Report of temporal action proposal

1. Introduction

In our method, the Boundary-Matching Network (BMN) proposed by Lin et al. [1] is used for temporal action proposal generation. In BMN, a Boundary-Matching (BM) mechanism is proposed to evaluate confidence scores of densely distributed proposals, which denote a proposal as a matching pair of starting and ending boundaries and combine all densely distributed BM pairs into the BM confidence map. Based on BM mechanism, the BMN can generate proposals with precise temporal boundaries as well as reliable confidence scores simultaneously, which is an efficient and end-to-end proposal generation method. Because the proposed method shows state-of-the-art performances on several benchmarks, it is also used here for the HACS dataset.

2. The Approach

As shown in Fig 1, BMN model contains three modules: Base Module handles the input feature sequence, and outputs feature sequence shared by the following two modules; Temporal Evaluation Module evaluates starting and ending probabilities of each location in video to generate boundary probability sequences; Proposal Evaluation Module contains the BM layer to transfer feature sequence to BM feature map, and contains a series of 3D and 2D convolutional layers to generate BM confidence map.

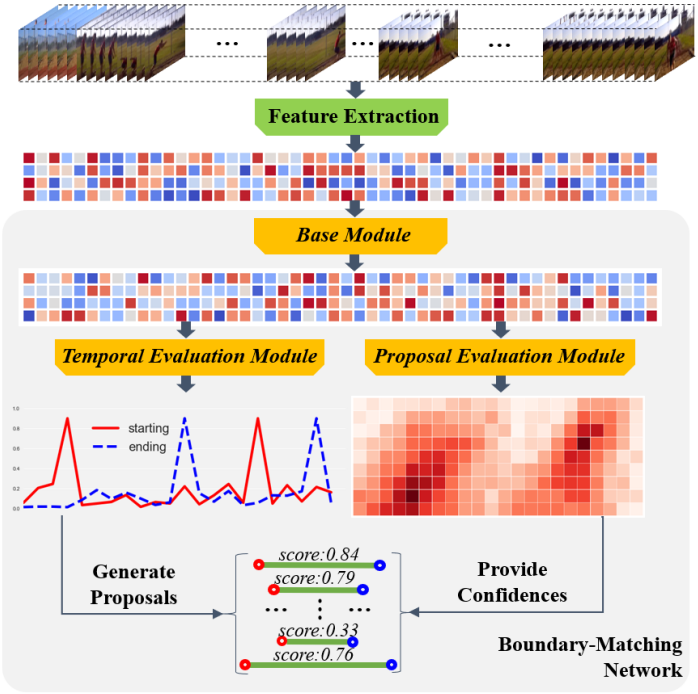

Figure 1. The framework of Boundary-Matching Network

Feature Extraction. The goal of the feature extraction is to extract the feature sequence from video. Temporal Shift Module for Efficient Video Understanding (TSM) by Jilin et al. [2] is used to extract feature sequence on training videos and action classification, which use their RGB and flow streams to train the model based on the temporal shift module, computationally cheap and meanwhile can capture temporal relationships. Nonlocal [3] operation is used to capturing long-range dependencies.

Base Module. The goal of the base module is to handle the input feature sequence, expand the receptive field and serve as backbone of network, to provide a shared feature sequence for TEM and PEM. Since untrimmed videos have uncertain temporal length, we adopt a long observation window with length l_w to truncate the untrimmed feature sequence with length l_f. We denote an observation window as $\omega = \{t_{\omega s}, t_{\omega e}, \Psi_\omega, F_\omega\}$, where $t_{\omega s}$ and $t_{\omega e}$ are the starting and ending time of ω separately, Ψ_ω and F_ω are annotations and feature sequence within the window separately. The window length $l_w = t_{\omega e} - t_{\omega s}$ is set depending on the dataset. The details of base module is shown in Table 1, including two temporal convolutional layers.

Table 1. The detailed architecture of BMN

layer	kernel	stride	dim	act	output size
Base Module					
$conv1d_1$	3	1	256	$relu$	$256\times T$
$conv1d_2$	3	1	128	$relu$	$128\times T$
Temporal Evaluation Module					
$conv1d_3$	3	1	256	$relu$	$256\times T$
$conv1d_4$	3	1	2	$sigmoid$	$2\times T$
Proposal Evaluation Module					
BM layer	N - 32				$128\times 32\times D\times T$
$conv3d_1$	32,1,1	32,0,0	512	$relu$	$512\times 1\times D\times T$
squeeze					$512\times D\times T$
$conv2d_1$	1,1	0,0	128	$relu$	$128\times D\times T$
$conv2d_2$	3,3	1,1	128	$relu$	$128\times D\times T$
$conv2d_3$	1,1	0,0	2	$sigmoid$	$2\times D\times T$

Temporal Evaluation Module (TEM). The goal of TEM is to evaluate the starting and ending probabilities for all temporal locations in untrimmed video. These boundary probability sequences are used for generating proposals during post processing. The details of TEM are shown in Table 1, where conv1d4 layer with two sigmoid activated filters output starting probability sequence $P_{\omega s}$ and ending probability sequence $P_{\omega e}$ for an observation window ω.

Proposal Evaluation Module (PEM). The goal of PEM is to generate Boundary-Matching (BM) confidence map, which contains confidence scores for densely distributed proposals. To achieve this, PEM contains BM layer and a series of 3d and 2d convolutional layers. For the details of BM layer, please refer to the paper [1] proposed by Lin et al.

To train our BMN model on HACS dataset, first extract feature sequence on training videos using their RGB and flow streams similar as on Activitynet-1.3 dataset. Then, using the parameters used in BMN for training Activitynet-1.3, we train our model on HACS dataset. And finally, to improve performance, the validation dataset is added to training dataset for model training. Then, we use the trained model for temporal action proposal on HACS dataset.

3. Experimental Results

Since HACS dataset has large number videos, need to be handle. the Feature Extract Model was trained with the partial Data of HACS in 8 GPU (NVIDIA Tesla P40) in 3 days. The train details like [4], We initialize network weights with pre-trained models from Kinetics-400, the learning rate of spatial networks is 0.001 and decreases to its 0.1 every 10 epoch. The Mini-batch stochastic gradient descent algorithm to learn the network parameters, the batch size is set to 64 and momentum set to 0.9.

As shown in Table 2, we have tried different models for better performance. Because the BMN method is based on the features generated by two stream action classification method, we first try

to use different feature extraction methods for generate the input features to BMN. And we try to use two models (Resnet50 and Resnet101) for action classification on RGB stream to generate the input feature. However, it obtain worse results than with one model (Resnet50) on RGB stream. Therefore, we finally use original two-stream method for BMN feature extraction. Then, for better performance on test dataset, we combine the training and validation datasets for finally model training and better performance. And finally, we obtain the result MAP = 0.2211.

Table 2 Description of Result

Number	Description	MAP on test dataset
1	Use two feature extraction models	0.1292
2	Use original two-stream method	0.2192
3	Training on training and validation datasets	0.2211

Reference

[1] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, Shilei Wen. BMN: Boundary-Matching Network for Temporal Action Proposal Generation. ICCV2019.

[2] Ji Lin, Chuang Gan, Song Han. TSM: Temporal Shift Module for Efficient Video Understanding. ICCV2019.

[3]Xiaolong Wang, Non-local Neural Networks.

[4] Limin Wang, Temporal Segment Networks: Towards Good Practices for Deep Action Recognition