Applied Intelligence (2022) 52:14053–14069
https://doi.org/10.1007/s10489-022-03382-x

Bottom-up improved multistage temporal convolutional network for action segmentation

Wenhe Chen · Yuan Chai · Miao Qi · Hui Sun · Qi Pu · Jun Kong, et al. ***[full author details at the end of the article]***

Accepted: 11 February 2022 / Published online: 2 March 2022

Abstract

Action segmentation involves locating and classifying human action segments in an untrimmed video, which is very important for understanding human activities. Segmenting actions in the video is a very challenging task due to the problem of ambiguous frames. Previous studies on this topic usually required additional inputs or constructed highly complicated network structures to achieve good performance. However, these additional inputs are not easy to obtain, and complicated network structures increase the costs of computation and storage. Hence, to mitigate these problems, we propose a bottom-up improved multistage temporal convolutional network (BUIMS-TCN) for action segmentation. Specifically, we first propose a smoothed dilated 1D convolution to learn the inherent local temporal dependencies. Second, we design an adaptive temporal fusion module (ATFM), which is a simple yet effective multiscale temporal-context information fusion module, to obtain better semantic feature representations. Finally, we introduce a new loss function to solve the imbalance between easy and hard samples. To the best of our knowledge, this is the first time that the above improvements have been incorporated into the action segmentation task. Extensive experiments verify that our model significantly outperforms the state-of-the-art baselines on three challenging benchmark datasets: Georgia Tech Egocentric Activities (GTEA), 50Salads, and the Breakfast dataset.

Keywords Action segmentation · Smoothed dilated 1D convolution · Adaptive temporal fusion module · Temporal convolutional network

1 Introduction

The automatic analysis of human actions in videos plays a critical role in various applications, such as anomaly detection [1, 2], human-computer interactions [3], intelligent services [4, 5], vision understanding [6], and intelligent control [7]. In the past few years, most studies focused on recognizing human actions in short, trimmed videos, and they have achieved great success due to the development of deep learning and large datasets [8–15]. However, videos in the real world usually contain multiple action segments, and they are untrimmed and relatively long [16]. To better understand human behaviors in these long videos, it is necessary to determine when and which categories of action segments occur sequentially in such a video. Hence, action segmentation has attracted more and more attention [17]. Action segmentation aims to simultaneously detect and classify each action segment in a video in time [18]. To obtain a good action segmentation result, activities consisting of dozens of actions in a long video should be effectively modeled.

Previous action segmentation methods, which are commonly based on sliding windows [19, 20], segmental models [21, 22] and recurrent models [23], cannot capture long-range temporal patterns and latent dependencies effectively. In addition, they are usually difficult to interpret and correctly train [18]. To relieve these drawbacks, temporal convolutional network (TCN) [18]-based action segmentation methods are developed. TCN-based methods contain a hierarchy of temporal convolutional filters (e.g., dilated 1D convolution) that have a large temporal receptive field with fewer parameters. Hence, TCN-based methods can not only model temporal patterns in long videos but also have high efficiency and require less memory.

The multistage temporal convolutional network (MS-TCN) [24] is a recently proposed state-of-the-art method based on a TCN and has become a widely used backbone network for action segmentation tasks. MS-TCN is composed of a series of dilated 1D convolutions and residual connection structures, which can expand the temporal receptive field with a small number of parameters and be applied to the full temporal resolution of a video. Though MS-TCN and some other TCN-based action segmentation methods can achieve exciting action segmentation results [1, 16], they still have two drawbacks: over-segmentation and ambiguous boundaries [1].

Over-segmentation means that incorrect predictions occur inside an action segment (as shown in Fig. 1b), and this is commonly caused by visual features in one action segment being too similar to those in other action segments. An ambiguous boundary indicates that an incorrect prediction occurs at the start or end of an action segment (as shown in Fig. 1d), which is caused by visual features of the boundary of two adjacent action segments changing too little (e.g., the action segment label suddenly changes from 'add_oil' to 'add_vinegar' at the boundary of two adjacent actions, but the visual features of these adjacent frames are gradually transformed). In this paper, we collectively refer to over-segmentation and ambiguous boundaries as the ambiguous frame problem for convenience.

When segmenting different actions in a video, the labels of informative frames are relatively easy to predict, while ambiguous frames are difficult to classify so that they will directly affect the prediction accuracy of the action segmentation model. Some existing studies [1, 16, 25] alleviate the ambiguous frame problem by manually adding additional labels to videos, designing complex input features, or adding more complex modules/branches to the architecture of MS-TCN, but they will highly increase the model complexity and the computational cost. That is, these works focus on introducing more information or more modules into MS-TCN to mitigate the ambiguous frame problem. To our knowledge, no work considers how to improve the underlying structure of MS-TCN to solve the ambiguous frame problem without an additional cost.

The underlying architecture of MS-TCN mainly consists of dilated 1D convolutions, residual connections and a loss function. In this paper, we first analyze the drawback of each module of MS-TCN that leads to the ambiguous frame problem in action segmentation results. Then, we provide corresponding improvements to relieve each drawback.

Chen et al. [26] proposed dilated convolution (also known as atrous convolutions), which has attracted much attention because it can expand the receptive field with fewer parameters. However, dilated convolution will yield grid artifacts [27, 28]. Grid artifacts generally refer to losing local spatial information when adopting dilated 2D convolutions to process 2D information (e.g., images). We believe that the problem of grid artifacts also exists in TCNs that use dilated 1D convolutions to process temporal information, and we term this kind of grid artifact as the temporal grid artifact. Some related works [27, 29, 30] have attempted to solve the problem of grid artifacts. However, few studies focused on solving the problem of temporal grid artifacts in temporal sequence modeling tasks (e.g., action segmentation).

To further clarify the concept of temporal grid artifacts, we first introduce the valid feature ratio (VFR) [30]. VFR is the ratio of the number of feature vectors involved in the computation to the total number of feature vectors in the convolution patch. When directly adopting dilated 1D convolutions to extract the temporal sequence context information, VFR usually decreases exponentially with the increase of the number of convolution layers, which will result in losing the local

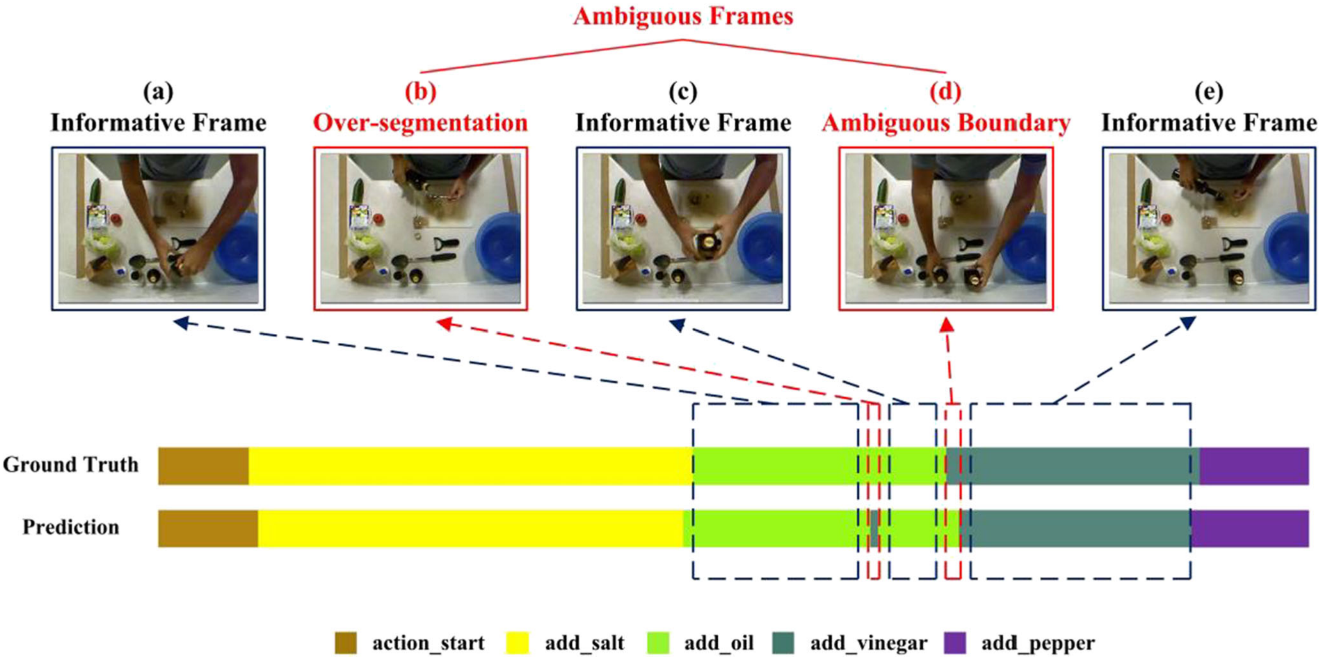


Fig. 1 Illustration of the ambiguous frame problem: over-segmentation and an ambiguous boundary. **a**–**e** Frames selected from one video containing several sequential action segments, where the informative frame represents the frame that can be correctly classified. **b** Over-segmentation occurs in the 'add_oil' action segment; that is, some intermediate frames in the 'add_oil' action are incorrectly predicted as 'add_vinegar'. **d** An ambiguous boundary appears at the beginning of the 'add_vinegar' action segment, which means that some frames at the boundary of the 'add_oil' and 'add_vinegar' actions are not correctly classified

temporal information and the dependence among the long-range temporal sequence information. As shown in Fig. 2, in convolution layer 2, the kernel size, dilation rate and receptive field of the dilated 1D convolution are 3, 2 and 5, respectively. Hence, the VFR of this dilated 1D convolution is 3/5, which means that the actual receptive field is 3 and the number of local temporal information loss is 2. In convolution layer 3, the VFR of the dilated 1D convolution is 3/9, which means the actual receptive field is still 3 but the number of local temporal information loss increases to 6. The phenomenon of local temporal information loss is termed temporal grid artifacts, which is one of the main factors that cause the ambiguous frame problem in action segmentation. Inspired by [29], we propose a smoothed dilated 1D convolution (SC), which introduces a smoothing operation into dilated 1D convolution, to mitigate the temporal grid artifacts caused by dilated 1D convolution.

Regarding the residual connection structure, Yu et al. [31] mentioned that the residual structure composed of dilated convolutions passes the high-frequency signals of the previous layer backward, making the problem of grid artifacts more serious. To address this problem, we propose an adaptive temporal fusion module (ATFM) inspired by [32, 33]. ATFM can adaptively integrate the temporal information of different scales and reduce grid artifacts caused by residual connections.

