The ActivityNet Large-Scale Activity Recognition Challenge 2018 Summary and Workshop Papers

Bernard Ghanem[1], Juan Carlos Niebles[2,3], Cees Snoek[4], Fabian Caba Heilbron[1], Humam Alwassel[1], Victor Escorcia[1], Ranjay Khrisna[2], Shyamal Buch[2], and Cuong Duc Dao[1]

[1]King Abdullah University of Science and Technology
[2]Stanford University
[3]Universidad del Norte
[4]Universiteit van Amsterdam

Abstract

The 3rd annual installment of the ActivityNet Large-Scale Activity Recognition Challenge, held as a full-day workshop in CVPR 2018, focused on the recognition of daily life, high-level, goal-oriented activities from user-generated videos as those found in internet video portals. The 2018 challenge hosted six diverse tasks which aimed to push the limits of semantic visual understanding of videos as well as bridge visual content with human captions. Three out of the six tasks were based on the ActivityNet dataset, which was introduced in CVPR 2015 and organized hierarchically in a semantic taxonomy. These tasks focused on tracing evidence of activities in time in the form of proposals, class labels, and captions. In this installment of the challenge, we hosted three guest tasks to enrich the understanding of visual information in videos. The guest tasks focused on complementary aspects of the activity recognition problem at large scale and involved three challenging and recently compiled datasets: the Kinetics-600 dataset from Google DeepMind, the AVA dataset from Berkeley and Google, and the Moments in Time dataset from MIT and IBM Research.

1. Introduction

This challenge was the 3rd annual installment of the ActivityNet Large-Scale Activity Recognition Challenge held as a full-day workshop in CVPR 2018. It focused on the recognition of daily life, high-level, goal-oriented activities from user-generated videos as those found in internet video portals. The 2018 challenge hosted six diverse tasks which aimed to push the limits of semantic visual understanding of videos as well as bridge visual content with human captions. Three out of the six tasks were based on the ActivityNet dataset [1], which was introduced in CVPR 2015 and organized hierarchically in a semantic taxonomy. These tasks focused on tracing evidence of activities in time in the form of proposals, class labels, and captions [4]. In this installment of the challenge, we hosted three guest tasks to enrich the understanding of visual information in videos. The guest tasks focused on complementary aspects of the activity recognition problem at large scale and involved three challenging and recently compiled datasets: the Kinetics-600 dataset [3] from AVA dataset [2] from Berkeley and Google, and the Moments in Time dataset [5] from MIT and IBM Research.

How to Cite the Challenge Results? We attach to this document a copy of all the papers submitted to the workshop. Please cite this summary by citing the short summary version on arXiv. In addition, if you want to cite a particular workshop paper, please search online (*e.g.* on arXiv) to see if the paper has been independently published and cite it from that source too; otherwise, just cite the arXiv version of this challenge summary.

2. Main Challenge Tasks

The challenge had three main tasks: **Temporal Action Proposals** (ActivityNet), **Temporal Action Localization** (ActivityNet), and **Dense-Captioning Events in Videos** (ActivityNet Captions). In the following subsections, we describe each task's objective, dataset, and evaluation metric. We finally give the top-3 results on each task.

2.1. Task 1: Temporal Action Proposals

Description and Objective. In many large-scale video analysis scenarios, one is interested in localizing and rec-

ognizing human activities occurring in short temporal intervals within long untrimmed videos. Current approaches for activity detection still struggle to handle large-scale video collections and efficiently addressing this task remains elusive to our visual systems. This is in part due to the computational complexity of current action recognition approaches and the lack of methods that propose fewer intervals in the video, where activity processing can be focused. These set of candidate temporal segments are widely known as *Action Proposals*.

To be applicable at large-scales and in practical scenarios, a useful action proposal method is driven by two competing goals. (i) The proposal method must be computationally efficient in representing, encoding, and scoring a temporal segment. (ii) The proposal method must be discriminative of activities that we are interested in, so as to only retrieve temporal segments that contain visual information indicative of these activity classes. Thus, this task is intended to push the state-of-the-art in action proposal generation algorithms forward.

Dataset. This task is evaluated on the ActivityNet version 1.3 dataset [1]. The dataset consists of more than 648 hours of untrimmed videos from a total of 20K videos. It contains 200 different daily activities such as: *walking the dog*, *long jump*, and *vacuuming floor*. The distribution among training, validation, and testing is roughly 50%, 25%, and 25% of the total videos, respectively.

Evaluation Metric. We use the area under the Average Recall vs. Average Number of Proposals per Video (AR-AN) curve as the evaluation metric for this task. A proposal is a true positive if it has a temporal intersection over union (tIoU) with a ground-truth segment that is greater than or equal to a given threshold (*e.g.* tIoU $>$ 0.5). AR is defined as the mean of all recall values using tIoU between 0.5 and 0.95 (inclusive) with a step size of 0.05. AN is defined as the total number of proposals divided by the number of videos in the testing subset. We consider 100 bins for AN, centered at values between 1 and 100 (inclusive) with a step size of 1, when computing the values on the AR-AN curve.

Top Results. Table 1 shows the top-3 submissions. Each entry in the table links to the corresponding paper submitted by the team. We also append all other papers submitted to the workshop to the end of this summary.

Rank	Organization	AUC
1	Baidu Vis	71.00
2	Shanghai Jiao Tong University	69.30
3	YH Technologies	67.78

Table 1. The top-3 submissions for task 1.

2.2. Task 2: Temporal Action Localization

Description and Objective. Despite the recent advances in large-scale video analysis, temporal action localization remains as one of the most challenging unsolved problems in computer vision. This search problem hinders various real-world applications ranging from consumer video summarization to surveillance, crowd monitoring, and elderly care. Therefore, we are committed to push forward the development of efficient and accurate automated methods that can search and retrieve events and activities in video collections. This task is intended to encourage computer vision researchers to design high performance action localization systems.

Dataset. This task is evaluated on the ActivityNet version 1.3 dataset [1]. The dataset consists of more than 648 hours of untrimmed videos from a total of 20K videos. It contains 200 different daily activities such as: *walking the dog*, *long jump*, and *vacuuming floor*. The distribution among training, validation, and testing is roughly 50%, 25%, and 25% of the total videos, respectively.

Evaluation Metric. We use the Interpolated Average Precision (AP) to evaluate the results on each activity category. The performance on the dataset is measured by the mean AP (mAP) over all the activity categories. To determine if a detection is a true positive, we inspect the tIoU with a ground truth segment, and check whether it is greater or equal to a given threshold (*e.g.* tIoU $>$ 0.5). The official metric used in this task is the average mAP, which is defined as the mean of all mAP values computed with tIoU thresholds between 0.5 and 0.95 (inclusive) with a step size of 0.05.

Top Results. Table 2 shows the top-3 submissions. Each entry in the table links to the corresponding paper submitted by the team. We also append all other papers submitted to the workshop to the end of this summary.

Rank	Organization	Average mAP
1	Shanghai Jiao Tong University	38.53
2	YH Technologies	35.49
3	Baidu Vis	35.27

Table 2. The top-3 submissions for task 2.

2.3. Task 3: Dense-Captioning Events in Videos

Description and Objective. Most natural videos contain numerous events. For example, in a video of a *man playing a piano*, the video might also contain another *man dancing* or a *crowd clapping*. This task aims to tackle the challenges of dense-captioning events, which involves both detecting and describing events in a video.

Dataset. This task is evaluated on the ActivityNet Captions dataset [4]. The dataset connects videos to a series of temporally annotated sentence descriptions. Each sentence

covers a unique segment of the video, describing multiple events that occur. These events may occur over very long or short periods of time and are not limited in any capacity, allowing them to co-occur. On average, each of the 20K videos in ActivityNet Captions contains 3.65 temporally localized sentences, resulting in a total of 100K sentences. The number of sentences per video follows a relatively normal distribution. Furthermore, as the video duration increases, the number of sentences also increases. Each sentence has an average length of 13.48 words, which is also normally distributed.

Evaluation Metric. Inspired by the dense-image captioning metric, we use a similar metric to measure the joint ability of our model to both localize and caption events. This metric computes the average precision (AP) across tIoU thresholds of 0.3, 0.5, and 0.7, when captioning the top-1000 proposals. We measure precision of captions using the traditional evaluation metrics: Bleu, METEOR and CIDEr.

Top Results. Table 3 shows the top-2 submissions. Each entry in the table links to the corresponding paper submitted by the team. We also append all other papers submitted to the workshop to the end of this summary.

Rank	Organization	Average Meteor
1	RUC and CMU	8.53
2	YH Technologies	8.13

Table 3. The top-2 submissions for task 3.

3. Hosted Challenge Tasks

In this installment of the challenge, we hosted three guest tasks to enrich the understanding of visual information in videos. These guest tasks focused on complementary aspects of the activity recognition problem at large scale and involved three challenging and recently compiled datasets: the Kinetics-600 dataset [3] from Google DeepMind, the AVA dataset [2] from Berkeley and Google, and the Moments in Time dataset [5] from MIT and IBM Research.

3.1. Task A: Trimmed Activity Recognition

Description and Objective. This task is intended to evaluate the ability of algorithms to recognize activities in trimmed video sequences. Here, videos contain a single activity, and all the clips have a standard duration.

Dataset. This task is evaluated on the Kinetics-600 dataset [3]. Kinetics is a large-scale, high-quality dataset of YouTube video URLs which include a diverse range of human focused actions. The dataset consists of approximately 500K video clips, and covers 600 human action classes with at least 600 video clips for each action class. Each clip lasts around 10s and is labeled with a single class. All of the clips have been through multiple rounds of human annotation, and each is taken from a unique YouTube video. The actions cover a broad range of classes including human-object interactions such as *playing instruments*, as well as human-human interactions such as *shaking hands* and *hugging*.

Evaluation Metric. We use the top-k accuracy on the testing set as the official metrics for this task. For each video, an algorithm should produce k labels l_j, $j = 1, .., k$. The ground truth label for the video is g. The error of the algorithm for that video would be: $e = \min_j d(l_j, g)$, with $d(x, y) = 0$ if $x = y$ and 1 otherwise. The overall error score for an algorithm is the average error over all videos. We will use $k = 1$ and $k = 5$ and the winner of the challenge is selected based on the average of these two errors.

Top Results. Table 4 shows the top-3 submissions. Each entry in the table links to the corresponding paper submitted by the team. We also append all other papers submitted to the workshop to the end of this summary.

Rank	Organization	Average Error
1	Baidu Vis	10.99
2	YH Technologies	11.69
3	QINIU and SARI	12.20

Table 4. The top-3 submissions for task A.

3.2. Task B: Spatio-temporal Action Localization

Description and Objective. This task is intended to evaluate the ability of algorithms to localize human actions in space and time. Each labeled video segment can contain multiple subjects, each performing potentially multiple actions. The goal is to identify these subjects and actions over continuous video clips. This task is divided into two tracks. Track #1 is strictly computer vision, *i.e.* participants are requested not to use signals derived from audio, metadata, etc. Track #2 lifts this restriction, allowing creative solutions that leverage any input modalities.

Dataset. This task is evaluated on the AVA Dataset version v2.1 [2]. The AVA dataset densely annotates 80 atomic visual actions in 430 15-minute movie clips, where actions are localized in space and time, resulting in 1.58M action labels with multiple labels per human occurring frequently. Clips are drawn from contiguous segments of movies, to open the door for temporal reasoning about activities. The dataset is split into 235 videos for training, 64 videos for validation, and 131 videos for test.

Evaluation Metric. We use the Frame-mAP at spatial IoU ≥ 0.5 as the metric for evaluating algorithms on this task. Since action frequency in AVA follows the natural distribution, the metric is averaged across the top 60 most common action classes in AVA.

Top Results. Tables 5 and 6 show the top-3 submissions for each track. Each entry in the tables links to the corresponding paper submitted by the team. We also append all

other papers submitted to the workshop to the end of this summary.

Rank	Organization	mAP@0.5IoU
1	Tsinghua University	21.08
2	Google DeepMind	21.03
3	YH Technologies	19.60

Table 5. The top-3 submissions for task B (computer vision only track).

