GTEA for Edit score). This is because capturing both local and global temporal features makes it easier to recognize simple-to-hard frames of the action segments. As shown in Fig. 5, compared with MS-TCN++ [17], the action boundary of the sequence predicted by ETSPNet is more accurate and closer to Ground Truth. But ETSPNet also strengthens the effect of noises and results in the drop of F1 score compared with MS-TCN++ [17].

LBS aims to reduce over segmentation errors. As a general unsupervised approach, LBS only deals with the initial predictions generated by current predictive models. So we directly combine LBS with state-of-the-art methods for experiments. As shown in Tables 4 and 5, with LBS refining the predicted labels, MS-TCN [4] and MS-TCN++ [17] gain higher F1 scores on three datasets than before. This effect is clearly visible in terms of reducing over segmentation errors, LBS achieves huge improvement on the basis of the method simply using smoothing loss function. Furthermore, to prove LBS can suppress burrs correctly, we visualized the prediction of labels as shown in Fig. 5. It shows that although the correct action clips of 'add_salt' whose length is less than L but confidence is above P, so LBS do not recognize it as burrs. Although ETSPNet can achieve a high accuracy, there are many burrs in its predictions, which results in a low F1 score. This is mainly due to the accuracy is to evaluate single frame and does not reflect over-segmentation errors. In contrast, the F1 score can penalize over-segmentation errors, which is more indispensable for temporal action segmentation task. The F1 score of ETSN is improved by large margin, and LBS ensures the completeness of the action clips. However, the accuracy of ETSN drops, because LBS may change the category of the correct clips which have only a few frames into the wrong category.

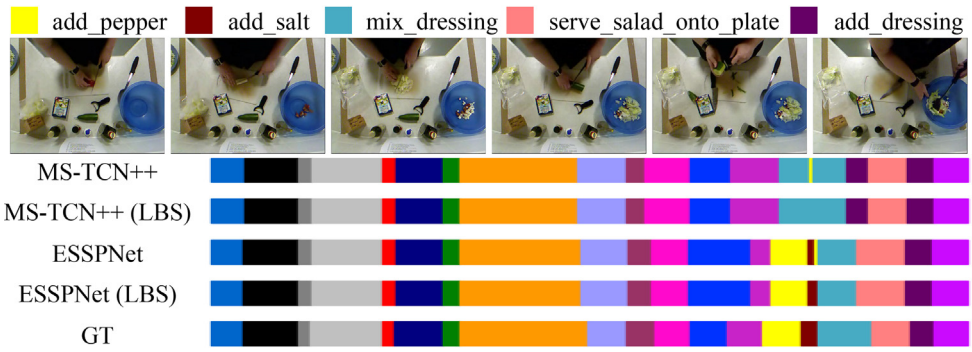


Fig. 5. The visualization of temporal action segmentation for differrnt methods with color-coding on 50salads. We compare the predictions of MS-TCN++ [17] and ETSPNet with that of the unions of them and LBS.

Table 3
Impact of L and P on the Breakfast dataset.

Impact of P		F1@{10, 25, 50}		Edit	Acc
ETSN($P = 0.58, L = 100$)	73.0	68.1	54.6	69.9	67.4
ETSN($P = 0.65, L = 100$)	74.0	68.6	55.6	70.5	67.1
ETSN($P = 0.68, L = 100$)	**74.0**	**69.0**	**56.2**	**70.3**	**67.8**
ETSN($P = 0.69, L = 100$)	73.4	68.4	55.0	69.3	67.0
ETSN($P = 0.70, L = 100$)	73.6	68.7	55.8	68.9	67.5
Impact of L		F1@{10, 25, 50}		Edit	Acc
ETSN($L = 90, P = 0.68$)	73.3	68.5	55.0	69.0	67.1
ETSN($L = 100, P = 0.68$)	**74.0**	**69.0**	**56.2**	**70.3**	**67.8**
ETSN($L = 110, P = 0.68$)	73.9	68.9	55.6	69.6	67.5

Table 4
Comparing our proposed method with existing methods on the Breakfast dataset. (* obtained from [3]).