In a long and untrimmed video, there is a large gap between the numbers of informative frames and ambiguous frames, as shown in Fig. 1. The ambiguous frame can be seen as a hard sample that is difficult to correctly recognize, and the informative frame can be seen as an easy sample that is easily correctly recognized. The loss function in MS-TCN has not considered the imbalance of the numbers of easy and hard samples. This causes the large numbers of easy samples to overwhelm the classifier during the training, thereby degrading the network performance. To alleviate this problem, we introduce the focal loss [34] into the MS-TCN to construct our network. The focal loss can automatically reduce the weights of easy samples during the training process and make the model quickly focus on hard samples.

In this paper, we combine the abovementioned three improvements to construct an end-to-end trained network named bottom-up improved multistage temporal convolutional network (BUIMS-TCN). BUIMS-TCN requires the same inputs as MS-TCN and can maintain a similarly low model complexity. We evaluate BUIMS-TCN on three challenging action segmentation datasets: the Georgia Tech Egocentric Activities (GTEA) [35], 50Salads [36], and Breakfast [37] datasets. A large number of qualitative and quantitative experiments show that BUIMS-TCN can significantly mitigate the ambiguous frame problem. In summary, our work makes four main contributions:

1) We propose an SC to maintain the exponential expansion of the receptive field without the loss of information.
2) We devise an ATFM that can more effectively integrate multiscale temporal context information and reduce grid artifacts caused by residual connections.
3) We introduce a new loss function to solve the problem of the imbalance between easy and hard samples, which makes the model focus on hard-to-recognize ambiguous frames during the training process.
4) To the best of our knowledge, this work is the first exploration of the ambiguous frame problem caused by MS-TCN's underlying structure. Our proposed BUIMS-TCN

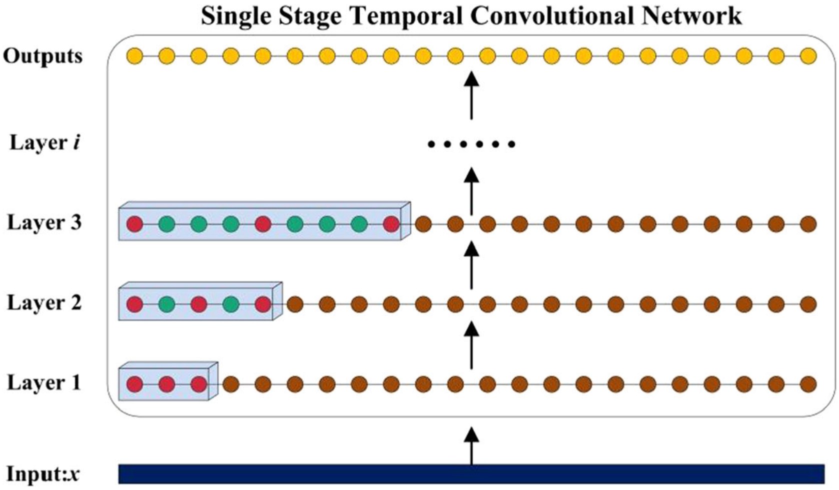


Fig. 2 Illustration of temporal grid artifacts. Taking a single-stage TCN as an example, the cuboid represents a dilated 1D convolution. In convolution layer 1, the kernel size is 3, and the dilation rate is 1. In convolution layer 2, the kernel size is 3, and the dilation rate is 2. By analogy, in the convolution of layer i, the kernel size is 3, and the dilation rate is 2^{i-1}. The red circles represent the features covered by a nonzero value in dilated convolution filters during the computation process, and the green circles represent the features covered by zero. The brown circles represent the remaining feature vectors that need to be computed for each layer. As the number of layers increases, the VFR decreases, and more local temporal information (the features covered by zero in dilated convolution filters) is lost, which leads to temporal grid artifacts

is end-to-end trained and requires no additional costs compared to MS-TCN. Extensive experiments demonstrate that BUIMS-TCN can achieve significant improvements over the state-of-the-art methods on the three challenging action segmentation datasets.

2 Related works

2.1 Action segmentation

With successful deep learning (DL) applications in the fields of image classification [38] and natural language processing [39, 40], considerable progress has been achieved in DL-based action segmentation tasks. TCNs [18] is a famous DL method, and the emergence of TCNs is a milestone in the field of action segmentation. Many researchers adopt TCN-based models for temporal action segmentation [23, 41, 42]. These models can capture long-range dependencies across frames without consuming massive labor costs and computing resources, which avoids the disadvantages of the models based on recurrent neural networks (RNNs) [43–45], sliding windows [19, 20] and other methods [46–48].

The recently proposed MS-TCN [24] is a state-of-the-art multistage TCN. Due to the good performance of MS-TCN, a lot of action segmentation studies [1, 16, 25] regard MS-TCN as a backbone network and add some additional manual labels, inputs, or more complicated network structures/branches to solve the ambiguous frame problem. For example, Chen et al. [16] proposed a self-supervised temporal domain adaptation (SSTDA) method, which contains two self-supervised auxiliary tasks (binary and sequential domain prediction) that are jointly aligned and embedded in the cross-domain feature space of local and global temporal dynamics. Wang et al. [1] proposed a dual-branch structure containing a stage cascade and a barrier generation module to solve the ambiguous frame problem. The stage cascade repeatedly inputs features into each stage, while the barrier generation module evaluates the boundary confidence and finally aggregates two branches by local barrier pooling. Ishikawa et al. [25] proposed a model consisting of an action segmentation branch (ASB) and a boundary regression branch (BRB). The BRB requires manual annotation of the ground truth of the action boundaries to participate in the training process, and the action boundaries predicted by the BRB refine the output from the ASB to mitigate the ambiguous frame problem.

The above MS-TCN based methods have obtained good action segmentation results, but to our knowledge, no research designs a new model based on analyzing the relationship between MS-TCN's underlying structure and the ambiguous frame problem. Hence, how to solve the ambiguous frame problem based on improving the underlying structure of MS-TCN is the main motivation of this paper and the main differences between our work and the abovementioned works.

2.2 Grid artifacts in dilated convolutions

Dilated convolution is widely used in deep convolutional neural networks (DCNNs) for 2D image processing, such as semantic image segmentation [26, 32] and object detection [49, 50]. Dilated convolution can expand the receptive field without additional parameters, but it has the problem of grid artifacts. Some existing solutions [27–31] have been proposed to address the grid artifacts problem. For instance, Wang et al. [27] proposed a hybrid dilated convolution (HDC), which compensates the local information of a single dilated convolution by adopting multiple dilated convolutions with the different dilation rates, and then integrated dense upsampling convolution (DUC) is used to restore the resolution of image segmentation. Yu et al. [31] added some traditional convolution blocks after the dilated convolution layer, and Hamaguchi et al. [28] added several dilated convolution layers with a decreasing dilation rate to alleviate the grid artifacts problem. These degridding methods tried to avoid losing local spatial information in a single dilated convolutional layer by adding different blocks in dilated convolutional layers, but the computational cost will increase exponentially with the number of dilated convolutional layers. Different from these studies, Wang and Ji [29] proposed two simple and effective degridding methods, which solve grid artifacts by smoothing the dilated convolution operation itself. Wu et al. [30] proposed a Kronecker convolution, which first adopts the Kronecker product expansion convolution kernel to consider the partial features neglected by dilated convolutions and then designs a tree structure to fuse the multiscale Kronecker convolutions. The abovementioned methods have achieved good performances, but methods [27, 28, 31] dramatically increase the computational cost. Method [30] does not greatly increase the computational cost, but it requires presetting a parameter to determine the shape of the convolution. Method [29] is a relatively good method among those mentioned.

The above methods [27–31] focus on relieving spatial grid artifacts in dilated 2D convolutions, rather than temporal grid artifacts in dilated 1D convolutions. Therefore, to alleviate temporal grid artifacts, we are inspired by [29] and propose an SC. To the best of our knowledge, this is the first time that temporal grid artifacts have been considered in action segmentation from the perspective of dilated 1D convolution.

2.3 Multiscale feature fusion

In the field of image segmentation, the full convolutional network (FCN) [51] has a relatively good performance. Therefore, many segmentation methods have been proposed based on the architecture of FCN, which are mainly divided into two categories: dilated FCNs [26, 32, 52, 53] and encoder-decoder networks [54–57].

Dilated FCNs adopt dilated convolution to maintain the receptive field and combine multiscale context modules to process high-level semantic features. For instance, Chen et al. [32] proposed atrous spatial pyramid pooling (ASPP), which employs parallel atrous convolutional layers with multiple dilation rates to capture multiscale contextual information. Zhang et al. [53] proposed a context-encoding module to capture global contextual information.

Encoder-decoder networks are composed of two parts—an encoder and a decoder. The encoder extracts multilevel feature maps, which are then combined with features processed by the decoder to produce the final result. Ronneberger et al. [55] introduced a skip connection to construct U-Net, which employes an encoder to learn features and uses the activation of a corresponding decoder to gradually restore spatial information. Lin et al. [56] proposed a generic multipath refinement network (RefineNet) that first concatenates features extracted by multiple windows with different sizes, then fuses them with the learnable weights. Chen et al. [57] combined the advantages of encoder-decoder networks and dilated FCNs, specifically, they employed a decoder to replace spatial pyramid pooling to refine segmentation results.

MS-TCN is a dilated FCN that utilizes a residual connection structure to fuse temporal contextual information. Yu et al. [31] pointed that the residual structure composed of dilated convolutions (called a dilated residual structure) passes the high-frequency signal from the previous layer by the same weight, which makes the temporal grid artifacts more serious. However, to the best of our knowledge, there is no relevant research on how to improve the dilated residual structure to decrease temporal grid artifacts. In this paper, we propose an ATFM that can adaptively fuse multiscale temporal context information, thereby greatly alleviating the temporal grid artifacts and ambiguous frame problem brought about by the dilated residual structure.

2.4 Sample imbalance

Sample imbalance is very common in the field of object detection. For instance, the object detection methods including one-stage detectors [58–60] and two-stage detectors [61–63] will first detect a large number of candidate positions of objects in an image, but only a few positions contain real objects. Hence, the number of negative samples (candidate positions are the background) is much larger than the number of positive samples (candidate positions contain objects). In general, negative samples are easily classified and do not provide much useful information to train the model, but due to a large number of negative samples, negative samples will dominate the training process, which makes the network biased toward identifying these easily classified negative samples while ignoring difficult-to-classify positive samples [34]. This results in network performance degradation. To solve this problem, some studies [64, 65] implement online hard example mining (OHEM) by sampling hard samples during training or utilizing more complex sampling/reweighting schemes. Li et al. [66] proposed a gradient harmonizing mechanism (GHM), which measures the distribution of easy and hard samples according to a gradient. Lin et al. [34] proposed a new loss function named focal loss that prevents easily classified negative samples from overwhelming the model during the training process. Focal loss solves the imbalance between easily and difficultly classified samples by reshaping the standard cross-entropy loss to automatically reduce the weight of easily classified samples and make the model focus on difficultly classified samples.