Rank	Organization	mAP@0.5IoU
1	Tsinghua University	20.99
2	YH Technologies	19.60
3	UMD	16.76

Table 6. The top-3 submissions for task B (full track).

3.3. Task C: Trimmed Event Recognition

Description and Objective. This task is intended to evaluate the ability of algorithms to classify events in trimmed 3-second videos. Here, videos contain a single activity, and all clips have a standard duration of 3 seconds. This task is divided into two tracks. The first track uses the Moments in Time dataset [5], a new large-scale dataset for video understanding, which has 800K videos in the training set. The second track use the Moments in Time Mini dataset, a subset of Moments in Time with 100k videos provided in the training set.

Dataset. This task is evaluated on the Moments in Time Dataset [5]. Moments in Time Dataset is a large-scale collection of 1M 3-second videos corresponding to spatial-audio-temporal events.

Evaluation Metric. We use the top-k accuracy on the testing set as the official metrics for this task. For each video, an algorithm should produce k labels l_j, $j = 1, .., k$. The ground truth label for the video is g. The error of the algorithm for that video would be: $e = \min_j d(l_j, g)$, with $d(x, y) = 0$ if $x = y$ and 1 otherwise. The overall error score for an algorithm is the average error over all videos. We will use $k = 1$ and $k = 5$ and the winner of the challenge is selected based on the average of these two errors.

Top Results. Tables 7 and 8 show the top-3 submissions for each track. Each entry in the tables links to the corresponding paper submitted by the team. We also append all other papers submitted to the workshop to the end of this summary.

Rank	Organization	Average Accuracy
1	Hikvision	52.91
2	Megvii	51.26
3	Qiniu AtLab	50.06

Table 7. The top-3 submissions for task C (full track).

Rank	Organization	Average Accuracy
1	Sun Yat-Sen University	47.72
2	Beihang University	45.49
3	National Taiwan University	45.10

Table 8. The top-3 submissions for task C (mini track).

References

[1] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[2] C. Gu, C. Sun, S. Vijayanarasimhan, C. Pantofaru, D. A. Ross, G. Toderici, Y. Li, S. Ricco, R. Sukthankar, C. Schmid, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. *CVPR*, 2018.

[3] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.

[4] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles. Dense-captioning events in videos. In *International Conference on Computer Vision (ICCV)*, 2017.

[5] M. Monfort, B. Zhou, S. A. Bargal, T. Yan, A. Andonian, K. Ramakrishnan, L. Brown, Q. Fan, D. Gutfruend, C. Vondrick, et al. Moments in time dataset: one million videos for event understanding.

YH Technologies at ActivityNet Challenge 2018

Ting Yao and Xue Li
YH Technologies Co., Ltd, Beijing, China
{tingyao.ustc, miya.lixue}@gmail.com

Abstract

This notebook paper presents an overview and comparative analysis of our systems designed for the following five tasks in ActivityNet Challenge 2018: temporal action proposals, temporal action localization, dense-captioning events in videos, trimmed action recognition, and spatio-temporal action localization.

***Temporal Action Proposals (TAP)**: To generate temporal action proposals from videos, a three-stage workflow is particularly devised for TAP task: a coarse proposal network (CPN) to generate long action proposals, a temporal convolutional anchor network (CAN) to localize finer proposals, and a proposal reranking network (PRN) to further identify proposals from previous stages. Specifically, CPN explores three complementary actionness curves (namely point-wise, pair-wise, and recurrent curves) that represent actions at different levels to generate coarse proposals, while CAN refines these proposals by a multi-scale cascaded 1D-convolutional anchor network.*

***Temporal Action Localization (TAL)**: For TAL task, we follow the standard "detection by classification" framework, i.e., first generate proposals by our temporal action proposal system and then classify proposals with two-stream P3D classifier.*

***Dense-Captioning Events in Videos (DCEV)**: For DCEV task, we firstly adopt our temporal action proposal system mentioned above to localize temporal proposals of interest in video, and then generate the descriptions for each proposal. Specifically, RNNs encode a given video and its detected attributes into a fixed dimensional vector, and then decode it to the target output sentence. Moreover, we extend the attributes-based CNNs plus RNNs model with policy gradient optimization and retrieval mechanism to further boost video captioning performance.*

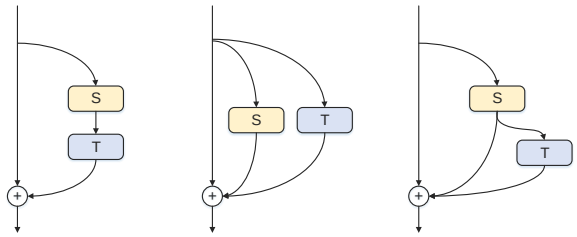
***Trimmed Action Recognition (TAR)**: We investigate and exploit multiple spatio-temporal clues for trimmed action recognition task, i.e., frame, short video clip and motion (optical flow) by leveraging 2D or 3D convolutional neural networks (CNNs). The mechanism of different quantization methods is studied as well. All activities are finally classified by late fusing the predictions from each clue.*

***Spatio-temporal Action Localization (SAL)**: Our system for SAL includes two main components: i.e., Recurrent Tubelet Proposal (RTP) networks and Recurrent Tubelet Recognition (RTR) networks. The RTP initializes action proposals of the start frame through a Region Proposal Network on the feature map and then estimates the movements of proposals in the next frame in a recurrent manner. The action proposals of different frames are linked to form the tubelet proposals. The RTR capitalizes on a multi-channel architecture, where in each channel, a tubelet proposal is fed into a CNN plus LSTM network to recurrently recognize action in the tubelet.*

1. Introduction

Recognizing activities in videos is a challenging task as video is an information-intensive media with complex variations. In particular, an activity may be represented by different clues including frame, short video clip, motion (optical flow) and long video clip. In this work, we aim at investigating these multiple clues to activity classification in trimmed videos, which consist of a diverse range of human focused actions.

Besides detecting actions in manually trimmed short video, researchers tend to develop techniques for detecting actions in untrimmed long videos in the wild. This trend motivates another challenging task—temporal action localization which aims to localize action in untrimmed long videos. We also explore this task in this work. However, most of the natural videos in the real world are untrimmed videos with complex activities and unrelated background/context information, making it hard to directly localize and recognize activities in them. One possible solution is to quickly localize temporal chunks in untrimmed videos containing human activities of interest and then conduct activity recognition over these temporal chunks, which largely simplifies the activity recognition for untrimmed videos. Generating such temporal action chunks in untrimmed videos is known as the task of temporal action proposals, which is also exploited here.



(a) P3D-A (b) P3D-B (c) P3D-C

Figure 1. Three Pseudo-3D blocks.

Furthermore, action detection with accurate spatio-temporal location in videos, i.e., spatio-temporal action localization, is another challenging task in video understanding and we study this task in this work. Compared to temporal action localization which temporally localizes actions, this task is more difficult due to the complex variations and large spatio-temporal search space.

In addition to the above four tasks tailored to activity which is usually the name of action/event in videos, the task of dense-captioning events in videos is explored here which goes beyond activities by describing numerous events within untrimmed videos with multiple natural sentences.

The remaining sections are organized as follows. Section 2 presents all the features which will be adopted in our systems, while Section 3 details the feature quantization strategies. Then the descriptions and empirical evaluations of our systems for five tasks are provided in Section 4-8 respectively, followed by the conclusions in Section 9.

2. Video Representations

We extract the video representations from multiple clues including frame, short clip, motion and long clip.

Frame. To extract frame-level representations from video, we uniformly sample 25 frames for each video/proposal, and then use pre-trained 2D CNNs as frame-level feature extractors. We choose the most popular 2D CNNs in image classification—ResNet [4].

Short Clip. In addition to frame, we take the inspiration from the most popular 3D CNN architecture C3D [20] and devise a novel Pseudo-3D Residual Net (P3D ResNet) architecture [16] to learn spatio-temporal video clip representation in deep networks. Particularly, we develop variants of bottleneck building blocks to combine 2D spatial and 1D temporal convolutions, as shown in Figure 1. The whole P3D ResNet is then constructed by integrating Pseudo-3D blocks into a residual learning framework at different placements. We fix the sample rate as 25 per video.

Motion. To model the change of consecutive frames, we apply another CNNs to optical flow "image," which can extract motion features between consecutive frames. When extracting motion features, we follow the setting of [22], which fed optical flow images, consisting of two-direction optical flow from multiple consecutive frames, into ResNet/P3D ResNet network in each iteration. The sample rate is also set to 25 per video.

Audio. Audio feature is the most global feature (though entire video) in our system. Although audio feature itself can not get very good result for action recognition, but it can be seen as powerful additional feature, since some specific actions are highly related to audio information. Here we utilize MFCC to extract audio features.

3. Feature Quantization

In this section, we describe two quantization methods to generate video-level/clip-level representations.

Average Pooling. Average pooling is the most common method to extract video-level features from consecutive frames, short clips and long clips. For a set of frame-level or clip-level features $F = \{f_1, f_2, ..., f_N\}$, the video-level representations are produced by simply averaging all the features in the set:

$$R_{pooling} = \frac{1}{N} \sum_{i:f_i \in F} f_i \ , \tag{1}$$

where $R_{pooling}$ denotes the final representations.

Compact Bilinear Pooling. Moreover, we utilize Compact Bilinear Pooling (CBP) [3] to produce highly discriminative clip-level representation by capturing the pairwise correlations and modeling interactions between spatial locations within this clip. In particular, given a clip-level feature $F_t \in \mathbb{R}^{W \times H \times D}$ (W, H and D are the width, height and channel numbers), Compact Bilinear Pooling is performed by kernelized feature comparison, which is defined as

$$R_{CBP} = \sum_{j=1}^{S} \sum_{k=1}^{S} \langle \phi(F_{t,j}), \phi(F_{t,k}) \rangle \ , \tag{2}$$

where $S = W \times H$ is the size of the feature map, $F_{t,j}$ is the region-level feature of j-th spatial location in F_t, $\phi(\cdot)$ is a low dimensional projection function, and $\langle \cdot \rangle$ is the second order polynomial kernel.

4. Trimmed Action Recognition

4.1. System

Our trimmed action recognition framework is shown in Figure 2 (a). In general, the trimmed action recognition process is composed of three stages, i.e., multi-stream feature extraction, feature quantization and prediction generation. For deep feature extraction, we follow the multi-stream approaches in [6, 13, 14, 15], which represented input video by a hierarchical structure including individual frame, short clip and consecutive frame. In addition to visual features,