Breakfast	F1@{10, 25, 50}			Edit	Acc
ED-TCN [13]*	–	–	–	–	43.3
HTK [11]	–	–	–	–	50.7
TCFPN [3]	–	–	–	–	52.0
HTK(64 [10])	–	–	–	–	56.3
GRU [21]*	–	–	–	–	60.6
GRU+ length prior [12]	–	–	–	–	61.3
MS-TCN (I3D) [4]	52.6	48.1	37.9	61.7	66.3
MS-TCN [4] (LBS)	70.7	65.0	52.0	69.1	68.4
MS-TCN++ [17]	64.1	58.6	45.9	65.6	67.6
MS-TCN++ [17] (LBS)	73.1	67.9	54.7	69.3	68.4
BCN [30]	68.7	65.5	55.0	66.2	70.4
ETSPNet	**62.4**	**57.0**	**45.2**	**66.2**	**68.1**
ETSN	**74.0**	**69.0**	**56.2**	**70.3**	**67.8**

4.3. Ablation study on hyper-parameters

The effect of the proposed removing burrs is controlled by two hyper-parameters: L and P. In this section, we study the impact of these parameters and see how they affect the performance of the proposed model.

Impact of *L*: To analyze the effect of this parameter, we train different models with different values of L. As shown in Table 3, L is of no obvious effect on the performance. Reducing L to 90 still improves the performance but drops compared to the default value of $L = 100$. Increasing its value to $L = 110$ also causes a degradation in performance. This drop in performance is due to too small L may lose some real burrs while too large L will incorrectly recognize true clips as suspicious burrs. And even LBS may delete some correct clips.

4.3.1. Impact of P

This hyper-parameter defines the threshold to distinguish the real clips and burrs. While setting $P = 0.68$ is the best performance, reducing P to 0.70 still gives an improvement over the baseline. This is mainly because when P is too large, LBS would judge some correct action clips as burrs.

4.4. Comparison with the state of the art

Compared with MS-TCN [4] and MS-TCN++ [17], although ETSPNet produces more burrs, LBS can remove them more easily. This is because the length of burrs generated by ETSPNet are all less than the pre-set burrs window size L of LBS, so LBS can remove all burrs in the predictions as shown in Fig. 5. In addition, since ETSPNet provides more accurate boundary for both action segments and

Table 5
Comparing our proposed method with existing methods on 50 Salads and GTEA dataset.

Dataset	50salads					GTEA				
Method	F1@{10, 25, 50}			Edit	Acc	F1@{10, 25, 50}			Edit	Acc
Spatial CNN [14]	32.3	27.1	18.9	24.8	54.9	41.8	36.0	25.1	–	54.1
Bi-LSTM [24]	62.6	58.3	47.0	55.6	55.7	66.5	59.0	43.6	–	55.5
Dilated TCN [13]	52.2	47.6	37.4	43.1	59.3	58.8	52.2	42.2	–	58.3
ST-CNN [14]	55.9	49.6	37.1	45.9	59.4	58.7	54.4	41.9	–	60.6
TUnet [22]	59.3	55.6	44.8	50.6	60.6	67.1	63.7	51.9	60.3	59.9
ED-TCN [13]	68.0	63.9	52.6	52.6	64.7	72.2	69.3	56.0	–	64.0
TResNet [6]	69.2	65.0	54.4	60.5	66.0	74.1	69.9	57.6	64.4	65.8
TRN [16]	70.2	65.4	56.3	63.7	66.9	77.4	71.3	59.1	72.2	67.8
TDRN + Unet [16]	69.6	65.0	53.6	62.2	66.1	78.1	73.8	62.2	73.7	69.3
TDRN [16]	72.9	68.5	57.2	66.0	68.1	79.2	74.4	62.7	74.1	70.1
LCDC + ED-TCN [19]	73.8	–	–	66.9	72.1	75.4	–	–	72.8	65.3
MS-TCN [4]	76.3	74.0	64.5	57.9	80.7	85.8	83.4	69.8	79.0	76.3
MS-TCN [4](LBS)	83.5	80.4	71.1	75.5	80.8	89.8	87.5	72.3	84.7	76.5
MS-TCN++ [17]	80.7	78.5	70.1	74.3	83.7	88.8	85.7	76.0	83.5	80.1
MS-TCN++ [17](LBS)	83.8	82.5	74.2	76.4	81.9	90.3	89.0	76.2	85.0	78.4
BCN [30]	82.3	81.3	74.0	74.3	84.4	88.5	87.1	77.3	84.4	79.8
ETSPNet	**74.2**	**72.6**	**65.7**	**66.9**	**83.1**	**87.0**	**84.1**	**73.5**	**82.7**	**78.5**
ETSN	**85.2**	**83.9**	**75.4**	**78.8**	**82.0**	**91.1**	**90.0**	**77.9**	**86.2**	**78.2**