The loss function adopted in MS-TCN does not consider the serious problem of the sample imbalance. In this paper, we introduce focal loss [34] into our proposed network. Extensive experiments show that adding focal loss can alleviate the problem of sample imbalance, enabling the model to effectively identify ambiguous frames.

3 Our work

In this section, we introduce the details of our proposed BUIMS-TCN. The architecture of BUIMS-TCN is shown in Fig. 3. In BUIMS-TCN, we first propose an SC to learn local temporal dependencies for solving the ambiguous frame problem. Then, we design an ATFM to adaptively fuse multiscale context information and relieve the defect of residual connections, which can further alleviate the ambiguous frame problem. Finally, we adopt a new loss function to focus on mining the information of hard samples (difficult-to-classify samples), which can solve the ambiguous frame problem caused by the imbalance between the numbers of easy and hard samples. Our proposed BUIMS-TCN can be trained end-to-end without additional labor costs or complicated network structures/branches, and only a few parameters are needed to be optimized to obtain the best performance. In addition, it is worth noting that all the modules of BUIMS-TCN are divisible and plug-and-play; hence, our modules can be easily transplanted for other tasks.

3.1 Smoothed dilated 1D convolution

Dilated convolutions [26] can effectively expand the receptive field of the given convolution filter without increasing the number of parameters and computational costs. Dilated convolution has received extensive attention in the field of deep learning, but Wang and Ji [29] pointed that there are no dependencies among the input or output units in a dilated convolution because neighboring units in the next layer are

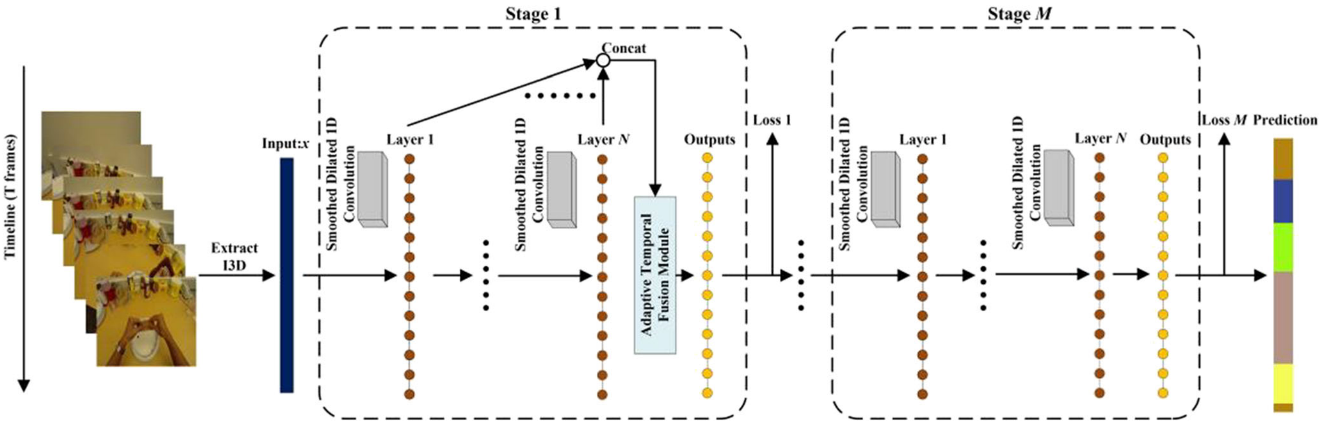

Timeline (T frames)
Extract I3D
Input:x
Stage 1
Concat
Smoothed Dilated 1D Convolution
Layer 1
Smoothed Dilated 1D Convolution
Layer N
Adaptive Temporal Fusion Module
Outputs
Loss 1
Stage M
Smoothed Dilated 1D Convolution
Layer 1
Smoothed Dilated 1D Convolution
Layer N
Outputs
Loss M
Prediction

Fig. 3 Overview of BUIMS-TCN. Given an untrimmed video, we extract inflated 3D (I3D) features and input them into BUIMS-TCN. Each stage of BUIMS-TCN is composed of several SCs, and a loss layer is added after each stage. The ATFM is only embedded in the first stage, and the prediction results are obtained in the final stage

related to totally different sets of units in the previous layer, which leads to the problem of grid artifacts. In dense prediction tasks such as image segmentation, grid artifacts will result in losing local spatial information and lacking contextual information during training.

The recently proposed MS-TCN is composed of a series of dilated 1D convolutions. We believe that grid artifacts also exist in action segmentation based on MS-TCN, and we call them temporal grid artifacts. In the previous sections, we have clearly illustrated that temporal grid artifacts are one of the main factors that cause the ambiguous frame problem. Hence, to alleviate the temporal grid artifacts, we propose a smoothed dilated 1D convolution (SC) operation. In the following text, we first briefly introduce dilated 1D convolution and then give the details of our proposed SC.

For a dilated 1D convolution with a filter w of size k, its output Z at the position i is defined as:

$$Z[i] = \sum_{s=1}^{k} f[i + r \times s] w[i] \tag{1}$$

where f represents the 1D input and r represents the dilation rate. When $r = 1$, the dilated 1D convolution degenerates into a standard 1D convolution. To intuitively understand dilated 1D convolution, we can regard the dilated 1D convolution as inserting $r - 1$ zeros between the two adjacent weights of w in the 1D convolution. Therefore, the receptive field of the dilated 1D convolution becomes $r \times (k - 1) + 1$.

Inspired by [29], we adopt separable and shared operations to smooth the dilated 1D convolution to enhance the dependencies between the local temporal features, which can effectively alleviate the temporal grid artifacts. “Separable” refers to the separable convolution from [67], and “shared” means that the weights in the convolutions for all channels are shared [29]. Specifically, a separable and shared convolution with a kernel size of $(2r - 1)$ is inserted before the dilated 1D convolution, thereby adding temporal dependencies among the feature maps produced by periodic subsampling [29]. The smoothing operation involves only a constant parameter $(2r - 1)$ that is independent of the number of channels so that the increased computational cost can be ignored. Figure 4 shows the structure of one SC with the kernel size of 3 and the dilation rate of 2.

Next, we further introduce the details of separable and shared operations in the SC. Figure 5 shows the separable and shared operations, and Fig. 6 shows the “smoothing process” of the SC.

In Fig. 5, we take the inputs and outputs of 4 feature channels as an example. The separable convolution operation includes 4 different 1D convolutions in total, with one convolution for each channel, while the separable and shared convolution operation only includes one 1D convolution shared among all channels.

In Fig. 6, the white boxes in the dilated 1D convolutions indicate that the weight values of those positions are 0, which means that the input features covered by white boxes will not be computed when using dilated 1D convolutions to process the input features. Hence, the output features will lose some local temporal information. For example, as shown in Fig. 6a, the output feature marked by the blue dotted circle does not contain any information from the input feature in the neighboring position marked by a magenta dotted circle. This is because when we use a dilated 1D convolution to process the input features to obtain the output feature marked by the blue dotted circle, the feature marked by the magenta dotted circle is covered by 0 and does not participate in the convolution operation.

To avoid losing local temporal information, we add a separable and shared convolution operation before the dilated 1D convolution to generate SCs, as shown in Fig. 6b. We also take the input feature marked by the magenta dotted circle as an example. Since the input features are first processed by a separable and shared convolution with a kernel size of 3, the feature information marked by the magenta dotted circle is smoothly transferred to the intermediate output feature marked by the green dotted circle. Then, the intermediate

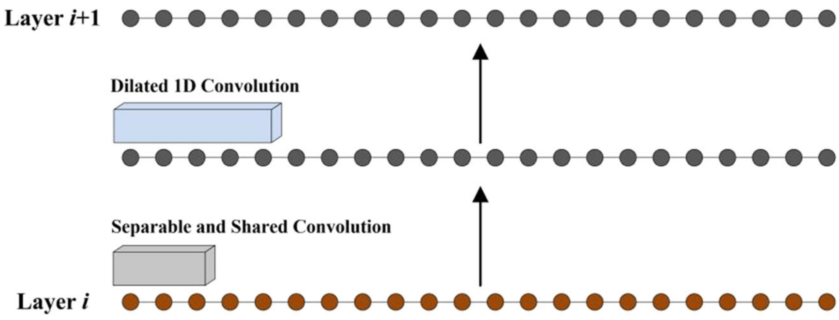


Fig. 4 Structure of the SC. The SC adds separable and shared convolutions before the dilated 1D convolution. The gray circles represent the smoothed feature maps, and the brown circles represent the original feature maps

output feature is further processed by a dilated 1D convolution, which is the same as the dilated 1D convolution in Fig. 6a, to obtain the final output feature. Therefore, the output feature marked by the blue dotted circle contains the feature information marked by the magenta dotted circle because the feature marked by a green dotted circle exerts the role of information transfer. That is, the feature at the nonzero position contains the local temporal information from its neighboring zero position when using the SCs, which effectively alleviates the loss of local temporal information and enhances the long-range temporal dependencies.

In the field of signal processing, signal filtering is the process of extracting the part of the signal that we are interested in, which aims to filter noise or false components in the signal, improve the signal-to-noise ratio, smooth signal, and so on. The “smooth” in our paper is similar to the signal filtering. For example, in Fig. 6b, the “Intermediate Output Features” marked by the green dotted circle is obtained by weighted summing itself (the corresponding position of “Input Features”) and other features (e.g., the feature marked by the magenta dotted circle) in the neighborhood through a separable and shared convolution with a kernel size of 3, which is similar to the signal filtering operation (e.g., Gaussian filter). Thus, it can maintain more useful information and avoid the information loss (e.g., the features marked by the magenta dotted circle will be lost) brought by directly adopting the dilated 1D convolution to process the input features. That is, the “smooth” can improve the “signal-to-noise ratio” to some extent, thus it is beneficial for improving the action segmentation results.

3.2 Adaptive temporal fusion module

It is usually beneficial to fuse features from different scales for high-level semantic recognition tasks. MS-TCN fuses and passes multiscale temporal information backward through the residual connection structure. Yu et al. [31] pointed that the residual connection passes the high-frequency signal of the previous layer with the same weight, producing more serious temporal grid artifacts and thus also exacerbating the ambiguous frame problem.