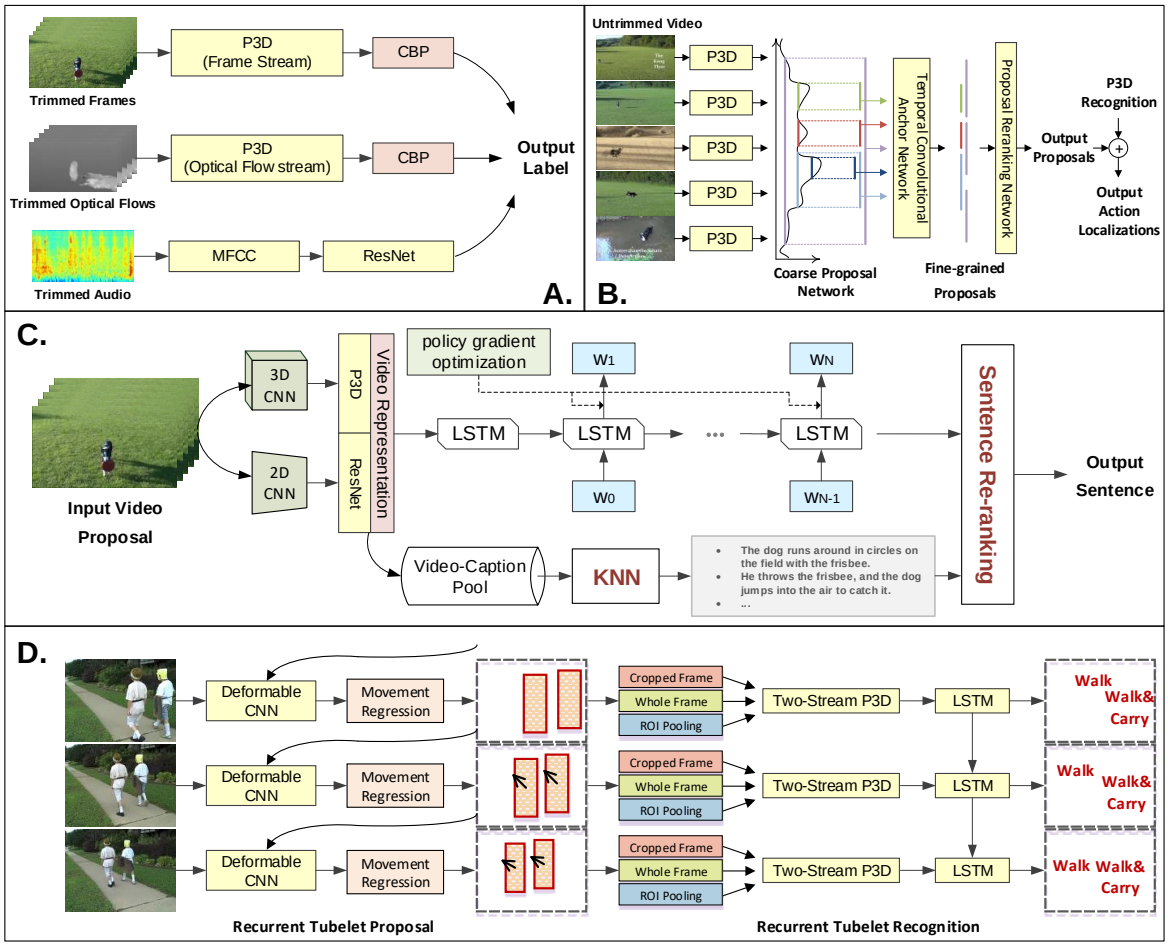


Figure 2. Frameworks of our proposed (a) trimmed action recognition system, (b) temporal action proposals system, (c) dense-captioning events in videos system, and (d) spatio-temporal action localization system.

the most commonly used audio feature MFCC is exploited to further enrich the video representations. After extraction of raw features, different quantization and pooling methods are utilized on different features to produce global representations of each trimmed video. Finally, the predictions from different streams are linearly fused by the weights tuned on validatoin set.

4.2. Experiment Results

Table 1 shows the performances of all the components in our trimmed action recognition system. Overall, the CBP on P3D ResNet (128-frame) achieves the highest top1 accuracy (78.47%) and top5 accuracy (93.99%) of single component. And by additionally apply this model on both frame and optical flow, the two-stream P3D achieves an abvious improvement, which gets top1 accuracy of 80.91% and top5 accuracy of 94.96%. For the final submission, we linearly fuse all the components.

5. Temporal Action Proposals

5.1. System

Figure 2 (b) shows the framework of temporal action proposals, which is mainly composed of three stages:

Coarse Proposal Network (CPN). In this stage, proposal candidates are generated by watershed temporal actionness grouping algorithm (TAG) based on actionness curve. Considering the diversity of action proposals, three actionness measures (namely point-wise, pair-wise and recurrent) that are complementary to each other are leveraged to produce the final actionness curve.

Temporal Convolutional Anchor Network (CAN). Next, we feed long proposals into our temporal convolutional anchor network for finer proposal generation. The temporal convolutional anchor network consists of multiple 1D convolution layers to generate temporal instances for proposal/background binary classification and bounding box regression.

Proposal Reranking Network (PRN). Given the short

Table 1. Comparison of different components in our trimmed action recognition framework on Kinetics validation set for trimmed action recognition task.

Stream	Feature	Layer	Quantization	Top1	Top5
Frame	ResNet	pool5	Ave	74.11%	91.51%
	ResNet	res5c	CBP	74.97%	91.48%
Short Clip	P3D ResNet (16-frame)	pool5	Ave	76.22%	92.92%
	P3D ResNet (128-frame)	pool5	Ave	77.94%	93.75%
	P3D ResNet (128-frame)	res5c	CBP	78.47%	93.99%
Motion	P3D ResNet (16-flow)	pool5	Ave	64.37%	85.76%
	P3D ResNet (128-flow)	pool5	Ave	69.87%	89.44%
	P3D ResNet (128-flow)	res5c	CBP	71.07%	90.00%
Audio	ResNet	pool5	Ave	21.91%	38.49%
Two-stream P3D	P3D ResNet (128-frame&flow)	res5c	CBP	80.91%	94.96%
Fusion all				83.75%	95.95%

Table 2. Area Under the average recall vs. average number of proposals per video Curve (AUC) of frame/flow input for P3D [16] network on ActivityNet validation set for temporal action proposals task.

Stream	CPN	CAN	PRN	AUC
Frame	√			60.27%
	√	√		63.20%
	√	√	√	64.21%
Optical Flow	√			59.83%
	√	√		63.43%
	√	√	√	64.02%
Fusion all				**67.36%**

proposals from the coarse stage and fine-grained proposals from the temporal convolutional anchor network, a reranking network is utilized for proposal refinement. To take video temporal structures into account, we extend the current part of proposal with its' start and end part. The duration of start and end parts are half of the current part. The proposal is then represented by concatenating features of each part to leverage the context information. In our experiments, the top 100 proposals are finally outputted.

5.2. Experiment Results

Table 2 shows the action proposal AUC performances of frame/optical flow input to P3D [16] with different stages in our system. The two stream P3D architecture is pre-trained on Kinetics [5] dataset. For all the single stream runs with different stages, the setting based on all three stages combination achieves the highest AUC. For the final submission, we combine all the proposals from the two streams and then select the top 100 proposals based on their weighted ranking probabilities. The linear fusion weights are tuned on validation set.

Table 3. Performance comparison of different methods on ActivityNet validation set for temporal action localization task. Results are evaluated by mAP with different IoU thresholds and average mAP of IoU threshold from 0.5 to 0.95 with step 0.05.

mAP	0.5	0.75	0.95	Avg mAP
Shou et al. [19]	43.83	25.88	0.21	22.77
Xiong et al. [23]	39.12	23.48	5.49	23.98
Lin et al. [8]	48.99	32.91	7.87	32.26
Ours	51.40	33.61	8.13	34.22

6. Temporal Action Localization

6.1. System

Without loss of generality, we follow the standard "detection by classification" framework, i.e., first generate proposals by temporal action proposals system and then classify proposals. The action classifier is trained with the above trimmed action recognition system (i.e., two-stream P3D) over the 200 categories on ActivityNet dataset [1].

6.2. Experiment Results

Table 3 shows the action localization mAP performance of our approach and baselines on validation set. Our approach consistently outperforms other state-of-the-art approaches in different IoU threshold and achieves 34.22% average mAP on validation set.

7. Dense-Captioning Events in Videos

7.1. System

The main goal of dense-captioning events in videos is jointly localizing temporal proposals of interest in videos and then generating the descriptions for each proposal/video clip. Hence we firstly leverage the temporal action proposal system described above in Section 5 to localize temporal proposals of events in videos (2 proposals for each video). Then, given each temporal proposal (i.e., video seg-

Table 4. Performance of our dense-captioning events in videos system on ActivityNet captions validation set, where B@N, M, R and C are short for BLEU@N, METEOR, ROUGE-L and CIDEr-D scores. All values are reported as percentage (%).

Model	B@1	B@2	B@3	B@4	M	R	C
LSTM-A_3	13.78	7.12	3.53	1.72	7.61	13.30	27.07
LSTM-A_3 + policy gradient	11.65	6.05	3.02	1.34	8.28	12.63	14.62
LSTM-A_3 + policy gradient + retrieval	11.91	6.13	3.04	1.35	8.30	12.65	15.61

ment describing one event), our dense-captioning system runs two different video captioning modules in parallel—the generative module for generating caption via the LSTM-based sequence learning model, and the retrieval module which can directly copy sentences from other visually similar video segments through KNN. Specifically, the generative module with LSTM is inspired from the recent successes of probabilistic sequence models leveraged in vision and language tasks (e.g., image captioning [21, 25], video captioning [9, 10, 12], video generation from captions [11] and dense video captioning [7, 24]). We mainly utilize the third design LSTM-A_3 in [26] which firstly encodes attribute representations into LSTM and then transforms video representations into LSTM at the second time step is adopted as the basic architecture. Note that we employ the policy gradient optimization method with reinforcement learning [18] to boost the video captioning performances specific to METEOR metric. For the retrieval module, we utilize KNN to find the visually similar video segments based on the extracted video representations. The captions associated with the top similar video segments are regarded as sentence candidates in retrieval module. In the experiment, we mainly choose the top 300 nearest neighbors for generating sentence candidates. Finally, a sentence re-ranking module is exploited to rank and select the final most consensus caption from the two parallel video captioning modules by considering the lexical similarity among all the sentence candidates. The overall architecture of our dense-captioning system is shown in Figure 2 (c).

7.2. Experiment Results

Table 4 shows the performances of our proposed dense-captioning events in videos system. Here we compare three variants derived from our proposed model. In particular, by additionally incorporating the policy gradient optimization scheme into the basic LSTM-A_3 architecture, we can clearly observe the performance boost in METEOR. Moreover, our dense-captioning model (LSTM-A_3 + policy gradient + retrieval) is further improved by injecting the sentence candidates from retrieval module in METEOR.

8. Spatio-temporal Action Localization

8.1. System

Figure 2 (d) shows the framework of spatio-temporal action localization, which includes two main components:

Table 5. Comparison of different components in our RTR on AVA validation set for spatio-temporal action localization task.

Stream	Feature	mAP@IoU=0.5
Frame	ResNet	13.68
Short Clip	P3D ResNet (16-frame)	19.12
Short Clip	P3D ResNet (128-frame)	19.40
Flow	P3D ResNet (16-frame)	15.20
Fusion	-	**22.20**

Recurrent Tubelet Proposal (RTP) networks. The Recurrent Tubelet Proposal networks produces action proposals in a recurrent manner. Specifically, it initializes action proposals of the start frame through a Region Proposal Network (RPN) [17] on the feature map. Then the movement of each proposal in the next frame is estimated from three inputs: feature maps of both current and next frames, and the proposal in current frame. Simultaneously, a sibling proposal classifier is utilized to infer the actionness of the proposal. To form the tubelet proposals, action proposals in two consecutive frames are linked by taking both their actionness and overlap ratio into account, followed by the temporal trimming on tubelet.

Recurrent Tubelet Recognition (RTR) networks. The Recurrent Tubelet Recognition networks capitalizes on a multi-channel architecture for tubelet proposal recognition. For each proposal, we extract three different semantic-level features, i.e., the features on proposal-cropped image, the features with RoI pooling on the proposal, and the features on whole frame. These features implicitly encode the spatial context and scene information, which could enhance the recognition capability on specific categories. After that, each of them is fed into a LSTM to model the temporal dynamics for tubelet recognition.

8.2. Experiment Results

We construct our RTP based on [2], which is mainly trained with the single RGB frames. For RTR, we extract the region representations with RoI pooling from multiple clues including frame, clip and motion. Table 5 shows the performances of all the components in our RTR. Overall, the P3D ResNet trained on clips (128 frames) achieves the highest frame-mAP (19.40%) of single component. For the final submission, all the components are linearly fused using the weights tuned on validation set. The final mAP on validation set is 22.20%.

9. Conclusion

In ActivityNet Challenge 2018, we mainly focused on multiple visual features, different strategies of feature quantization and video captioning from different dimensions. Our future works include more in-depth studies of how fusion weights of different clues could be determined to boost the action recognition/temporal action proposals/temporal action localization/spatio-temporal action localization performance and how to generate open-vocabulary sentences for events in videos.

References

[1] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[2] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *CVPR*, 2017.

[3] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell. Compact bilinear pooling. In *CVPR*, 2016.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[5] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[6] Q. Li, Z. Qiu, T. Yao, T. Mei, Y. Rui, and J. Luo. Action recognition by learning deep multi-granular spatio-temporal video representation. In *ICMR*, 2016.