burrs, ETSN can outperform the state-of-the-art temporal action segmentation methods. In Tables 4 and 5, our model outperforms the state-of-the-art methods on the three datasets with respect to F1 score and segmental edit distance (such as a large margin up to 5.3% for the F1@10 score on the Breakfast dataset).

Although the accuracy of the the-state-of-art method is higher than that of ETSN, it is difficult to predict correct action boundary and even lose short action clips. In additon, the F1 score can penalize over-segmentation errors and not penalize for minor temporal shifts between the predictions and ground truth, which is more essential for temporal action segmentation task. In this sense, compared with the union of MS-TCN++ [17] and LBS, ETSN predicts more accurate action boundary, and ensures the completeness of action segments, which is closer to ground truth, as shown in Fig. 5. It is worth noting that compared with baseline our model uses fewer computational cost.

5. Conclusion

We present a new framework called ETSN for temporal action segmentation, which consists of two components: ETSPNet for capturing multi-scale features and Local Burr Suppression for recognizing and removing burrs of the predictions. Our empirical evaluation on the benchmark 50Salads, GTEA and Breakfast demonstrate that ETSN outperforms the state-of-the-art models by a large margin. The superior performance of ETSN owes to the fact that it is able to predict action segments more precisely and greatly reduce over-segmentation errors. It implies our two-step approach achieves significantly performance than the method that simply stacking deeper temporal convolutions layers and using smoothing loss. In future research, we will investigate more effective methods to remove burrs, and evaluate the proposed method on more datasets, such as supervised methods to remove burrs that exist in temporal action segmentation task.

CRediT authorship contribution statement

Yunheng Li: Conceptualization, Methodology, Software, Writing - original draft, Writing - review & editing, Investigation. **Zhuben Dong:** Methodology, Validation, Formal analysis. **Kaiyuan Liu:** Methodology, Validation, Formal analysis, Visualization. **Lin Feng:** Conceptualization, Validation, Formal analysis, Supervision. **Lianyu Hu:** Validation, Resources, Data curation, Supervision. **Jie Zhu:** Validation, Supervision. **Li Xu:** Data curation, Investigation. **Yuhan wang:** Writing - review & editing. **Shenglan Liu:** Methodology, Visualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

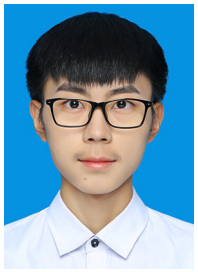
Acknowledgement

This work is supported by Liao Ning Revitalization Talents Program (No. XLYC1806006), a grant from the National Natural Science Foundation of China (No. 61672130, No. 61972064), the Fundamental Research Funds for the Central Universities (No. DUT19RC(3)012), and Dalian Youth Star of Science and Technology (No. 2019RQ035).