Inspired by [32, 33], we propose a simple yet effective multiscale temporal context information fusion module named adaptive temporal fusion module (ATFM). ATFM introduces an attention mechanism to adaptively weight hierarchical temporal context features and effectively aggregate multiscale temporal information, which can reduce the transmission of invalid high-frequency signals and improve the fusion performance. Specifically, we first extract multiscale temporal context features and concatenate them as the input of the ATFM. Then, ATFM automatically assigns a temporal weight map to the temporal context features of each scale. Finally, ATFM aggregates these features to generate high-level semantic temporal context information. The implementation form is as follows:

$$o_n = f_n(o_{n-1}) \tag{2}$$

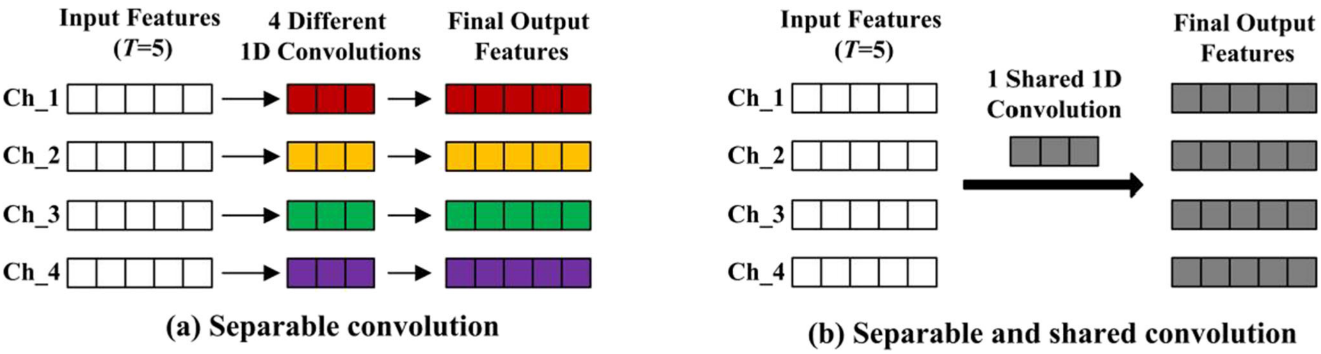
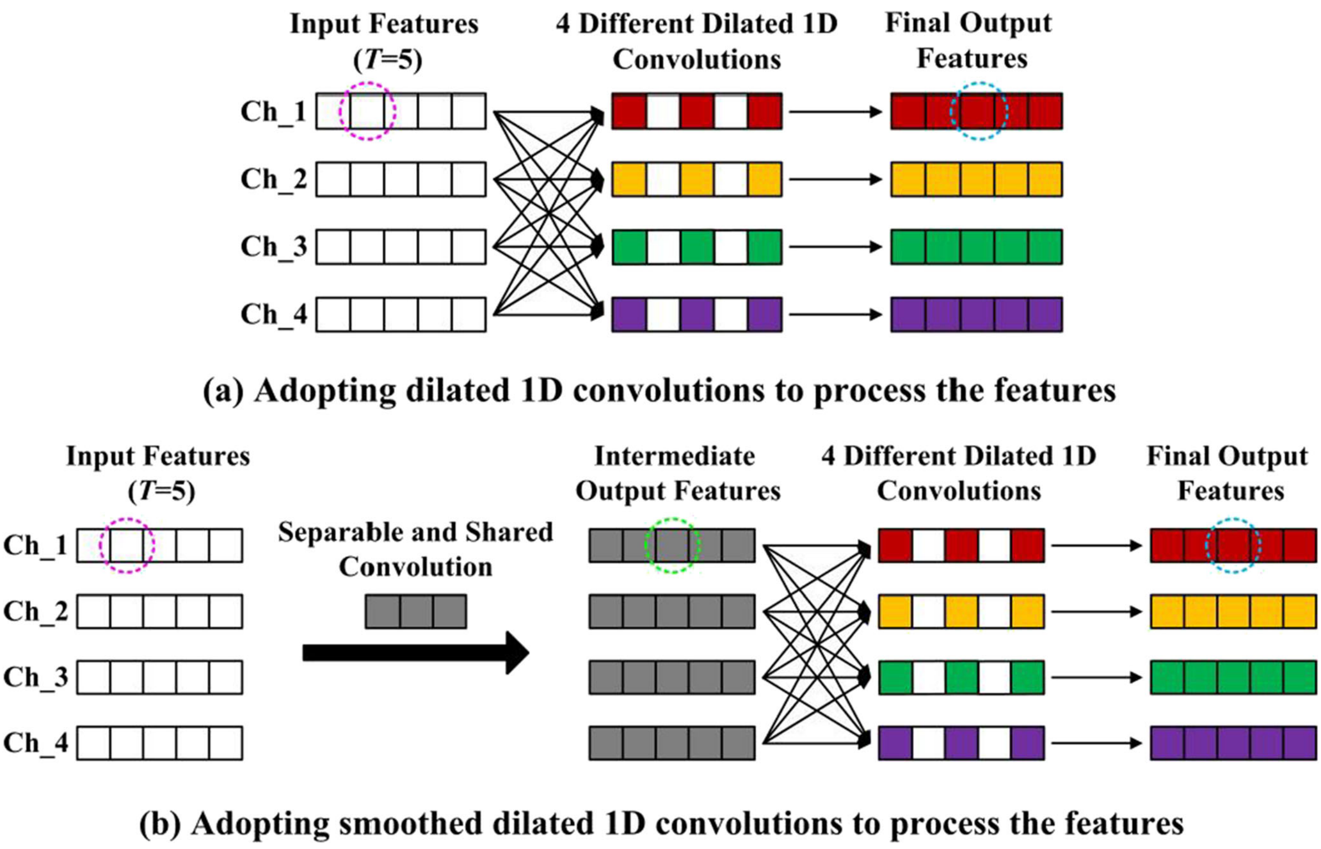


Fig. 5 Illustrations of the separable and shared operations. **a** shows the separable convolution proposed by [67]; **b** shows the separable and shared convolution proposed by [29]. Ch_*i* represents the *i*-th channel, and $T = 5$ indicates that there are 5 frames


Input Features
(T=5)
4 Different Dilated 1D Convolutions
Final Output Features
Ch_1
Ch_2
Ch_3
Ch_4
Input Features
(T=5)
Separable and Shared Convolution
Intermediate Output Features
4 Different Dilated 1D Convolutions
Final Output Features
Ch_1
Ch_2
Ch_3
Ch_4

(a) Adopting dilated 1D convolutions to process the features

(b) Adopting smoothed dilated 1D convolutions to process the features

Fig. 6 "Smoothing process" of the SC. **a** Dilated 1D convolutions with a kernel size of 3 and a dilation rate of 2 adopted to deal with the input features (5 frames and 4 channels); **b** the corresponding SC, which contains an additional separable and shared convolution operation, utilized to deal with the same input features. Ch_i represents the i-th channel, and $T = 5$ indicates that there are 5 frames

$$H(x) = E(g(o_1, o_2, \dots, o_n), \quad \mathrm{n}\in N) \tag{3}$$

where $o_0 = x$ and x represent the input features (I3D features) of the network; o_n represents the output of the n-th dilated 1D convolutional layer; f_n represents the convolution operation of the corresponding layer; H represents the output result of the ATFM; E represents the adaptively weighted fusion operator; g represents the concatenation operator; and N represents the number of dilated 1D convolutional layers in each stage. The temporal feature sizes of all scales are ($P \times Q$), P represents the number of feature channels, and Q represents the number of frames.

As shown in Fig. 7, the ATFM module consists of a 1D convolution with a kernel size of 1, a rectified linear unit (ReLU), a 1D convolution with a kernel size of 3, an activation function (sigmoid), and other additional operations. ATFM is a plug-and-play module with few parameters. It is worth mentioning that we only add ATFM into the first stage since ATFM is designed to fuse the multiscale temporal context information. In our proposed BUIMS-TCN, the information input into the first stage is the semantic features, while the information input into the subsequent stages is the softmax values of the output of the previous stage. Thus, we can obtain the best action segmentation results when only integrating ATFM into the first stage. In the experiments, we also prove that only integrating ATFM into the first stage of BUIMS-TCN can greatly alleviate the ambiguous frame problem to achieve the best performance.

3.3 Loss function

We explore the relationship between MS-TCN's underlying structure and the ambiguous frame problem in a bottom-up manner, that is, from the convolution operation to the residual connection structure and finally to the loss function.

As shown in Fig. 1, the ambiguous frames are difficult to recognize, and the number of ambiguous frames in long untrimmed videos is usually far less than the number of informative frames that are easy to recognize. The design of the loss function of MS-TCN ignores the problem of the imbalance between the numbers of ambiguous and informative frames. This leads to a large number of easily classified informative frames being used to train the classifier model, which thereby overwhelms the cross-entropy loss and dominates the gradient, causing the model to quickly converge to a locally optimal solution. To solve this problem, we introduce the focal loss function [34], which can solve the problem of

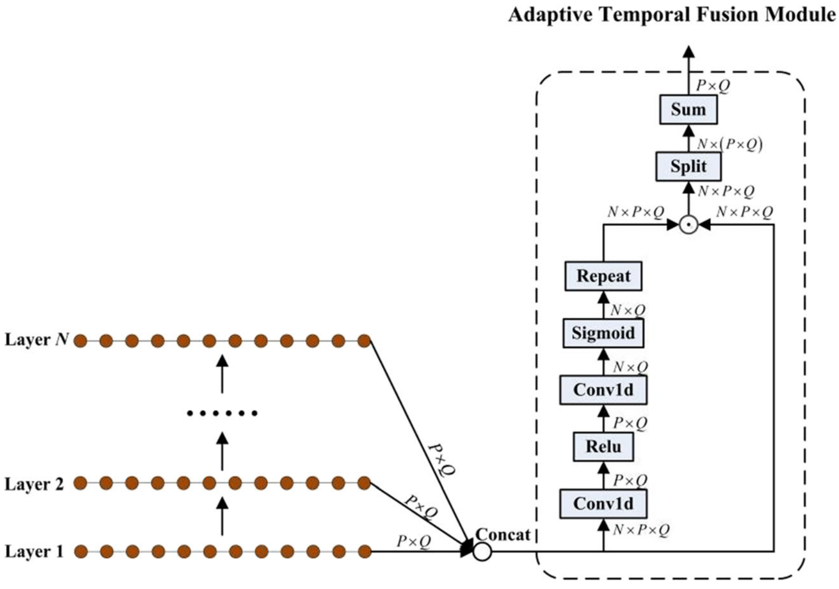


Fig. 7 Illustration of the ATFM module. ATFM introduces the attention mechanism to adaptively weight multiscale temporal context features and output a high-level semantic feature map

sample imbalance by reshaping the standard cross-entropy loss to reduce the weight of informative frames during the training process.