[7] Y. Li, T. Yao, Y. Pan, H. Chao, and T. Mei. Jointly localizing and describing events for dense video captioning. In *CVPR*, 2018.

[8] T. Lin, X. Zhao, and Z. Shou. Temporal Convolution Based Action Proposal: Submission to ActivityNet 2017. *CoRR*, 2017.

[9] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, 2016.

[10] Y. Pan, Z. Qiu, T. Yao, H. Li, and T. Mei. Seeing bot. In *SIGIR*, 2017.

[11] Y. Pan, Z. Qiu, T. Yao, H. Li, and T. Mei. To create what you tell: Generating videos from captions. In *MM Brave New Idea*, 2017.

[12] Y. Pan, T. Yao, H. Li, and T. Mei. Video captioning with transferred semantic attributes. In *CVPR*, 2017.

[13] Z. Qiu, D. Li, C. Gan, T. Yao, T. Mei, and Y. Rui. Msr asia msm at activitynet challenge 2016. In *CVPR workshop*, 2016.

[14] Z. Qiu, Q. Li, T. Yao, T. Mei, and Y. Rui. Msr asia msm at thumos challenge 2015. In *THUMOS'15 Action Recognition Challenge*, 2015.

[15] Z. Qiu, T. Yao, and T. Mei. Deep quantization: Encoding convolutional activations with deep generative model. In *CVPR*, 2017.

[16] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 2017.

[17] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[18] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. *arXiv preprint arXiv:1612.00563*, 2016.

[19] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang. CDC: Convolutional-De-Convolutional Network for Precise Temporal Action Localization in Untrimmed Videos. In *CVPR*, 2017.

[20] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[21] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.

[22] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. Towards good practices for very deep two-stream convnets. *arXiv preprint arXiv:1507.02159*, 2015.

[23] Y. Xiong, Y. Zhao, L. Wang, D. Lin, and X. Tang. A Pursuit of Temporal Accuracy in General Activity Detection. *CoRR*, 2017.

[24] T. Yao, Y. Li, Z. Qiu, F. Long, Y. Pan, D. Li, and T. Mei. Msr asia msm at activitynet challenge 2017: Trimmed action recognition, temporal action proposals and dense-captioning events in videos. In *CVPR ActivityNet Challenge Workshop*, 2017.

[25] T. Yao, Y. Pan, Y. Li, and T. Mei. Incorporating copying mechanism in image captioning for learning novel objects. In *CVPR*, 2017.

[26] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei. Boosting image captioning with attributes. In *ICCV*, 2017.

Boundary Sensitive Network: Submission to ActivityNet Challenge 2018

Tianwei Lin, Haisheng Su, Xu Zhao*

Department of Automation,
Shanghai Jiao Tong University
{wzmsltw,suhaisheng,zhaoxu}@sjtu.edu.cn

Abstract. In this technical paper, we describe our approach used in the submission to the **temporal action proposal generation (task 1)** and **temporal action localization (detection) (task 2)** of ActivityNet Challenge 2018. Since we believe that the main bottleneck for temporal action localization is the quality of action proposals, we mainly focus on the temporal action proposal generation task and adopt a novel proposal generation method we proposed recently, called Boundary-Sensitive Network (BSN) [1]. To generate high quality proposals, BSN first locates temporal boundaries with high probabilities, then directly combines these boundaries as proposals. Finally, with Boundary-Sensitive Proposal feature, BSN retrieves proposals by evaluating the confidence of whether a proposal contains an action within its region. BSN achieves the state-of-the-art performances on both temporal action proposal generation task and temporal action localization task. The full version of our paper can be found in [1].

Keywords: Temporal action proposal generation · Temporal action detection · Temporal convolution · Untrimmed video

1 Introduction

Nowadays, with fast development of digital cameras and Internet, the number of videos is continuously booming, making automatic video content analysis methods widely required. One major branch of video analysis is action recognition, which aims to classify manually trimmed video clips containing only one action instance. However, videos in real scenarios are usually long, untrimmed and contain multiple action instances along with irrelevant contents. This problem requires algorithms for another challenging task: temporal action detection, which aims to detect action instances in untrimmed video including both temporal boundaries and action classes. It can be applied in many areas such as video recommendation and smart surveillance.

Similar with object detection in spatial domain, temporal action detection task can be divided into two stages: proposal and classification. Proposal generation stage aims to generate temporal video regions which may contain action instances, and classification stage aims to classify classes of candidate proposals. Although classification methods have reached convincing performance, the detection precision is still low in

* Corresponding author.

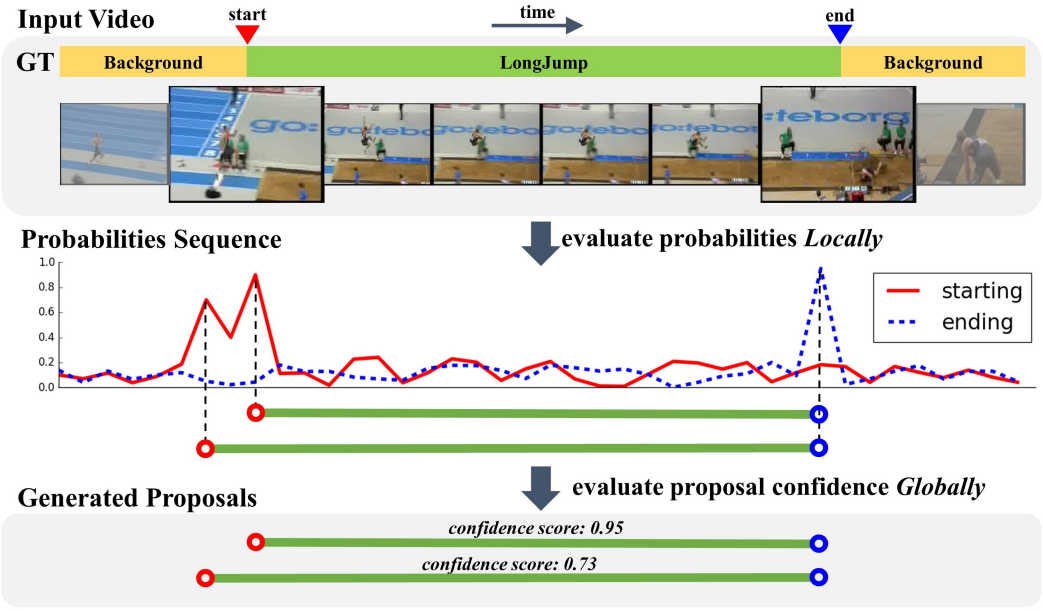


Fig. 1: Overview of our approach. Given an untrimmed video, (1) we evaluate boundaries and actionness probabilities of each temporal location and generate proposals based on boundary probabilities, and (2) we evaluate the confidence scores of proposals with proposal-level feature to get retrieved proposals.

many benchmarks [2, 3]. Thus recently temporal action proposal generation has received much attention [4–7], aiming to improve the detection performance by improving the quality of proposals. High quality proposals should come up with two key properties: (1) proposals can cover truth action regions with both high recall and high temporal overlap, (2) proposals are retrieved so that high recall and high overlap can be achieved using fewer proposals to reduce the computation cost of succeeding steps.

To achieve high proposal quality, a proposal generation method should generate proposals with flexible temporal durations and precise temporal boundaries, then retrieve proposals with reliable confidence scores, which indicate the probability of a proposal containing an action instance. Most recently proposal generation methods [4–6, 8] generate proposals via sliding temporal windows of multiple durations in video with regular interval, then train a model to evaluate the confidence scores of generated proposals for proposals retrieving, while there is also method [7] making external boundaries regression. However, proposals generated with pre-defined durations and intervals may have some major drawbacks: (1) usually not temporally precise; (2) not flexible enough to cover variable temporal durations of ground truth action instances, especially when the range of temporal durations is large.

To address these issues and generate high quality proposals, we propose the Boundary-Sensitive Network (BSN), which adopts "*local to global*" fashion to locally combine high probability boundaries as proposals and globally retrieve candidate proposals using proposal-level feature as shown in Fig 1. In detail, BSN generates proposals in three steps. **First**, BSN evaluates the probabilities of each temporal location in video whether it is inside or outside, at or not at the boundaries of ground truth action instances, to generate starting, ending and actionness probabilities sequences as local information. **Second**, BSN generates proposals via directly combining temporal locations with high

starting and ending probabilities separately. Using this bottom-up fashion, BSN can generate proposals with flexible durations and precise boundaries. **Finally**, using features composed by actionness scores within and around proposal, BSN retrieves proposals by evaluating the confidence of whether a proposal contains an action. These proposal-level features offer global information for better evaluation.

In summary, the main contributions of our work are three-folds:

(1) We introduce a new architecture (BSN) based on "*local to global*" fashion to generate high quality temporal action proposals, which *locally* locates high boundary probability locations to achieve precise proposal boundaries and *globally* evaluates proposal-level feature to achieve reliable proposal confidence scores for retrieving.

(2) Extensive experiments demonstrate that our method achieves significantly better proposal quality than other state-of-the-art proposal generation methods, and can generate proposals in unseen action classes with comparative quality.

(3) Integrating our method with existing action classifier into detection framework leads to significantly improved performance on temporal action detection task.

2 Related work

Action recognition. Action recognition is an important branch of video related research areas and has been extensively studied. Earlier methods such as improved Dense Trajectory (iDT) [9, 10] mainly adopt hand-crafted features such as HOF, HOG and MBH. In recent years, convolutional networks are widely adopted in many works [11–14] and have achieved great performance. Typically, two-stream network [11, 12, 14] learns appearance and motion features based on RGB frame and optical flow field separately. C3D network [13] adopts 3D convolutional layers to directly capture both appearance and motion features from raw frames volume. Action recognition models can be used for extracting frame or snippet level visual features in long and untrimmed videos.

Object detection and proposals. Recent years, the performance of object detection has been significantly improved with deep learning methods. R-CNN [15] and its variations [16, 17] construct an important branch of object detection methods, which adopt "detection by classifying proposals" framework. For proposal generation stage, besides sliding windows [18], earlier works also attempt to generate proposals by exploiting low-level cues such as HOG and Canny edge [19, 20]. Recently some methods [17, 21, 22] adopt deep learning model to generate proposals with faster speed and stronger modelling capacity. In this work, we combine the properties of these methods via evaluating boundaries and actionness probabilities of each location using neural network and adopting "*local to global*" fashion to generate proposals with high recall and accuracy.

Boundary probabilities are also adopted in LocNet [23] for revising the horizontal and vertical boundaries of existing proposals. Our method differs in (1) BSN aims to generate while LocNet aims to revise proposals and (2) boundary probabilities are calculated repeatedly for all boxes in LocNet but only once for a video in BSN.

Temporal action detection and proposals. Temporal action detection task aims to detect action instances in untrimmed videos including temporal boundaries and action classes, and can be divided into proposal and classification stages. Most detection methods [8, 24, 25] take these two stages separately, while there is also method [26, 27] tak-

ing these two stages jointly. For proposal generation, earlier works [28–30] directly use sliding windows as proposals. Recently some methods [4–8] generate proposals with pre-defined temporal durations and intervals, and use multiple methods to evaluate the confidence score of proposals, such as dictionary learning [5] and recurrent neural network [6]. TAG method [25] adopts watershed algorithm to generate proposals with flexible boundaries and durations in *local* fashion, but without *global* proposal-level confidence evaluation for retrieving. In our work, BSN can generate proposals with flexible boundaries meanwhile reliable confidence scores for retrieving.

Recently temporal action detection method [31] detects action instances based on class-wise start, middle and end probabilities of each location. Our method is superior than [31] in two aspects: (1) BSN evaluates probabilities score using temporal convolution to better capture temporal information and (2) "*local to global*" fashion adopted in BSN brings more precise boundaries and better retrieving quality.