References

[1] J. Carreira, A. Zisserman, Quo vadis, action recognition? A new model and the kinetics dataset, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[2] M.H. Chen, B. Li, Y. Bao, G. AlRegib, Z. Kira, Action segmentation with joint self-supervised temporal domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
[3] L. Ding, C. Xu, Weakly-supervised action segmentation with iterative soft boundary assignment, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[4] Y.A. Farha, J. Gall, Ms-tcn: multi-stage temporal convolutional network for action segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3575–3584.
[5] A. Fathi, X. Ren, J.M. Rehg, Learning to recognize objects in egocentric activities, in: CVPR 2011, IEEE, 2011, pp. 3281–3288..
[6] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[7] Y. He, C. Zhu, J. Wang, M. Savvides, X. Zhang, Bounding box regression with uncertainty for accurate object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[8] T.S. Kim, A. Reiter, InterpreTable 3d human action analysis with temporal convolutional networks, in: in: 2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW), IEEE, 2017, pp. 1623–1631.
[9] H. Kuehne, A. Arslan, T. Serre, The language of actions: recovering the syntax and semantics of goal-directed human activities, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 780–787.
[10] H. Kuehne, J. Gall, T. Serre, An end-to-end generative framework for video segmentation and recognition, in: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2016, pp. 1–8..
[11] H. Kuehne, A. Richard, J. Gall, Weakly supervised learning of actions from transcripts, Comput. Vision Image Understand. 163 (2017) 78–89.
[12] H. Kuehne, A. Richard, J. Gall, A hybrid rnn-hmm approach for weakly supervised temporal action segmentation, IEEE Trans. Pattern Anal. Mach. (2018), intelligence.
[13] C. Lea, M.D. Flynn, R. Vidal, A. Reiter, G.D. Hager, Temporal convolutional networks for action segmentation and detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 156–165.
[14] C. Lea, A. Reiter, R. Vidal, G.D. Hager, Segmental spatiotemporal cnns for fine-grained action segmentation, in: European Conference on Computer Vision, Springer, 2016, pp. 36–52.
[15] P. Lei, S. Todorovic, Temporal deformable residual networks for action segmentation in videos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[16] P. Lei, S. Todorovic, Temporal deformable residual networks for action segmentation in videos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 6742–6751.
[17] S.J. Li, Y. AbuFarha, Y. Liu, M.M. Cheng, J. Gall, Ms-tcn++: multi-stage temporal convolutional network for action segmentation, in: IEEE Trans. Pattern Anal. Mach. Intell., 2020.
[18] X. Luo, H. Li, X. Yang, Y. Yu, D. Cao, Capturing and understanding workers' activities in far-field surveillance videos with deep action recognition and bayesian nonparametric learning, Comput. Aided Civil Infrastruct. Eng. 34 (2019) 333–351.
[19] K.N.C. Mac, D. Joshi, R.A. Yeh, J. Xiong, R.S. Feris, M.N. Do, Learning motion in feature space: locally-consistent deformable convolution networks for fine-grained action detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 6282–6291.
[20] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, H. Hajishirzi, Espnet: efficient spatial pyramid of dilated convolutions for semantic segmentation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 552–568.
[21] A. Richard, H. Kuehne, J. Gall, Weakly supervised action learning with rnn based fine-to-coarse modeling, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[22] O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-assisted Intervention, Springer, 2015, pp. 234–241.
[23] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, Adv. Neural Inf. Process. Syst. (2014) 568–576.
[24] B. Singh, T.K. Marks, M. Jones, O. Tuzel, M. Shao, A multi-stream bi-directional recurrent neural network for fine-grained action detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[25] S. Stein, S.J. McKenna, Combining embedded accelerometers with computer vision for recognizing food preparation activities, in: Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, 2013, pp. 729–738.
[26] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015.