To clearly illustrate the impact of focal loss, we first explain in detail how hard and easy samples are defined. Hard samples are frames with large classification errors; that is, hard samples are frames that are difficult to correctly classify. In contrast, easy samples are frames that are easy to correctly classify (in our paper, easy samples are also called informative frames, while hard samples are also called ambiguous frames). In the process of training, the smaller the predicted probability $y_{t,\ c}$ of a sample from class c, the harder it is to classify correctly. Therefore, samples from class c with a small predicted probability $y_{t,\ c}$ are defined as hard samples, while samples from class c with a large $y_{t,\ c}$ are regarded as easy samples. In the following text, we will provide a further explanation of focal loss to illustrate why it can reduce the weight of easy samples during the training process.

The focal loss is proposed based on the cross-entropy loss. The cross-entropy loss L_{cls} [24] can be defined as:

$$L_{cls} = \frac{1}{T}\sum_{t=1}^{T} -log\left(y_{t,c}\right) \tag{4}$$

where $y_{t,\ c}$ represents the model's predicted probability for class c at time t; T represents the length of the video; and C represents the number of classes.

The cross-entropy loss ignores the class imbalance. To address this problem, a weighting factor $\alpha \in [0,\ 1]$ is introduced for different classes [34], and the α-balanced cross-entropy loss can be described as:

$$L'_{cls} = \frac{1}{T}\sum_{t=1}^{T} -\alpha log\left(y_{t,c}\right) \tag{5}$$

The α-balanced cross-entropy loss is a simple extension of cross-entropy loss. Although this loss balances the importance of samples in the different classes, it does not distinguish between easy and hard samples [34].

To reduce the weight of easy samples and thus focus training on hard samples, a modulating factor $(1 - y_{t,\ c})^{\gamma}$ is further introduced to generate the focal loss L_{fl} [34]:

$$L_{fl} = -\alpha\left(1-y_{t,c}\right)^{\gamma} log\left(y_{t,c}\right) \tag{6}$$

where γ is a tunable focusing parameter and $\gamma \geq 0$. γ is used to smoothly adjust the rate at which easy samples are downweighted. In the experiments, γ is set to 2 because this value can achieve the best performance.

In the focal loss, when a sample is misclassified and $y_{t,\ c}$ is small, the modulating factor $(1 - y_{t,\ c})^{\gamma}$ is close to 1, which means that the modulating factor does not affect the loss value. Conversely, when $y_{t,\ c}$ is large and near 1, the modulation factor $(1 - y_{t,\ c})^{\gamma}$ is close to 0, which means that the loss value of well-classified easy samples is downweighted. That is, the focal loss can reduce the weights of easy samples during the training process.

Different loss functions consider the different aspects of the model (e.g., the cross-entropy loss focuses on optimizing all temporal sequence frames which can obtain a good framewise classification accuracy, while the focal loss focuses on optimizing ambiguous frames, which can reduce over-segmentation errors), so combining them can ensure that the model converges to the global optimum. Therefore, to further solve the ambiguous frame problem and obtain the best action segmentation performance, we fuse the original loss functions (cross-entropy loss L_{cls} and smoothing loss $L_{T\ -\ MSE}$) of MS-TCN and the focal loss as the final loss function in our proposed BUIMS-TCN. Thus, the final loss function has the following form:

$$L = L_{cls} + \lambda L_{T-MSE} + L_{fl} \tag{7}$$

where L_{cls} is the cross-entropy loss (as shown in Eq. (4)); L_{T-MSE} is the smoothing loss of MS-TCN, which is defined in Eqs. (8)–(10); and λ is a hyperparameter that sets the weight of L_{T-MSE}.

$$L_{T-MSE} = \frac{1}{TC}\sum_{t,c}\widetilde{\Delta}_{t,c}^{2} \tag{8}$$

$$\widetilde{\Delta}_{t,c} = \begin{cases} \Delta_{t,c} : & \Delta_{t,c} \leq \tau \\ \tau \quad : & \text{otherwise} \end{cases} \tag{9}$$

$$\Delta_{t,c} = \left| logy_{t,c} - logy_{t-1,c} \right| \tag{10}$$

where τ is a hyperparameter used to truncate the smoothing loss L_{T-MSE}.

4 Experiments

4.1 Experimental setup

4.1.1 Implementation setting

We use PyTorch to implement the proposed BUIMS-TCN. BUIMS-TCN has 4 stages, and each stage includes 10 dilated convolution layers. The dilation rate of each layer is double that of the previous layer. Dropout is used after each layer and the probability is set to 0.5. The number of convolutional patches in each convolution layer is 64, and the size of the convolution kernel is 3. The parameter λ in Eq. (7) is set to 0.15, and the parameter τ in Eq. (9) is set to 4. We set the values of these hyperparameters in our proposed BUIMS-TCN to the same as those in MS-TCN, which aims to better compare the performance of MS-TCN and BUIMS-TCN. In addition, in focal loss L_{fl}, we set $\gamma = 2$, and α is selected according to the dataset. In all experiments, we use the Adam optimizer with a learning rate of 0.0005 without weight decay.

4.1.2 Datasets

We utilize three challenging benchmark datasets, the GTEA [35], 50Salads [36], and Breakfast [37] datasets, to evaluate the performance of the proposed BUIMS-TCN.

50Salads is constructed by recording salad-making activities performed by 25 actors. This dataset is composed of 50 videos corresponding to 17 action classes. Each video is approximately 6.4 min long and contains 9000 to 18,000 frames and an average of 20 action instances. We perform 5-fold cross-validation on the 50Salads dataset and report the average action segmentation results for evaluation.

GTEA contains 28 egocentric videos of 7 daily activities performed by 4 subjects. The videos in this dataset are recorded by a camera mounted on the actor's head. On average, each video contains 11 action classes (including the background class) and 20 action instances. We perform 4-fold cross-validation on the GTEA dataset and compute the average action segmentation results for evaluation.

The **Breakfast** dataset is the largest one in these three datasets. This dataset includes 1712 videos with a total duration of 66.7 h and a total of 48 different action classes. The videos in this dataset are the record of people cooking breakfast in 18 different kitchens. On average, each video contains 6 action instances. We perform 4-fold cross-validation on the Breakfast dataset and calculate the average action segmentation results for evaluation.

In all datasets, the temporal video resolution is 15 fps, and the input of our proposed BUIMS-TCN is the inflated 3D (I3D) [10] features of the video frames.

4.1.3 Evaluation metrics

We use the following evaluation metrics: the framewise accuracy (**Acc**), segmental edit distance (**Edit**) and segmental F1 score (**F1@{10, 25, 50}**) at the temporal intersection over union (tIoU) thresholds 10%, 25% and 50% to evaluate the action segmentation results. Acc is commonly used to evaluate the framewise accuracy, which is not sensitive to over-segmentation errors; thus, even when the segmentation results have not the temporal continuity (e.g., the predicted labels of frames within one action segment are not the same), high Acc scores can still be achieved. Therefore, in addition to Acc, we also use Edit [18] and segmental F1 score [68] to evaluate over-segmentation errors. The larger the Acc, Edit and F1 score are, the better the action segmentation results.

4.2 Results and analysis

4.2.1 Quantitative analysis

In this section, we compare our proposed BUIMS-TCN with the baseline and state-of-the-art methods on three challenging benchmark datasets: the GTEA, 50Salads, and Breakfast datasets. The comparison results are shown in Tables 1, 2, 3.

From Tables 1, 2, 3, it can be seen that our model outperforms the other methods on all datasets and evaluation metrics, except for Edit on the 50Salads dataset. For the relatively small GTEA and 50Salads datasets, two metrics (the F1 score and Edit) are used to evaluate the action segmentation results. Since MS-TCN is the basic structure, comparing it directly to our model can enable us to intuitively see the performance improvement gained by our BUIMS-TCN. From Tables 1 and 2, it can be seen that both the F1 score and Edit of our BUIMS-TCN are higher (up to 3.6% and 6.1% for Edit, up to 1.9% and 4.8% for F1@10, up to 1.2% and 5.8% for F1@25, and up to 2.0% and 7.9% for F1@50) than those of MS-TCN on the GTEA and 50Salads datasets. Table 3 shows that on the largest dataset Breakfast, compared to MS-TCN, the F1 score and Edit of our BUIMS-TCN are highly improved (by up to 8.5% for Edit, up to 18.4% for F1@10, up to 17.1% for F1@25 and up to 12.7% for F1@50). The results in Tables 1, 2, 3 verify the effectiveness and rationality of our BUIMS-TCN for solving the ambiguous frame problem. In addition, from Tables 1, 2, 3, we can also see that the scale and recognition difficulty of GTEA, 50Salads, and Breakfast gradually increase, but our BUIMS-TCN gains the largest performance improvement compared to MS-TCN on the Breakfast dataset. That is, the more difficult the segmentation task is, the better the performance obtained by our proposed BUIMS-TCN.

It should be mentioned that Edit measures whether the ordering of predicted action instances is the same as that in the ground truth, but it does not consider the framewise accuracy and the specific timings of the action boundaries. The Edit result of our BUIMS-TCN on the 50Salads dataset is not optimal potentially because the number of action instances in each video in 50Salads is greater than those in the other two datasets, resulting in the videos in 50Salads containing more combinations of action sequences. This requires the network to have more advanced reasoning capabilities to predict instance-level actions, but MS-TCN is only a frame-level action reasoning network. And when we construct our BUIMS-TCN based on the architecture of MS-TCN, all improvements that we made aim at solving the ambiguous frame problem, without considering the problem of instance-level ambiguous actions. That is, we have not focused on how to improve the accuracy of the sequence of predicted action instances. In the future, we will try to solve instance-level action modeling rather than only frame-level action modeling.