3 Our Approach

3.1 Problem Definition

An untrimmed video sequence can be denoted as $X = \{x_n\}_{n=1}^{l_v}$ with l_v frames, where x_n is the n-th frame in X. Annotation of video X is composed by a set of action instances $\Psi_g = \{\varphi_n = (t_{s,n}, t_{e,n})\}_{n=1}^{N_g}$, where N_g is the number of truth action instances in video X, and $t_{s,n}$, $t_{e,n}$ are starting and ending time of action instance φ_n separately. Unlike detection task, classes of action instances are not considered in temporal action proposal generation. Annotation set Ψ_g is used during training. During prediction, generated proposals set Ψ_p should cover Ψ_g with high recall and high temporal overlap.

3.2 Video Features Encoding

To generate proposals of input video, first we need to extract feature to encode visual content of video. In our framework, we adopt two-stream network [12] as visual encoder, since this architecture has shown great performance in action recognition task [32] and has been widely adopted in temporal action detection and proposal generation tasks [25, 26, 33]. Two-stream network contains two branches: spatial network operates on single RGB frame to capture appearance feature, and temporal network operates on stacked optical flow field to capture motion information.

To extract two-stream features, as shown in Fig 2(a), first we compose a snippets sequence $S = \{s_n\}_{n=1}^{l_s}$ from video X, where l_s is the length of snippets sequence. A snippet $s_n = (x_{t_n}, o_{t_n})$ includes two parts: x_{t_n} is the t_n-th RGB frame in X and o_{t_n} is stacked optical flow field derived around center frame x_{t_n}. To reduce the computation cost, we extract snippets with a regular frame interval σ, therefore $l_s = l_v/\sigma$. Given a snippet s_n, we concatenate output scores in top layer of both spatial and temporal networks to form the encoded feature vector $f_{t_n} = (f_{S,t_n}, f_{T,t_n})$, where f_{S,t_n}, f_{T,t_n} are output scores from spatial and temporal networks separately. Thus given a snippets sequence S with length l_s, we can extract a feature sequence $F = \{f_{t_n}\}_{n=1}^{l_s}$. These two-stream feature sequences are used as the input of BSN.

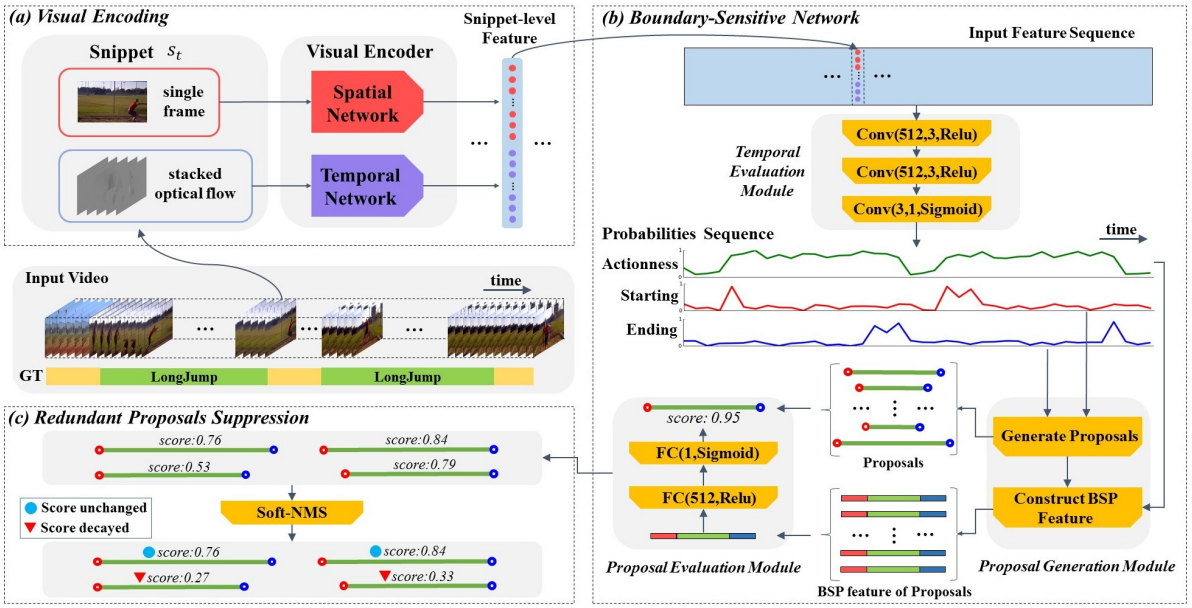


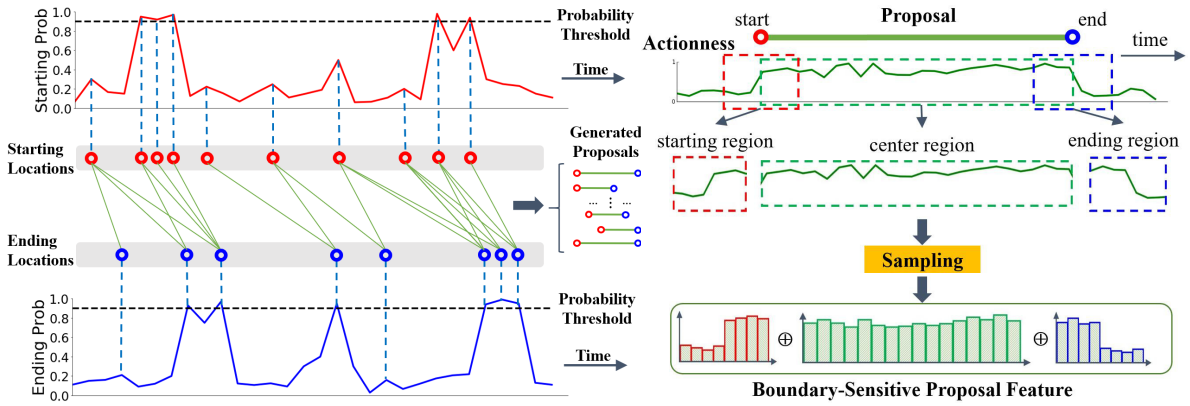
Fig. 2: The framework of our approach. (a) Two-stream network is used for encoding visual features in snippet-level. (b) The architecture of Boundary-Sensitive Network: *temporal evaluation module* handles the input feature sequence, and evaluates starting, ending and actionness probabilities of each temporal location; *proposal generation module* generates proposals with high starting and ending probabilities, and construct Boundary-Sensitive Proposal (BSP) feature for each proposal; *proposal evaluation module* evaluates confidence score of each proposal using BSP feature. (c) Finally, we use Soft-NMS algorithm to suppress redundant proposals by decaying their scores.

3.3 Boundary-Sensitive Network

To achieve high proposal quality with both precise temporal boundaries and reliable confidence scores, we adopt "*local to global*" fashion to generate proposals. In BSN, we first generate candidate boundary locations, then combine these locations as proposals and evaluate confidence score of each proposal with proposal-level feature.

Network architecture. The architecture of BSN is presented in Fig 2(b), which contains three modules: temporal evaluation, proposal generation and proposal evaluation. *Temporal evaluation module* is a three layers temporal convolutional neural network, which takes the two-stream feature sequences as input, and evaluates probabilities of each temporal location in video whether it is inside or outside, at or not at boundaries of ground truth action instances, to generate sequences of starting, ending and actionness probabilities respectively. *Proposal generation module* first combines the temporal locations with separately high starting and ending probabilities as candidate proposals, then constructs Boundary-Sensitive Proposal (BSP) feature for each candidate proposal based on actionness probabilities sequence. Finally, *proposal evaluation module*, a multilayer perceptron model with one hidden layer, evaluates the confidence score of each candidate proposal based on BSP feature. Confidence score and boundary probabilities of each proposal are fused as the final confidence score for retrieving.

Temporal evaluation module. The goal of temporal evaluation module is to evaluate starting, ending and actionness probabilities of each temporal location, where three binary classifiers are needed. In this module, we adopt temporal convolutional layers upon



(a) Generate proposals (b) Construct BSP feature

Fig. 3: Details of proposal generation module. (a) Generate proposals. First, to generate candidate boundary locations, we choose temporal locations with high boundary probability or being a probability peak. Then, we combine candidate starting and ending locations as proposals when their duration satisfying condition. (b) Construct BSP feature. Given a proposal and actionness probabilities sequence, we can sample actionness sequence in starting, center and ending regions of proposal to construct BSP feature.

feature sequence, with good modelling capacity to capture local semantic information such as boundaries and actionness probabilities.

A temporal convolutional layer can be simply denoted as $Conv(c_f, c_k, Act)$, where c_f, c_k and Act are filter numbers, kernel size and activation function of temporal convolutional layer separately. As shown in Fig 2(b), the temporal evaluation module can be defined as $Conv(512, 3, Relu) \rightarrow Conv(512, 3, Relu) \rightarrow Conv(3, 1, Sigmoid)$, where the three layers have same stride size 1. Three filters with sigmoid activation in the last layer are used as classifiers to generate starting, ending and actionness probabilities separately. For convenience of computation, we divide feature sequence into non-overlapped windows as the input of temporal evaluation module. Given a feature sequence F, temporal evaluation module can generate three probability sequences $P_S = \{p^s_{t_n}\}_{n=1}^{l_s}$, $P_E = \{p^e_{t_n}\}_{n=1}^{l_s}$ and $P_A = \{p^a_{t_n}\}_{n=1}^{l_s}$, where $p^s_{t_n}$, $p^e_{t_n}$ and $p^a_{t_n}$ are respectively starting, ending and actionness probabilities in time t_n.

Proposal generation module. The goal of proposal generation module is to generate candidate proposals and construct corresponding proposal-level feature. We achieve this goal in two steps. First we locate temporal locations with high boundary probabilities, and combine these locations to form proposals. Then for each proposal, we construct Boundary-Sensitive Proposal (BSP) feature.

As shown in Fig 3(a), to locate where an action likely to start, for starting probabilities sequence P_S, we record all temporal location t_n where $p^s_{t_n}$ (1) has high score: $p^s_{t_n} > 0.9$ or (2) is a probability peak: $p^s_{t_n} > p^s_{t_{n-1}}$ and $p^s_{t_n} > p^s_{t_{n+1}}$. These locations are grouped into candidate starting locations set $B_S = \{t_{s,i}\}_{i=1}^{N_S}$, where N_S is the number of candidate starting locations. Using same rules, we can generate candidate ending locations set B_E from ending probabilities sequence P_E. Then, we generate temporal regions via combing each starting location t_s from B_S and each ending location t_e from B_S. Any temporal region $[t_s, t_e]$ satisfying $d = t_e - t_s \in [d_{min}, d_{max}]$ is denoted as a candidate proposal φ, where d_{min} and d_{max} are minimum and maximum durations

of ground truth action instances in dataset. Thus we can get candidate proposals set $\Psi_p = \{\varphi_i\}_{i=1}^{N_p}$, where N_p is the number of proposals.

To construct proposal-level feature as shown in Fig 3(b), for a candidate proposal φ, we denote its center region as $r_C = [t_s, t_e]$ and its starting and ending region as $r_S = [t_s - d/5, t_s + d/5]$ and $r_E = [t_e - d/5, t_e + d/5]$ separately. Then, we sample the actionness sequence P_A within r_c as f_c^A by linear interpolation with 16 points. In starting and ending regions, we also sample actionness sequence with 8 linear interpolation points and get f_s^A and f_e^A separately. Concatenating these vectors, we can get Boundary-Sensitive Proposal (BSP) feature $f_{BSP} = (f_s^A, f_c^A, f_e^A)$ of proposal φ. BSP feature is highly compact and contains rich semantic information about corresponding proposal. Then we can represent a proposal as $\varphi = (t_s, t_e, f_{BSP})$.

Proposal evaluation module. The goal of proposal evaluation module is to evaluate the confidence score of each proposal whether it contains an action instance within its duration using BSP feature. We adopt a simple multilayer perceptron model with one hidden layer as shown in Fig 2(b). Hidden layer with 512 units handles the input of BSP feature f_{BSP} with Relu activation. The output layer outputs confidence score p_{conf} with sigmoid activation, which estimates the overlap extent between candidate proposal and ground truth action instances. Thus, a generated proposal can be denoted as $\varphi = (t_s, t_e, p_{conf}, p_{t_s}^s, p_{t_e}^e)$, where $p_{t_s}^s$ and $p_{t_e}^e$ are starting and ending probabilities in t_s and t_e separately. These scores are fused to generate final score during prediction.