[27] L. Wang, W. Li, W. Li, L. Van Gool, Appearance-and-relation networks for video classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[28] L. Wang, Y. Qiao, X. Tang, Action recognition with trajectory-pooled deep-convolutional descriptors, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
[29] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal segment networks: towards good practices for deep action recognition, in: European Conference on Computer Vision, Springer, 2016, pp. 20–36.
[30] Z.Z. Wang, Z.T. Gao, L.M. Wang, Z.F. Li, G.S. Wu, Boundary-aware cascade networks for temporal action segmentation, in: ECCV, Springer, 2020..
[31] X. Yan, S. Hu, Y. Mao, Y. Ye, H. Yu, Deep multi-view learning methods: a review, Neurocomputing (2021).
[32] X. Yan, Y. Ye, X. Qiu, M. Manic, H. Yu, Cmib: unsupervised image object categorization in multiple visual contexts, IEEE Trans. Ind. Inf. 16 (2019) 3974–3986.
[33] Z. Yao, G. Zhang, D. Lu, H. Liu, Learning crowd behavior from real data: a residual network method for crowd simulation, Neurocomputing 404 (2020) 173–185.
[34] X. Zhang, H. Gu, K. Ma, Dynamical mechanism for conduction failure behavior of action potentials related to pain information transmission, Neurocomputing 387 (2020) 293–308.
[35] X. Zhang, D. Ma, H. Yu, Y. Huang, P. Howell, B. Stevens, Scene perception guided crowd anomaly detection, Neurocomputing 414 (2020) 291–302.
[36] X. Zhang, X. Yang, W. Zhang, G. Li, H. Yu, Crowd emotion evaluation based on fuzzy inference of arousal and valence, Neurocomputing 445 (2021) 194–205.



Yunheng Li received the B.S. degree of Electrical engineering and its automation from Dalian University of Technology, China, 2020. Currently, he is working toward the M.S. degree in the School of Computer Science and Technology, Dalian University of Technology, China. His research interests include temporal action segmentation.

ZhuBen Dong currently studying as an undergraduate majoring in Mechanical Engineering at Dalian University of Technology, Dalian, China. His research interests include skeleton-based video classification.

Kaiyuan Liu is currently working toward the B.E. degree in the School of Computer Science and Technology, Dalian University of Technology, China. His research interests include human motion analysis, graph neural networks and machine learning.

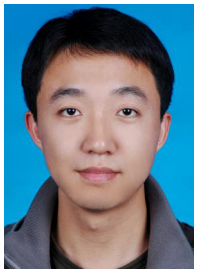
Lianyu Hu received the B.S. degree of Electronics and Information Engineering from Dalian University of Technology, China, 2018. Currently, he is a M.S. degree candidate in the Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology. His research interests include action recognition, graph convolution networks and skeleton-based video classification.

Lin Feng received the BS degree in electronic technology from Dalian University of Technology, China, in 1992, the MS degree in power engineering from Dalian University of Technology, China, in 1995, and the PhD degree in mechanical design and theory from Dalian University of Technology, China, in 2004. He is currently a professor and doctoral supervisor in the School of Innovation Experiment, Dalian University of Technology, China. His research interests include intelligent image processing, robotics, data mining, and embedded systems.

Jie Zhu received the Bachelor Degree in the school of Computer Science and Technology, Xidian University, Xi'an, Shanxi, China, in 2007. Currently, he is a senior product expert of Alibaba, Shenzhen, Guangdong, China. His research interests include database, AI.

Li Xu received the M.S. degree in software engineering and the Ph.D. degree in computer science from the Dalian University of Technology, Dalian, China, in 2010 and 2015, respectively. ACM senior member, CCF senior member. He is currently the Senior Product Director of the Alibaba Corporation, in charge of cloud native multimodal database Lindorm. He has more than ten years of platform product development experience.

Yuhan Wang is currently studying as an undergraduate majoring in Computer Science And Technology at Dalian University of Technology, Dalian, China. Her research interests include computer networks and computer architecture.



Shenglan Liu received the Ph.D. degree in the School of Computer Science and Technology, Dalian University of Technology, Dalian, Liaoning, China, in 2015. Currently, he is an associate professor with the School of Innovation and Entrepreneurship, Dalian University of Technology, Dalian, Liaoning, China. His research interests include manifold learning, human perception computing. Dr. Liu is currently the editorial board member of Neurocomputing.