Table 1 Comparing our BUIMS-TCN with the state-of-the-art methods on GTEA

GTEA	F1@{10, 25, 50}			Edit	Acc
Spatial CNN [21]	41.8	36.0	25.1	–	54.1
Bi-LSTM [43]	66.5	59.0	43.6	–	55.5
Dilated TCN [18]	58.8	52.2	42.2	–	58.3
TUnet [55]	67.1	63.7	51.9	60.3	59.9
ST-CNN [21]	58.7	54.4	41.9	–	60.6
ED-TCN [18]	72.2	69.3	56.0	–	64.0
LCDC+ED-TCN [47]	75.4	–	–	72.8	65.3
TResNet [69]	74.1	69.9	57.6	64.4	65.8
TRN [41]	77.4	71.3	59.1	72.2	67.8
TDRN+UNet [41]	78.1	73.8	62.2	73.7	69.3
TDRN [41]	79.2	74.4	62.7	74.1	70.1
MS-TCN [24]	87.5	85.4	74.6	81.4	79.2
MS-TCN++ [17]	88.8	85.7	76.0	83.5	80.1
BUIMS-TCN	**89.4**	**86.6**	**76.6**	**85.0**	**80.6**

Table 2 Comparing our BUIMS-TCN with the state-of-the-art methods on 50Salads

50Salads	F1@{10, 25, 50}			Edit	Acc
Spatial CNN [21]	32.3	27.1	18.9	24.8	54.9
IDT+LM [46]	44.4	38.9	27.8	45.8	48.7
Bi-LSTM [43]	62.6	58.3	47.0	55.6	55.7
Dilated TCN [18]	52.2	47.6	37.4	43.1	59.3
ST-CNN [21]	55.9	49.6	37.1	45.9	59.4
TUnet [55]	59.3	55.6	44.8	50.6	60.6
ED-TCN [18]	68.0	63.9	52.6	52.6	64.7
TResNet [69]	69.2	65.0	54.4	60.5	66.0
TDRN+UNet [41]	69.6	65.0	53.6	62.2	66.1
TRN [41]	70.2	65.4	56.3	63.7	66.9
TDRN [41]	72.9	68.5	57.2	66.0	68.1
LCDC+ED-TCN [47]	73.8	–	–	66.9	72.1
MS-TCN [24]	76.3	74.0	64.5	67.9	80.7
MS-TCN++ [17]	80.7	78.5	70.1	**74.3**	83.7
BUIMS-TCN	**81.1**	**79.8**	**72.4**	74.0	**83.9**

To determine the impact of the input features, we compare our proposed BUIMS-TCN with MS-TCN on the GTEA dataset based on the different I3D features extracted by the I3D model with and without fine-turning. Fine-tuning means that we use the I3D model fine-tuned on GTEA [24] to extract the I3D features of videos in the GTEA dataset and apply these extracted features as the input of each action segmentation network. The comparison results are shown in Table 4, which shows that our model is better than MS-TCN regardless of whether fine-tuned I3D features are utilized.

Table 3 Comparing our BUIMS-TCN with the state-of-the-art methods on the Breakfast dataset (* obtained from [42])

Breakfast	F1@{10, 25, 50}			Edit	Acc
ED-TCN [18]*	–	–	–	–	43.3
HTK [48]	–	–	–	–	50.7
TCFPN [42]	–	–	–	–	52.0
HTK (64) [23]	–	–	–	–	56.3
GRU [44]*	–	–	–	–	60.6
GRU+length prior [45]	–	–	–	–	61.3
MS-TCN [24]	52.6	48.1	37.9	61.7	66.3
MS-TCN++ [17]	64.1	58.6	45.9	65.6	67.6
BUIMS-TCN	**71.0**	**65.2**	**50.6**	**70.2**	**68.7**

4.2.2 Qualitative analysis

To further verify the performance of our proposed BUIMS-TCN, we visualize the results obtained by BUIMS-TCN and MS-TCN, as shown in Fig. 8. From Fig. 8, it can be seen that in the action segmentation results of our BUIMS-TCN, the duration of predicted action of one class is more complete, and the actions from different classes are better distinguished. This demonstrates that the ambiguous frame problem is well solved by our BUIMS-TCN. It should be mentioned that BUIMS-TCN has no additional labor costs, complicated structures, or additional branches, and it creates only a minor extra computational burden compared to MS-TCN; thus, BUIMS-TCN can maintain both effectiveness and efficiency during the processes of training and testing. For example, training our BUIMS-TCN for 50 epochs on a single GTX TitanXp GPU only requires approximately 12 min for the 50Salads dataset, but BUIMS-TCN can greatly improve the performance.

4.2.3 Structure analysis and efficiency comparison

To illustrate the differences between our proposed BUIMS-TCN and other action segmentation networks proposed based on MS-TCN (e.g., BCN [1], SSTDA [16], ASRF [25] and MS-TCN++ [17]), we first briefly introduce each network and then compare the structures of these different networks.

BCN [1] has a dual-branch structure containing a stage cascade and a barrier generation module to solve the ambiguous frame problem. The stage cascade repeatedly inputs the features into each stage and fuses the output features of all stages. The barrier generation module uses a multilayer convolutional layer to compute the boundary confidence. This dual-branch structure increases not only the number of input features but also the number of parameters and calculations. SSTDA [16] proposes two self-supervised auxiliary tasks (binary and sequence domain prediction). SSTDA increases the number of input features, and it adds additional structures and operations in two identical MS-TCN structures for two auxiliary tasks, which increases the required parameters and calculations, the difficulty of network convergence, and the number of required training epochs. ASRF [25] consists of a long-term feature extractor and two branches (an action segmentation branch and a boundary regression branch). ASRF requires manual annotate the ground truth of the action boundary in the video for training the model. In addition, ASRF uses two identical MS-TCN structures as the dual-branch structure, which doubles the number of parameters and calculations. MS-TCN++ [17] modifies the first stage of MS-TCN. Specifically, a dilated 1D convolution in each layer is replaced with two differently scaled dilated 1D convolutions, which increases the number of parameters and calculations exponentially.

Comparing the structures of our proposed BUIMS-TCN with the abovementioned networks, we can conclude that: 1) Our BUIMS-TCN is proposed based on a single MS-TCN with only minor modifications, while SSTDA and ASRF employ two MS-TCNs and add additional structures/operations to the MS-TCNs, which not only increases the number of input features and requires manual annotation of the boundary labels (e.g., ASRF) but also highly increases the number of parameters, calculations and the training time, as shown in Table 5. 2) Although BUIMS-TCN, BCN and MS-TCN++ are constructed based on a single MS-TCN, the structures of BCN and MS-TCN++ are more complex. BCN is a dual-branch structure composed of a stage cascade branch and a barrier generation branch, while our BUIMS-TCN is a relatively simple structure that contains only one branch. MS-TCN++ increases the number of convolution layers and adopts two convolution kernels with different sizes in each layer, while the number of convolution layers in our BUIMS-TCN is the same as that in MS-TCN. In addition, the parameters, calculations, and training time of BCN and MS-TCN++ are larger than those of our BUIMS-TCN, as shown in Table 5. 3) When comparing all abovementioned networks to MS-TCN, only our BUIMS-TCN requires almost the same parameters, calculations and training time as that of MS-TCN, as shown in Table 5. That is, BUIMS-TCN can improve the action segmentation performance of MS-TCN and still maintain the same efficiency as MS-TCN.

Table 5 lists the numbers of parameters, calculations (FLOPs) and the training times of the different networks. "Training time" in Table 5 is the runtime (minutes) of training 50 epochs on the 50Salads dataset by a single GTX TitanXp GPU. From Table 5, it can be found that our BUIMS-TCN is significantly more efficient than BCN, SSTDA, ASRF and MS-TCN++. For instance, regarding FLOPs, MS-TCN++ is approximately 2 times our model, ASRF is approximately 6 times our model, SSTDA is approximately 32 times our model, and BCN is approximately 38 times our model. Regarding the number of parameters, MS-TCN++ is slightly larger than our model, ASRF is approximately 3 times our mode, SSTDA is approximately 6 times our mode, and BCN is approximately 7 times our mode. Regarding the training time, our BUIMS-TCN is the same as the baseline (MS-TCN) and less than all other networks.

Table 4 Effect of fine-tuning on the GTEA dataset

		F1@{10, 25, 50}			Edit	Acc
w/o FT	MS-TCN [24]	85.8	83.4	69.8	79.0	76.3
	BUIMS-TCN	**88.5**	**86.5**	**76.0**	**82.4**	**78.1**
with FT	MS-TCN [24]	87.5	85.4	74.6	81.4	79.2
	BUIMS-TCN	**89.4**	**86.6**	**76.6**	**85.0**	**80.6**

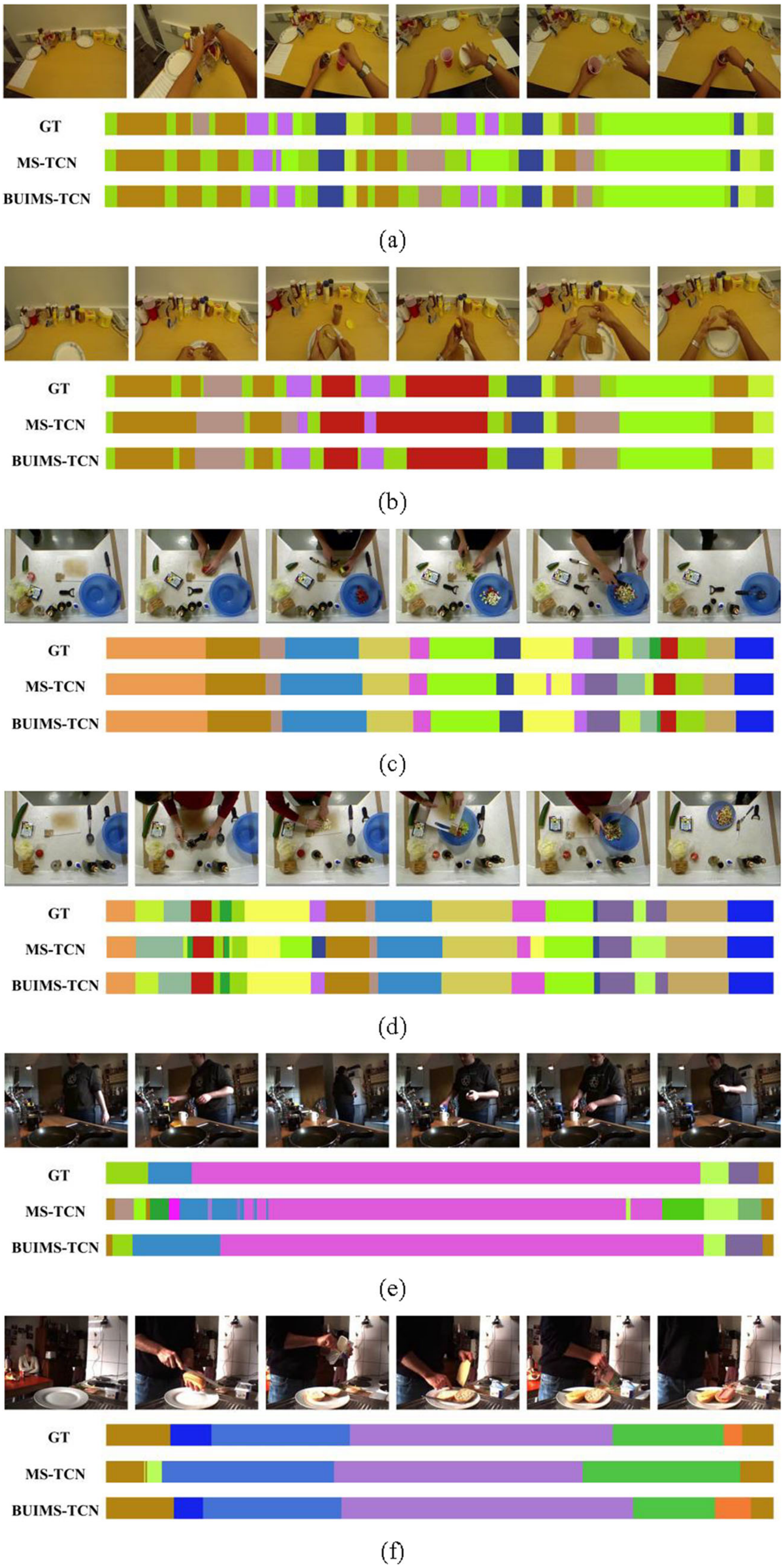


Fig. 8 Qualitative results of temporal action segmentation for three datasets: **a**, **b** GTEA, **c**, **d** 50Salads, and **e**, **f** the Breakfast dataset

Table 5 Numbers of parameters, FLOPs and training times of different networks on 50Salads

Model	Parameters	FLOPs	Training Time
BCN [1]	5.72 M	40.9 M	23 Mins
SSTDA [16]	4.60 M	34.5 M	21 Mins
ASRF [25]	2.31 M	6.56 M	15 Mins
MS-TCN++ [17]	1.00 M	2.16 M	13 Mins
BUIMS-TCN	**0.82 M**	**1.08 M**	**12 Mins**
MS-TCN [24]	0.80 M	0.74 M	12 Mins

4.2.4 Ablation study analysis

In this section, we analyze the performances of the SC, ATFM and loss function. To ensure that all comparisons are fair, we strictly follow the structure and parameter settings of MS-TCN (basic structure), gradually add different modules to the basic structure to construct new networks, and provide the corresponding action segmentation results of each new network.