3.4 Training of BSN

In BSN, temporal evaluation module is trained to learn local boundary and actionness probabilities from video features simultaneously. Then based on probabilities sequence generated by trained temporal evaluation module, we can generate proposals and corresponding BSP features and train the proposal evaluation module to learn the confidence score of proposals. The training details are introduced in this section.

Temporal evaluation module. Given a video X, we compose a snippets sequence S with length l_s and extract feature sequence F from it. Then we slide windows with length $l_w = 100$ in feature sequence without overlap. A window is denoted as $\omega = \{F_\omega, \Psi_\omega\}$, where F_ω and Ψ_ω are feature sequence and annotations within the window separately. For ground truth action instance $\varphi_g = (t_s, t_e)$ in Ψ_ω, we denote its region as action region r_g^a and its starting and ending region as $r_g^s = [t_s - d_g/10, t_s + d_g/10]$ and $r_g^e = [t_e - d_g/10, t_e + d_g/10]$ separately, where $d_g = t_e - t_s$.

Taking F_ω as input, temporal evaluation module generates probabilities sequence $P_{S,\omega}$, $P_{E,\omega}$ and $P_{A,\omega}$ with same length l_w. For each temporal location t_n within F_ω, we denote its region as $r_{t_n} = [t_n - d_s/2, t_n + d_s/2]$ and get corresponding probability scores $p_{t_n}^s$, $p_{t_n}^e$ and $p_{t_n}^a$ from $P_{S,\omega}$, $P_{E,\omega}$ and $P_{A,\omega}$ separately, where $d_s = t_n - t_{n-1}$ is temporal interval between two snippets. Then for each r_{t_n}, we calculate its IoP ratio with r_g^a, r_g^s and r_g^e of all φ_g in Ψ_ω separately, where IoP is defined as the overlap ratio with groundtruth proportional to the duration of this proposal. Thus we can represent information of t_n as $\phi_n = (p_{t_n}^a, p_{t_n}^s, p_{t_n}^e, g_{t_n}^a, g_{t_n}^s, g_{t_n}^e)$, where $g_{t_n}^a$, $g_{t_n}^s$, $g_{t_n}^e$ are maximum matching overlap IoP of action, starting and ending regions separately.

Given a window of matching information as $\Phi_\omega = \{\phi_n\}_{n=1}^{l_s}$, we can define training objective of this module as a three-task loss function. The overall loss function consists of actionness loss, starting loss and ending loss:

$$L_{TEM} = \lambda \cdot L_{bl}^{action} + L_{bl}^{start} + L_{bl}^{end}, \tag{1}$$

where λ is the weight term and is set to 2 in BSN. We adopt the sum of binary logistic regression loss function L_{bl} for all three tasks, which can be denoted as:

$$L_{bl} = \frac{1}{l_w} \sum_{i=1}^{l_w} \left(\alpha^+ \cdot b_i \cdot log(p_i) + \alpha^- \cdot (1 - b_i) \cdot log(1 - p_i) \right), \tag{2}$$

where $b_i = sign(g_i - \theta_{IoP})$ is a two-values function for converting matching score g_i to $\{0, 1\}$ based on threshold θ_{IoP}, which is set to 0.5 in BSN. Let $l^+ = \sum g_i$ and $l^- = l_w - l^+$, we can set $\alpha^+ = \frac{l_w}{l^-}$ and $\alpha^- = \frac{l_w}{l^+}$, which are used for balancing the effect of positive and negative samples during training.

Proposal evaluation module. Using probabilities sequences generated by trained temporal evaluation module, we can generate proposals using proposal generation module: $\Psi_p = \{\varphi_n = (t_s, t_e, f_{BSP})\}_{n=1}^{N_p}$. Taking f_{BSP} as input, for a proposal φ, confidence score p_{conf} is generated by proposal evaluation module. Then we calculate its Intersection-over-Union (IoU) with all φ_g in Ψ_g, and denote the maximum overlap score as g_{iou}. Thus we can represent proposals set as $\Psi_p = \{\varphi_n = \{t_s, t_e, p_{conf}, g_{iou}\}\}_{n=1}^{N_p}$. We split Ψ_p into two parts based on g_{iou}: Ψ_p^{pos} for $g_{iou} > 0.7$ and Ψ_p^{neg} for $g_{iou} < 0.3$. For data balancing, we take all proposals in Ψ_p^{pos} and randomly sample the proposals in Ψ_p^{neg} to insure the ratio between two sets be nearly 1:2.

The training objective of this module is a simple regression loss, which is used to train a precise confidence score prediction based on IoU overlap. We can define it as:

$$L_{PEM} = \frac{1}{N_{train}} \sum_{i=1}^{N_{train}} (p_{conf,i} - g_{iou,i})^2, \tag{3}$$

where N_{train} is the number of proposals used for training.

3.5 Prediction and Post-processing

During prediction, we use BSN with same procedures described in training to generate proposals set $\Psi_p = \left\{\varphi_n = (t_s, t_e, p_{conf}, p_{t_s}^s, p_{t_e}^e)\right\}_{n=1}^{N_p}$, where N_p is the number of proposals. To get final proposals set, we need to make score fusion to get final confidence score, then suppress redundant proposals based on these score.

Score fusion for retrieving. To achieve better retrieving performance, for each candidate proposal φ, we fuse its confidence score with its boundary probabilities by multiplication to get the final confidence score p_f:

$$p_f = p_{conf} \cdot p_{t_s}^s \cdot p_{t_e}^e. \tag{4}$$

After score fusion, we can get generated proposals set $\Psi_p = \{\varphi_n = (t_s, t_e, p_f)\}_{n=1}^{N_p}$, where p_f is used for proposals retrieving. In section 4.2, we explore the recall performance with and without confidence score generated by proposal evaluation module.

Redundant proposals suppression. Around a ground truth action instance, we may generate multiple proposals with different temporal overlap. Thus we need to suppress redundant proposals to obtain higher recall with fewer proposals.

Soft-NMS [34] is a recently proposed non-maximum suppression (NMS) algorithm which suppresses redundant results using a score decaying function. First all proposals are sorted by their scores. Then proposal φ_m with maximum score is used for calculating overlap IoU with other proposals, where scores of highly overlapped proposals is decayed. This step is recursively applied to the remaining proposals to generate rescored proposals set. The Gaussian decaying function of Soft-NMS can be denoted as:

$$p'_{f,i} = \begin{cases} p_{f,i}, & iou(\varphi_m, \varphi_i) < \theta \\ p_{f,i} \cdot e^{-\frac{iou(\varphi_m, \varphi_i)^2}{\varepsilon}}, & iou(\varphi_m, \varphi_i) \geq \theta \end{cases} \tag{5}$$

where ε is parameter of Gaussian function and θ is pre-fixed threshold. After suppression, we get the final proposals set $\Psi'_p = \left\{\varphi_n = (t_s, t_e, p'_f)\right\}_{n=1}^{N_p}$.

4 Experiments

In the full version of BSN paper [1], we conduct experiments on both ActivityNet-1.3 and THUMOS-14 datasets including many ablation studys, where BSN achieves great performance on both datasets. In this challenge report, we mainly introduce new improvements and experiments of BSN on ActivityNet Challenge 2018. And for convenience, we denote BSN introduced in [1] as BSN-baseline.

4.1 Dataset and setup

Dataset. ActivityNet-1.3 [2] is a large dataset for general temporal action proposal generation and detection, which contains 19994 videos with 200 action classes annotated and was used in the ActivityNet Challenge 2017 and 2018. ActivityNet-1.3 is divided into training, validation and testing sets by ratio of 2:1:1.

Evaluation metrics. In temporal action proposal generation task, Average Recall (AR) calculated with multiple IoU thresholds is usually used as evaluation metrics. Following conventions, we use IoU thresholds set $[0.5 : 0.05 : 0.95]$. To evaluate the relation between recall and proposals number, we evaluate AR with Average Number of proposals (AN) on both datasets, which is denoted as AR@AN. Area under the AR vs. AN curve (AUC) is also used as metrics, where AN varies from 0 to 100.

In temporal action detection task, mean Average Precision (mAP) is used as evaluation metric, where Average Precision (AP) is calculated on each action class respectively. On ActivityNet-1.3, mAP with IoU thresholds $\{0.5, 0.75, 0.95\}$ and average mAP with IoU thresholds set $[0.5 : 0.05 : 0.95]$ are used.

Implementation details. Here, we mainly introduce the implementation details adopted in BSN-baseline, the improvement details will be introduced later. For visual feature encoding, we use the two-stream network [12] with architecture described in [35], where BN-Inception network [36] is used as temporal network and ResNet network [37] is

Table 1: Comparison between our method with other state-of-the-art proposal generation methods on ActivityNet-1.3 in terms of AR@AN and AUC.

Method	AR@10 (val)	AR@100 (val)	AUC (val)	AUC (test)
Uniform Random	29.02	55.71	44.88	-
Zhao et al. [25]	-	63.52	53.02	-
Dai et al. [41]	-	-	59.58	61.56
Yao et al. [42]	-	-	63.12	64.18
Lin et al. [39]	52.50	73.01	64.40	64.80
BSN-baseline [1]	-	74.16	66.17	66.26
+ improvement A	54.78	74.62	66.26	-
+ improvement B	55.23	75.62	67.17	67.34
+ improvement C	55.12	76.10	67.53	67.46
+ improvement D	55.66	76.45	67.88	67.99
+ improvement E	**56.91**	**77.30**	**68.92**	**69.30**

used as spatial network. Two-stream network is implemented using Caffe [38] and pretrained on ActivityNet-1.3 training set. During feature extraction, the interval σ of snippets is set to 16 on ActivityNet-1.3.

Since the duration of videos are limited, we follow [39] to rescale the feature sequence of each video to new length $l_w = 100$ by linear interpolation, and the duration of corresponding annotations to range [0,1]. In BSN, temporal evaluation module and proposal evaluation module are both implemented using Tensorflow [40]. Temporal evaluation module is trained with batch size 16 and learning rate 0.001 for 10 epochs, then 0.0001 for another 10 epochs, and proposal evaluation module is trained with batch size 256 and same learning rate. For Soft-NMS, we set the threshold θ to 0.8 by empirical validation, while ε in Gaussian function is set to 0.75 on both datasets.

4.2 Temporal Proposal Generation

The proposal performance on ActivityNet-1.3 of our method and previous state-of-the-art methods are shown in Table 1. Our method (BSN-baseline) has significantly better performance than previous method, and we further improve BSN in many aspects to achieve better performance. These improvements are introduced in the following, where each improvement is conducted based on the last improvement.

(A) *Threshold in proposal generation module*: In BSN-baseline, while generating proposals, we choose temporal locations as candidate boundary locations where boundary probability is higher than a threshold or is a probability peak. Here, we modify the threshold from 0.9 to $0.5 \cdot p_{max}$, where p_{max} is the maximum boundary probability in the video.

(B) *Video feature*: In BSN-baseline, we adopt two-stream network [35] pretrained on ActivityNet-1.3 for video feature extraction. Here, we further adopt two-stream network [32] and pseudo-3d network [43] pretrained on Kinetics-400 dataset for video feature extraction. To fuse these features, we train temporal evaluation mod-

Table 2: Action detection results on validation and testing set of ActivityNet-1.3 in terms of mAP@$tIoU$ and average mAP, where our proposals are combined with video-level classification results generated by [44].

	validation				testing
Method	0.5	0.75	0.95	Average	Average
Wang et al. [44]	42.28	3.76	0.05	14.85	14.62
SCC [45]	40.00	17.90	4.70	21.70	19.30
CDC [46]	43.83	25.88	0.21	22.77	22.90
TCN [41]	-	-	-	-	23.58
SSN [47]	39.12	23.48	5.49	23.98	28.28
Lin et al. [39]	48.99	32.91	7.87	32.26	33.40
BSN-baseline [1]	52.50	33.53	8.85	33.72	34.42
BSN-improved	**57.77**	**37.75**	**10.14**	**37.95**	**38.53**

ule using these features separately, then average the output of temporal evaluation module trained with different features.