The detailed experimental settings of the ablation study are as follows. 1) **Testing the performance of the SC:** The SC is used in different stages of the basic structure. 2) **Testing the performance of the ATFM:** An ATFM is added to the basic structure with the SC. 3) **Testing the performance of the loss function:** The focal loss (L_{fl}) is introduced into the optimal structure constructed by adding both the SC and ATFM, and various combinations of L_{fl} and the original loss functions in the basic structure are compared. The ablation study maintains consistent results on the three datasets; thus, we only use the 50Salads dataset as an example to show the ablation study results, as shown in Tables 6, 7, 8, 9.

Table 6 shows that as the number of stages adding SC increases, the F1 score, Edit and Acc gradually improves. The best performance is achieved when SC is used in all four stages. This demonstrates that: 1) SC is effective in alleviating the ambiguous frame problem and can effectively establish the local temporal information dependencies and 2) adding SC in all four stages does not lead to model overfitting. Accordingly, this setting is adopted in the subsequent experiments.

Table 6 Results of testing SC on the 50Salads dataset

	F1@{10, 25, 50}			Edit	Acc
MS-TCN (w/o SC)	76.3	74.0	64.5	67.9	80.7
MS-TCN (1 stage with SC)	78.0	75.1	66.8	70.0	81.0
MS-TCN (2 stages with SC)	78.5	75.4	67.1	71.2	81.2
MS-TCN (3 stages with SC)	78.8	75.6	67.1	71.6	81.3
MS-TCN (4 stages with SC)	**79.0**	**75.7**	**67.3**	**71.8**	**81.4**

Table 7 Results of testing ATFM on the 50Salads dataset

	F1@{10, 25, 50}			Edit	Acc
MS-TCN+SC (w/o ATFM)	79.0	75.7	67.3	71.8	81.4
MS-TCN+SC (1 stage with ATFM)	**80.5**	76.6	**69.5**	**72.3**	**82.7**
MS-TCN+SC (2 stages with ATFM)	79.4	**76.8**	68.1	71.9	82.1
MS-TCN+SC (3 stages with ATFM)	77.3	75.2	68.2	70.6	82.6
MS-TCN+SC (4 stages with ATFM)	76.3	74.5	66.0	69.4	82.6

ATFM is a simple yet effective multiscale temporal context information fusion module. From Table 7, we can see that ATFM can further mitigate the ambiguous frame problem and improve the action segmentation performance. Specifically, ATFM achieves the best performance when it is added in the first stage. This is because only the features input into the first stage contain strong semantic information, while the features input into the subsequent stages, which are the softmax values of the output of the previous stage, generally do not include semantic information. Hence, only adding ATFM in the first stage can ensure that the model learns better semantic representations and achieves good performance. When ATFM is embedded in the subsequent stages, the model cannot capture any semantic information, which may even decrease the performance of the model. This setting (only

Table 8 Results of testing different α values on the 50Salads dataset

α	F1@{10, 25, 50}			Edit	Acc
0.10	79.9	77.6	70.7	72.5	83.2
0.15	**81.4**	79.6	71.4	74.2	82.9
0.30	80.0	78.3	69.8	72.0	82.0
0.50	79.6	77.5	68.8	72.6	82.5
0.75	**81.4**	78.2	70.1	74.6	82.2
0.90	81.1	79.2	71.1	**75.3**	82.4
0.95	81.1	**79.8**	**72.4**	74.0	**83.9**

Table 9 Results of testing the different loss function combinations on the 50Salads dataset

Loss Function	F1@{10, 25, 50}			Edit	Acc
L_{cls}	77.3	74.9	67.0	68.7	81.8
L_{fl}	76.0	72.7	63.5	70.2	78.2
$L_{cls}+L_{fl}$	79.4	77.4	69.4	72.0	82.6
$L_{cls}+\lambda L_{T-MSE}$	80.5	76.6	69.5	72.3	82.7
$L_{fl}+\lambda L_{T-MSE}$	71.4	67.4	55.6	63.6	74.4
$L_{cls}+\lambda L_{T-MSE}+L_{fl}$	**81.1**	**79.8**	**72.4**	**74.0**	**83.9**

adding ATFM in the first stage) is employed in the subsequent experiments.

Regarding the loss function, we add the focal loss L_{fl} to the previous optimal model structure with both SC and ATFM; thus, here, the loss function can be described as $L_{cls} + \lambda L_{T-MSE} + L_{fl}$ with the default setting λ =0.15. In the focal loss L_{fl}, the parameter α (shown in Eq. (7)) is used to balance the distribution of easy samples and hard samples. Hence, we first test the performance of the model with different α values and then choose an optimal α value for the subsequent experiments. Table 8 shows the action segmentation results with different α values. From Table 8, we can see that when α = 0.95, the best performance is achieved, which means that the proportion of hard samples in the 50Salads dataset is approximately 5%. This allows the model to focus on recognizing these hard samples in the process of training, which can effectively solve the ambiguous frame problem. For different datasets, the optimal α values are different. The sample distribution of GTEA is similar to that of 50Salads, both of them contain relatively few hard samples. The best results on the GTEA dataset are obtained when α = 0.90. The ambiguous frame problem of the Breakfast dataset is extremely serious, that is, the number of hard samples is relatively large; thus, the optimal result for this dataset can be obtained when α = 0.30.

To test the impact of the loss function on the performance of the model, we test different combinations of three losses: focal loss (L_{fl}), cross-entropy loss (L_{cls}) and smoothing loss (L_{T-MSE}). The testing results are shown in Table 9. In Table 9, we use the previous optimal model structure with both SC and ATFM, adopt α = 0.95 as a fixed parameter, and set the other parameters to be consistent with those of MS-TCN.

From Table 9, several conclusions can be obtained: 1) L_{cls} is superior to L_{fl}. This is because the former focuses on recognizing all temporal sequence frames (both ambiguous frames and informative frames), while the latter is biased toward recognizing ambiguous frames. Thus, only employing L_{fl} will cause the model to be trapped in a local optimum. 2) Combining L_{cls} with L_{fl} or L_{T-MSE} can further improve the action segmentation performance. This is because L_{cls} can achieve good framewise classification accuracy, while it ignores over-segmentation errors to some extent. In contrast, L_{fl} and L_{T-MSE} focus on identifying ambiguous frames and maintaining consistency between neighboring frames, which is beneficial for reducing over-segmentation errors. Hence, integrating L_{fl} or L_{T-MSE} with L_{cls} can greatly reduce over-segmentation errors (e.g., the values of the F1 score and Edit are greatly increased). 3) Combining L_{fl} and L_{T-MSE} will degrade the performance of the model. This is because both L_{fl} and L_{T-MSE} cannot guarantee framewise classification accuracy, which leads to worse segmentation results than those achieved by combining one of them with L_{cls}. In addition, combining L_{fl} and L_{T-MSE} is inferior to only utilizing L_{fl}. This may be because directly fusing L_{T-MSE} and L_{fl} will cause that L_{fl} cannot converge well. 4) The best performance is gained by fusing L_{cls}, L_{fl} and L_{T-MSE} as the loss function of the model. The reason for this is that the different loss functions are complementary, and combining them can ensure that the model converges to the global optimum.

Based on the results of the above ablation study, the optimal structure containing SC in all four stages, ATFM in the first stage, and a loss function of $L_{cls} + \lambda L_{T-MSE} + L_{fl}$ is selected as the final structure of BUIMS-TCN.

5 Conclusion

The classification and localization of action segments in long untrimmed videos are very important for understanding human activities. In this paper, we propose a new temporal action segmentation model named BUIMS-TCN. BUIMS-TCN is constructed by incorporating three improvements (a smoothed dilated 1D convolution, an ATFM and a new loss function) into MS-TCN's underlying structure. Different from the existing models, our proposed BUIMS-TCN not only avoids additional manual labeling and complicated network structures/branches but also maintains relatively high computational efficiency. The experimental results demonstrate that BUIMS-TCN can effectively address the ambiguous frame problem and achieve a state-of-the-art performance compared with the existing models, especially on the largest dataset, Breakfast. We hope that BUIMS-TCN can become a new and stronger backbone for action segmentation in the future.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10489-022-03382-x.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China under Grants 61907007 and 62107009, in part by the National Key R&D Program of China under Grant 2020YFA0714102, in part by the Fund of the Jilin Provincial Science and Technology Department under Grants 20210201077GX, 20200201199JC, 20200401086GX and 20200401081GX, and in part by the Fundamental Research Funds for the Central Universities under Grants 2412020FZ029, 2412020FZ031, and 2412019FZ049.

References

1. Wang Z, Gao Z, Wang L, Li Z, Wu G (2020) Boundary-Aware Cascade Networks for Temporal Action Segmentation. In: European Conference on Computer Vision (ECCV), pp 34–51
2. Ahmed M, Mahmood AN, Hu J (2016) A survey of network anomaly detection techniques. J Netw Comput Appl 60:19–31
3. Liu H, Fang S, Zhang Z, Li D, Lin K, Wang J (2021) MFDNet: collaborative poses perception and matrix fisher distribution for head pose estimation. IEEE Transactions on Multimedia:1–1
4. Li D, Liu H, Zhang Z, Lin K, Fang S, Li Z, Xiong NN (2021) CARM: confidence-aware recommender model via review

representation learning and historical rating behavior in the online platforms. Neurocomputing 455:283–296

5. Shen X, Yi B, Liu H, Zhang W, Zhang Z, Liu S et al (2021) Deep Variational matrix factorization with knowledge embedding for recommendation system. IEEE Trans Knowl Data Eng 33:1906–1918
6. Liu T, Liu H, Li Y, Zhang Z, Liu S (2019) Efficient blind signal reconstruction with wavelet transforms regularization for educational robot infrared vision sensing. IEEE/ASME Transactions on Mechatronics 24:384–394
7. Liu T, Liu H, Li Y, Chen Z, Zhang Z-l, Liu S (2020) Flexible FTIR spectral imaging enhancement for industrial robot infrared vision sensing. IEEE Transactions on Industrial Informatics 16:544–554
8. Xu B, Ye H, Zheng Y, Wang H, Luwang T, Jiang Y (2019) Dense dilated network for video action recognition. IEEE Trans Image Process 28:4941–4953
9. Feichtenhofer C, Pinz A, Wildes R (2016) Spatiotemporal residual networks for video action recognition. In: Advances in Neural Information Processing Systems (NIPS), pp 3468–3476
10. Carreira J, Zisserman A (2017) Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 4724–4733
11. Feichtenhofer C, Fan H, Malik J, He K (2019) SlowFast Networks for Video Recognition. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp 6201–6210
12. Zhang X, Huang Y, Mi Y, Pei Y, Zou Q, Wang S (2021) Video sketch: a middle-level representation for action recognition. Appl Intell 51:2589–2608
13. Yao G, Lei T, Zhong J, Jiang P (2018) Learning multi-temporal-scale deep information for action recognition. Appl Intell 49:2017–2029
14. Majd M, Safabakhsh R (2018) A motion-aware ConvLSTM network for action recognition. Appl Intell 49:2515–2521
15. Ding C, Liu K, Cheng F, Belyaev E (2020) Spatio-temporal attention on manifold space for 3D human action recognition. Appl Intell 51:560–570
16. Chen M-H, Li B, Bao SY-Z, Al-Regib G, Kira Z (2020) Action Segmentation With Joint Self-Supervised Temporal Domain Adaptation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 9451–9460
17. Li S, Farha YA, Liu Y, Cheng M-M, Gall J (2020) MS-TCN++: multi-stage temporal convolutional network for action segmentation. IEEE Trans Pattern Anal Mach Intell PP:1
18. Lea CS, Flynn MD, Vidal R, Reiter A, Hager G (2017) Temporal Convolutional Networks for Action Segmentation and Detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1003–1012
19. Gao J, Chen K, Nevatia R (2018) Ctap: Complementary temporal action proposal generation. In: European conference on computer vision (ECCV), pp 68–83
20. Hendry, Chen RC (2019) Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning. Image Vis. Comput 87:47–56
21. Lea C, Reiter A, Vidal R, Hager GD (2016) Segmental spatiotemporal CNNs for fine-grained action segmentation. In: European Conference on Computer Vision (ECCV), pp 36–52
22. Richard A, Kuehne H, Gall J (2018) Action Sets: Weakly Supervised Action Segmentation Without Ordering Constraints. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 5987–5996
23. Kuehne H, Gall J, Serre T (2016) An end-to-end generative framework for video segmentation and recognition. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), pp 1–8
24. Farha YA, Gall J (2019) MS-TCN: Multi-Stage Temporal Convolutional Network for Action Segmentation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 3570–3579
25. Ishikawa Y, Kasai S, Aoki Y, Kataoka H (2021) Alleviating Over-segmentation Errors by Detecting Action Boundaries. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), pp 2321–2330
26. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille A (2018) DeepLab: semantic image segmentation with deep convolutional nets, Atrous convolution, and fully connected CRFs. IEEE Trans Pattern Anal Mach Intell 40:834–848
27. Wang P, Chen P, Yuan Y, Liu D, Huang Z, Hou X et al (2018) Understanding Convolution for Semantic Segmentation. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp 1451–1460
28. Hamaguchi R, Fujita A, Nemoto K, Imaizumi T, Hikosaka S (2018) Effective Use of Dilated Convolutions for Segmenting Small Object Instances in Remote Sensing Imagery. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp 1442–1450
29. Wang Z, Ji S (2021) Smoothed dilated convolutions for improved dense prediction. Data Min Knowl Disc 35:1–27
30. Wu T, Tang S, Zhang R, Cao J, Li J (2019) Tree-Structured Kronecker Convolutional Network for Semantic Segmentation. In: 2019 IEEE International Conference on Multimedia and Expo (ICME), pp 940–945
31. Yu F, Koltun V, Funkhouser T (2017) Dilated Residual Networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 636–644
32. L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," ArXiv, vol. abs/1706.05587, 2017
33. Guo C, Fan B, Zhang Q, Xiang S, Pan C (2020) AugFPN: Improving Multi-Scale Feature Learning for Object Detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 12592–12601
34. Lin T-Y, Goyal P, Girshick RB, He K, Dollár P (2020) Focal loss for dense object detection. IEEE Trans Pattern Anal Mach Intell 42: 318–327
35. Fathi A, Ren X, Rehg JM (2011) Learning to recognize objects in egocentric activities. In: 2011 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 3281–3288
36. Stein S, McKenna S (2013) 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, pp 729–738
37. Kuehne H, Arslan AB, Serre T (2014) The Language of Actions: Recovering the Syntax and Semantics of Goal-Directed Human Activities. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp 780–787
38. Liu H, Nie H, Zhang Z, Li Y (2021) Anisotropic angle distribution learning for head pose estimation and attention understanding in human-computer interaction. Neurocomputing 433:310–322
39. Li Z, Liu H, Zhang Z, Liu T, Xiong NN (2021) Learning knowledge graph embedding with heterogeneous relation attention networks. IEEE Transactions on Neural Networks and Learning Systems
40. Zhang Z, Li Z, Liu H, Xiong NN (2020) Multi-scale dynamic convolutional network for knowledge graph embedding. IEEE Ann Hist Comput:1
41. Lei P, Todorovic S (2018) Temporal Deformable Residual Networks for Action Segmentation in Videos. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 6742–6751
42. Ding L, Xu C (2018) Weakly-Supervised Action Segmentation with Iterative Soft Boundary Assignment. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 6508–6516

43. Singh B, Marks TK, Jones MJ, Tuzel O, Shao M (2016) A Multi-stream Bi-directional Recurrent Neural Network for Fine-Grained Action Detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1961–1970
44. Richard A, Kuehne H, Gall J (2017) Weakly Supervised Action Learning with RNN Based Fine-to-Coarse Modeling. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1273–1282
45. Kuehne H, Richard A, Gall J (2020) A hybrid RNN-HMM approach for weakly supervised temporal action segmentation. IEEE Trans Pattern Anal Mach Intell 42:765–779
46. Richard A, Gall J (2016) Temporal Action Detection Using a Statistical Language Model. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 3131–3140
47. Mac K-NC, Joshi D, Yeh RA, Xiong J, Feris R, Do M (2019) Learning Motion in Feature Space: Locally-Consistent Deformable Convolution Networks for Fine-Grained Action Detection. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp 6281–6290
48. Kuehne H, Richard A, Gall J (2017) Weakly supervised learning of actions from transcripts. Comput Vis Image Underst 163:78–89
49. Li Y, Chen Y, Wang N, Zhang Z (2019) Scale-aware trident networks for object detection. In: IEEE/CVF International Conference on Computer Vision (CVPR), pp 6054–6063
50. Dai J, Li Y, He K, Sun J (2016) R-fcn: Object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems (NIPS), pp 379–387
51. Shelhamer E, Long J, Darrell T (2017) Fully convolutional networks for semantic segmentation. IEEE Trans Pattern Anal Mach Intell 39:640–651
52. Vo DM, Lee S-W (2018) Semantic image segmentation using fully convolutional neural networks with multi-scale images and multi-scale dilated convolutions. Multimed Tools Appl 77:18689–18707
53. Zhang H, Dana K, Shi J, Zhang Z, Wang X, Tyagi A et al (2018) Context Encoding for Semantic Segmentation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 7151–7160
54. Islam MA, Rochan M, Bruce NDB, Wang Y (2017) Gated Feedback Refinement Network for Dense Image Labeling. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 4877–4885
55. O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in MICCAI, 2015
56. Lin G, Milan A, Shen C, Reid I (2017) RefineNet: Multi-path Refinement Networks for High-Resolution Semantic Segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 5168–5177
57. Chen L-C, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. In: European Conference on Computer Vision (ECCV), pp 801–818
58. Redmon J, Divvala S, Girshick RB, Farhadi A (2016) You Only Look Once: Unified, Real-Time Object Detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 779–788
59. Redmon J, Farhadi A (2017) YOLO9000: Better, Faster, Stronger. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 6517–6525
60. Liu W, Anguelov D, Erhan D, Szegedy C, Reed SE, Fu C-Y et al (2016) SSD: Single Shot MultiBox Detector. In: European Conference on Computer Vision (ECCV), pp 21–37
61. Chen Y, Li W, Sakaridis C, Dai D, Van Gool L (2018) Domain adaptive faster r-cnn for object detection in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 3339–3348
62. He K, Gkioxari G, Dollár P, Girshick RB (2020) Mask R-CNN. IEEE Trans Pattern Anal Mach Intell 42:386–397
63. Lin T-Y, Dollár P, Girshick RB, He K, Hariharan B, Belongie SJ (2017) Feature Pyramid Networks for Object Detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 936–944
64. Shrivastava A, Gupta A, Girshick RB (2016) Training Region-Based Object Detectors with Online Hard Example Mining. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 761–769
65. Bulò SR, Neuhold G, Kontschieder P (2017) Loss Max-Pooling for Semantic Image Segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 7082–7091
66. Li B, Liu Y, Wang X (2019) Gradient harmonized single-stage detector. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 8577–8584
67. Chollet F (2017) Xception: Deep Learning with Depthwise Separable Convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1800–1807
68. Lea CS, Vidal R, Hager G (2016) Learning convolutional action primitives for fine-grained action recognition. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp 1642–1649
69. He K, Zhang X, Ren S, Sun J (2016) Deep Residual Learning for Image Recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 770–778

Affiliations

Wenhe Chen[1] · Yuan Chai[1] · Miao Qi[1] · Hui Sun[2] · Qi Pu[1] · Jun Kong[2,3] · Caixia Zheng[1]

✉ Jun Kong
kongjun@nenu.edu.cn

✉ Caixia Zheng
zhengcx789@nenu.edu.cn

[1] College of Information Sciences and Technology,
Northeast Normal University, Changchun 130117, China

[2] Changchun Humanities and Sciences College, Changchun 130117,
China

[3] Key Laboratory of Applied Statistics of MOE, Northeast Normal
University, Changchun 130024, China