(C) *Ensemble with SSAD-prop* [26, 39]: For better evaluating the confidence of proposals, we ensemble the results of BSN with the results of SSAD-prop [39]. For each proposal φ_{bsn} generated by BSN, we find a proposal φ_{ssad} generated by SSAD-prop which has maximum IoU. Then we fuse the confidence score of φ_{ssad} and φ_{bsn} via $p'_{bsn} = p_{bsn} \cdot p_{ssad}$, where p'_{bsn} is new confidence score of φ_{bsn}.

(D) *Prediction with original video duration*: In BSN-baseline, for convenience, we rescale the feature length to a fix new length $l_w = 100$. However, there are many short action instances, where the ratio between action duration and video duration is even lower than 0.01. To better capture these short action instances, during prediction, we use the original feature sequence instead of rescaled feature sequence.

(E) *Ensemble of original and fix video duration*: During analysis, we found that make prediction using original video length can benefit the recall performance of short action instances, however, can also damage the recall performance of long action instances. Thus, we make a combination between fix-scale and original-scale predictions: for fix-scale predictions, we take all proposals with duration larger than 25 seconds; for original-scale predictions, we take all proposals with duration smaller than 25 seconds. And we conduct Soft-NMS on combined proposals and output final results.

The experiment results of these improvements are shown in Table 1, which suggest that these improvements can bring salient performance promotion. With improved BSN, we finally achieve 69.30 of AUC in testing set, and win the second place of temporal action proposal generation task in ActivityNet Challenge 2018.

4.3 Action Localization with Our Proposals

To conduct temporal action localization, we put BSN proposals into "detection by classifying proposals" temporal action localization framework with state-of-the-art action

classifier, where temporal boundaries of detection results are provided by our proposals. We use top-1 video-level class generated by classification model [44] for all proposals in a video and keep BSN confidence scores of proposals for retrieving. And we use 100 proposals per video during temporal action detection.

In ActivityNet Challenge 2018, comparing with BSN-baseline, we also adopt improvements introduced above but with two differences: (1) first, we use rescaled video feature with $l_w = 64$ during prediction; (2) second, we set θ in Soft-NMS to 0 here. So why we make these adjustments? Since the localization metric (mAP) mainly depends on first several proposals (as discussed in our previous notebook [39]) and the proposal metric (AUC) depends on first 100 proposals, improvements or configurations which benefit proposal performance may harm localization performance. Thus, as in our previous notebook [39], we suggest that AR with small proposals amount should has higher weight in evaluation metric of proposal generation.

Experiment results shown in Table 2 suggest that our proposed method (BSN-baseline) has significantly better performance than previous state-of-the-art methods, and our new improvements can bring further performance promotion. With improved BSN, we finally achieve 38.52% of mAP in testing set, and win the first place of temporal action localization task in ActivityNet Challenge 2018.

5 Conclusion

In this challenge notebook, we have introduced our recent work: the Boundary-Sensitive Network (BSN) for temporal action proposal generation. Our method can generate proposals with flexible durations and precise boundaries via directly combing locations with high boundary probabilities, and make accurate retrieving via evaluating proposal confidence score with proposal-level features. Thus BSN can achieve high recall and high temporal overlap with relatively few proposals. And we also introduce the improvements we conducted during ActivityNet Challenge 2018, these improvements bring further performance promotion, and can also reveal the direction of how to make better temporal action proposal generation and localization.

References

1. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: Bsn: Boundary sensitive network for temporal action proposal generation. arXiv preprint arXiv:1806.02964 (2018)
2. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: Activitynet: A large-scale video benchmark for human activity understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 961–970
3. Jiang, Y.G., Liu, J., Zamir, A.R., Toderici, G., Laptev, I., Shah, M., Sukthankar, R.: Thumos challenge: Action recognition with a large number of classes. In: ECCV Workshop. (2014)
4. Buch, S., Escorcia, V., Shen, C., Ghanem, B., Niebles, J.C.: SST: Single-stream temporal action proposals. In: IEEE International Conference on Computer Vision. (2017)
5. Caba Heilbron, F., Carlos Niebles, J., Ghanem, B.: Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 1914–1923
6. Escorcia, V., Heilbron, F.C., Niebles, J.C., Ghanem, B.: Daps: Deep action proposals for action understanding. In: European Conference on Computer Vision, Springer (2016) 768–784
7. Gao, J., Yang, Z., Sun, C., Chen, K., Nevatia, R.: Turn tap: Temporal unit regression network for temporal action proposals. arXiv preprint arXiv:1703.06189 (2017)
8. Shou, Z., Wang, D., Chang, S.F.: Temporal action localization in untrimmed videos via multi-stage cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 1049–1058
9. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 3169–3176
10. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 3551–3558
11. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 1933–1941
12. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems. (2014) 568–576
13. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 4489–4497
14. Wang, L., Xiong, Y., Wang, Z., Qiao, Y.: Towards good practices for very deep two-stream convnets. arXiv preprint arXiv:1507.02159 (2015)
15. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2014) 580–587
16. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1440–1448
17. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. (2015) 91–99
18. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE transactions on pattern analysis and machine intelligence **32**(9) (2010) 1627–1645
19. Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. International journal of computer vision **104**(2) (2013) 154–171

20. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: European Conference on Computer Vision, Springer (2014) 391–405
21. Kuo, W., Hariharan, B., Malik, J.: Deepbox: Learning objectness with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 2479–2487
22. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. arXiv preprint arXiv:1612.03144 (2016)
23. Gidaris, S., Komodakis, N.: Locnet: Improving localization accuracy for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 789–798
24. Singh, G., Cuzzolin, F.: Untrimmed video classification for activity detection: submission to activitynet challenge. arXiv preprint arXiv:1607.01979 (2016)
25. Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Lin, D., Tang, X.: Temporal action detection with structured segment networks. arXiv preprint arXiv:1704.06228 (2017)
26. Lin, T., Zhao, X., Shou, Z.: Single shot temporal action detection. In: Proceedings of the 25nd ACM international conference on Multimedia. (2017)
27. Buch, S., Escorcia, V., Ghanem, B., Fei-Fei, L., Niebles, J.C.: End-to-end, single-stream temporal action detection in untrimmed videos. In: Proceedings of the British Machine Vision Conference. (2017)
28. Karaman, S., Seidenari, L., Del Bimbo, A.: Fast saliency based pooling of fisher encoded dense trajectories. In: ECCV THUMOS Workshop. (2014)
29. Oneata, D., Verbeek, J., Schmid, C.: The lear submission at thumos 2014. ECCV THUMOS Workshop (2014)
30. Wang, L., Qiao, Y., Tang, X.: Action recognition and detection by combining motion and appearance features. THUMOS14 Action Recognition Challenge **1** (2014) 2
31. Yuan, Z., Stroud, J.C., Lu, T., Deng, J.: Temporal action localization by structured maximal sums. arXiv preprint arXiv:1704.04671 (2017)
32. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: towards good practices for deep action recognition. In: European Conference on Computer Vision, Springer (2016) 20–36
33. Gao, J., Yang, Z., Nevatia, R.: Cascaded boundary regression for temporal action detection. arXiv preprint arXiv:1705.01180 (2017)
34. Bodla, N., Singh, B., Chellappa, R., Davis, .L.S.: Improving object detection with one line of code. arXiv preprint arXiv:1704.04503 (2017)
35. Xiong, Y., Wang, L., Wang, Z., Zhang, B., Song, H., Li, W., Lin, D., Qiao, Y., Gool, L.V., Tang, X.: Cuhk & ethz & siat submission to activitynet challenge 2016. arXiv preprint arXiv:1608.00797 (2016)
36. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. (2015) 448–456
37. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
38. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia, ACM (2014) 675–678
39. Lin, T., Zhao, X., Shou, Z.: Temporal convolution based action proposal: Submission to activitynet 2017. arXiv preprint arXiv:1707.06750 (2017)
40. Abadi, M., Agarwal, A., Barham, P., et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)

41. Dai, X., Singh, B., Zhang, G., Davis, L.S., Chen, Y.Q.: Temporal context network for activity localization in videos. In: 2017 IEEE International Conference on Computer Vision (ICCV), IEEE (2017) 5727–5736
42. Ghanem, B., Niebles, J.C., Snoek, C., Heilbron, F.C., Alwassel, H., Khrisna, R., Escorcia, V., Hata, K., Buch, S.: Activitynet challenge 2017 summary. arXiv preprint arXiv:1710.08011 (2017)
43. Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with pseudo-3d residual networks. In: 2017 IEEE International Conference on Computer Vision (ICCV), IEEE (2017) 5534–5542
44. Wang, R., Tao, D.: Uts at activitynet 2016. AcitivityNet Large Scale Activity Recognition Challenge **2016** (2016) 8
45. Heilbron, F.C., Barrios, W., Escorcia, V., Ghanem, B.: Scc: Semantic context cascade for efficient action detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 2. (2017)
46. Shou, Z., Chan, J., Zareian, A., Miyazawa, K., Chang, S.F.: Cdc: Convolutional-deconvolutional networks for precise temporal action localization in untrimmed videos. arXiv preprint arXiv:1703.01515 (2017)
47. Xiong, Y., Zhao, Y., Wang, L., Lin, D., Tang, X.: A pursuit of temporal accuracy in general activity detection. arXiv preprint arXiv:1703.02716 (2017)

Action Pyramid Networks for Proposal and Activity Detection: Submission to ActivityNet Challenge 2018 Task1 and Task 2

Xiao Liu, Fan Yang, Xin Li, Jun Yu, Rujiao Long, Xiang Long and Shilei Wen
Baidu Vis

Abstract

This report details our solution to ActivityNet 2018 Task 1 (temporal action proposals) and Task 2 (temporal action localization). For both tasks, we focus on exploring end-to-end trained networks with two-stream features as input. To this end, we devise a novel Action Pyramid Networks (APN), which enjoys three favorable properties when compared with conventional methods. First, 2D convolutions are carried out on the input two stream features such that the temporal-channel patterns are jointly modeled. Second, a feature pyramid architecture is exploited to enlarge the receptive field of small proposals. Third, multi-scale anchor boxes and ROI poolings are combined to generate enough proposal candidates. The APN improves the state-of-the-art temporal action proposal and temporal action detection performance.

1. Approach Overview

We extract RGB and optical flow features from image frames, and use the two stream features as the input of APN. Soft-NMS is conducted on the output of APN to produce the final output.

1.1. Features

We use two stream features [5], include RGB features for spatial information and stacked optical flow fields for temporal information. For both streams, we train frame-level classifiers with 201-way outputs. Similar with [8], the classifiers are trained based on the Temporal Segment Network [7]. The annotated action instances are regarded as positive samples of the 200 action classes, and the regions between annotated instances are regarded as the samples of the background class. We try different ConvNet architectures and find Inception-ResNet-v2 [6] pre-trained on Kinetics-400 outperform others in both spatial and temporal components. After feature training, we densely extract the output from the last pooling layer of Incepiton-Resnet-v2 at 5-fps for further processes. The length of each feature sequence is resized to 512 by linear interpolation before putting into the APN, such that inputs of a training video to the APN are two 512×1536 sequences, where the two 1536-dimensional vectors are RGB and flow features. We also try using the densely extracted audio features [1], but find it help slightly.

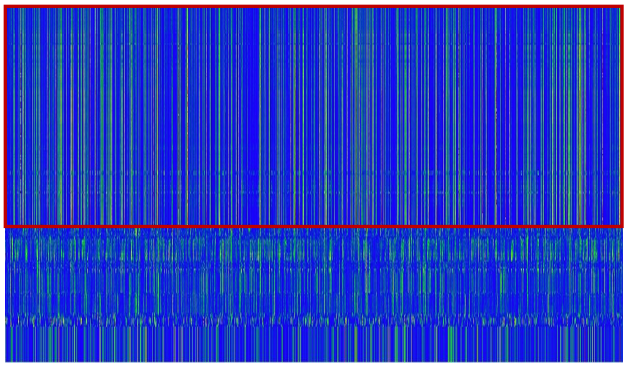


Figure 1. The "temporal-channel" image of a input video. The horizontal axis shows different channels and the vertical axis is the temporal axis. Lighter colors mean stronger activation values. The red box indicates an action instance.

1.2. Action Pyramid Networks

Temporal-Channel 2D Convolution. The input of each stream is regarded as a 512×1536 image with single channel, we thus can use 2D ConvNet to process the "temporal-channel" image. Figure 1 shows an example of "temporal-channel" image, where the horizontal axis shows different channels and the vertical axis is the temporal axis. Lighter colors mean stronger activation values. The red box indicates an action instance. From the example, we have two observations: 1) The action region has a strong pattern that differs from the background region. 2) Directly concatenating all the 1536 channels as a pattern is sensitive to unexpected values, while concatenating a local part of the channels is more stable thanks to the invariance brought by deep ConvNet. From these observations, the APN is built by stacking very deep local 2D convolutions, which is different from most previous end-to-end trained methods that use relative shallow 1D convolutions.

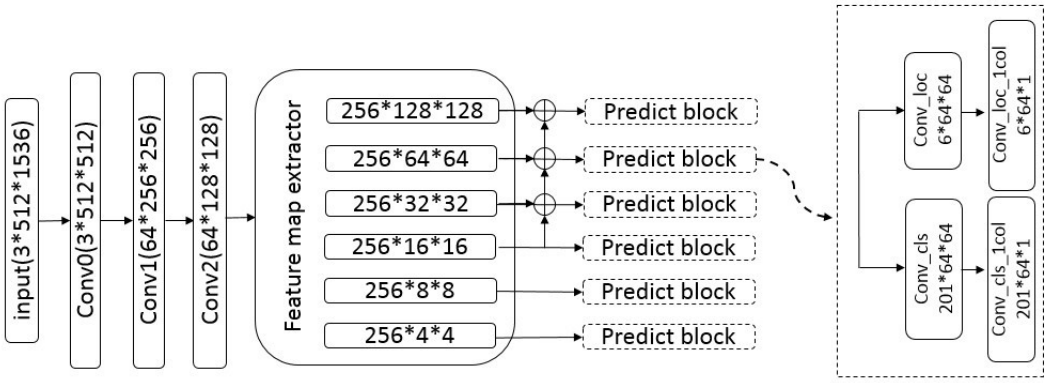


Figure 2. The architecture of APN.

Multi-Scale Feature Pyramid. A convolution with 2D kernel $(1, 9)$ and stride $(1, 3)$ is used to make the input into a 512×512 feature map. Two $(3, 3)$ convolutions with stride (2, 2) are then used to generate a 128×128 feature map. We then regard it as a standard input of 2D FPN [2] with multi-scale feature pyramid. A 6-scale feature pyramid with side lengths of $\{128, 64, 32, 16, 8, 4\}$ is generated, and on the top of each anchor feature map, a convolution layer is used to squeeze the channel axis. E.g., for the finest 128×128 feature map, we use a convolution with 2D kernel $(128, 1)$ to project it into a new size of 128. Top-down lateral connections are added between adjacent scales to enlarge the receptive filed of proposals in high-resolution anchor feature maps. Figure 2 shows the architecture of proposed APN.

Multi-Scale Anchor Boxes and ROI Pooling. For each anchor feature map, we use a ROI pooling layer to generate candidate proposals. We use a ROI pooling with pooling size of 4 pixels and padding size of 2 pixels. An example is shown in Figure 3. The ROI pooling layer generates 5 candidate positions on the 4×1 feature map.

Similar with [3], we also use multi-scale anchor boxes. Combining multi-scale anchor boxes and ROI poolings increase the number of candidate positions and enlarge the receptive filed of proposals.

Loss Function. We use two loss functions, the classification loss function and the regression loss function, which shares the same definition as most previous work.

2. Experiment Results

We add 2000 validation videos to the training set, and leave the others for validation.

For task 1, in our reduced validation set, the "temporal-channel" APN achieves 67.5 AR-AN score. We also train a 1D APN with only temporal convolution, and achieves the AR-AN score of 66.8. Merging the two models and a TAG model [8] by Soft-NMS [4], we achieves 70.03 on the validation set, and 70.99 on the testing server.

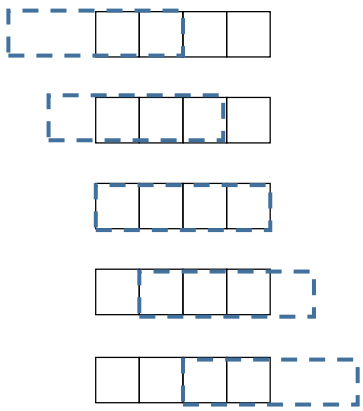


Figure 3. A ROI pooling layer generates 5 candidate positions on the 4×1 feature map.

For task 2, APN achieves 35.1 mAP in our reduced validation set, and 35.27 on the testing server.

3. Conclusions

In this work, we propose Action Pyramid Network for temporal proposal and activity detection. We introduces a novel "temporal-channel" convolution for this task, and uses pyramid hierarchy, ROI pooling and multi-scale anchor prediction for obtaining high quality proposals. The experiments demonstrate the effectiveness of APN. Codes and more details will be released soon.

References

[1] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson. Cnn architectures for large-scale audio classification. In *arXiv preprint arXiv:1609.09430*, 2017.

[2] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[3] T. Lin, X. Zhao, and Z. Shou. Single shot temporal action detection. In *ACM Multimedia*, 2017.

[4] R. C. N. Bodla, B. Singh and L. Davis. Soft-nms – improving object detection with one line of code. In *ICCV)*, 2017.

[5] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[6] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *arXiv preprint arXiv:1602.07261*, 2016.

[7] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[8] Y. Xiong, Y. Zhao, L. Wang, D. Lin, and X. Tang. A pursuit of temporal accuracy in general activity detection. In *CVPR*, 2017.

Temporal Action Proposals & Localization: Submission to ActivityNet Challenge 2018

Xiaoning Liu
liuxiaoning_sx@qiyi.com

Abstract

In this paper, a brief description is provided of the method used for the task of temporal action proposals(task1) & localization(task2). Based on Single-Stream Temporal Action Proposals （SST） and Temporal Relation Network（TRN）classifier, we propose a joint optimization strategy for two tasks, which use the classification results of the proposals generated by TRN to optimize the proposals and then use the optimized proposals to generate new location results. The results prove that there is an effective improvement of the original results.

1. INTRODUCTION

Recognizing activities in videos is a challenging task as video is an information-intensive media with complex variations, especially when the duration of videos and the number of activities are varied. The ActivityNet 1.3 dataset includes about 20000 videos. The video durations vary from less than 10s to more than 300s and the number of activities per video is about 2, which means some of them have more than five activities and some others have none. The current temporal action proposals algorithms choose the segment just based on the proposal scores of candidate proposals. This may cause a decline in the confidence of the result. Meanwhile, the performance of the location task 2 is heavily dependent on the task 1. Therefore, we propose a strategy of joint optimization of task 1 and task 2. Firstly, the C3D feature of activitynet1.3 can be generated by the C3D network. Secondly, the SST is used to generate the original candidate proposals. Thirdly, the top 100 candidate proposals of each video are chosen to get the classification score using TRN. Fourthly, the re-rank results of original candidate proposals are given by combining the proposal score and the classification score. Finally, the results of the location task can be get by using TRN on the optimized proposals. In addition, we also tried to optimize some hyper-parameters of the SST and TRN.

2.Method Description

2.1 SST

The SST is an excellent time dimension algorithm. The advantages of SST include: first, it can handle long video sequences with only one forward propagation to process the entire video(online), and it can handle video of any

length without dealing with overlapping time windows. Second, it achieved an excellent result on proposal generation task. Particularly, the SST proposals provide a strong benchmark for temporal action localization. Combining this approach with existing classification tasks can improve the classification performance. The architecture of the SST is composed of the C3D visual encoder and the GRU sequence encoder. In the specific implementation, different number of anchors and different sampling lengths are compared, where the best NMS threshold and score threshold are used.

2.2 TRN

Temporal relational reasoning is critical for activity recognition, forming the building blocks for describing the steps of an event. A single activity can consist of several temporal relations at both short-term and long-term timescales, the ability to model such relations is very important for activity recognition. The Temporal Relation Network (TRN) proposed by Bolei Zhou et al [2] is designed to learn and reason about temporal dependencies between video frames at multiple time scales. It is an effective and interpretable network which is able to learn intuitive and interpretable visual common sense knowledge in videos. The networks used for extracting image features is very important for visual recognition tasks, here we use an 8 segment multi-scale TRN with an inceptionV3 base and Inception with Batch Normalization (BN-Inception) base separately.

2.3 Ensemble

The proposals result of the SST have more than 100 candidate proposals per video. We consider whether it is possible to use another criterion to improve the confidence of the score of the model. Not difficult to understand that the high probability activity segment corresponds to a high probability classification score in the case of the classification is reliable. Here the proposals score of the SST and the classification score by the TRN of the original proposals are combined with different weights after being normalized. In particular, the classification results involve different number of top proposals and average score criterion. Then we can get the re-rank proposals results for task 1. After that, we can get the better location results based on the optimized proposals result.

3.Experimental Results

The AcvivityNet 1.3 contains 10024 training videos (15410 instances), 4926 validation videos (7654 instances) and 5044 testing videos (labels withheld). The instances are divided into 200 categories. The performance on the validation dataset of AcvivityNet 1.3 are as table 1:

Task	Model	Performance
1	SST	29.6(AUC)
	SST+TRN	38.3(AUC)
	fine-tune	42.9(AUC)
2	TRN(original)	0.114(mAP)
	TRN(re-rank)	0.132(mAP)

Table1: AcrivityNet 1.3 validation results

4.Conclusion

The SST has been proved that it is an excellent time dimension algorithm on proposals task. The recently proposed TRN method is effective for recognizing daily activities with learning intuitive and interpretable visual common sense knowledge in videos. The joint optimization strategy can better combine the advantages of both the SST and the TRN. So the proposals task and the location task have achieved better results than independent, and there is still a lot of space for optimization in this way.

5.Acknowledgement

This work was finished during an internship work in IQIYI，many thanks to Jie Liu, Tao Wang，and Dongyang Cai for helpful comments and discussion.

References

[1] Buch, Shyamal, et al. "Sst: Single-stream temporal action proposals." *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017.

[2] Zhou, Bolei, Alex Andonian, and Antonio Torralba. "Temporal Relational Reasoning in Videos." *arXiv preprint arXiv:1711.08496* (2017).

[3] Qiu, Zhaofan, Ting Yao, and Tao Mei. "Learning spatio-temporal representation with pseudo-3d residual networks." *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017.

[4]

SIAT-USTC Submission to ActivityNet Challenge 2018

Xin Yu[1,2], Yifan Yang[1], Linjie Xing[1], Ou Xiaoxuan[2], Xiaojiang Peng[1], and Yu Qiao[1]

[1]Shenzhen Institutes of Advanced Technology, CAS, China
[2]University of Science and Technology of China,China

ABSTRACT

This paper presents the method for our submission to temporal action proposal (task 1) and temporal action localization (task 2) of ActivityNet Challenge 2018.

1 Our Method

1.1 Feature Extraction

In our approach, a long untrimmed video is decomposed into video units, which are reused as basic building blocks of temporal proposals. We extract two-stream features in a similar way described in CBR [1]. We adopt two-stream network which is pre-trained on ActivityNet v1.3 training set.We segment video into unit snippets without overlap. In each snippet, we use spatial network to extract appearance feature with 8 interval frames, and we use the output of "Flatten-673" layer in ResNet network as feature. For motion feature, we compute optical flows using 6 consecutive frames around the center frame of a snippet, then these optical flows are used for extracting motion feature with temporal network, where the output of "global-pool" layer in BN-Inception network is used as feature.

1.2 Temporal Action Proposals

Temporal Boundary-aware Action Proposal (TBAP). We use out Temporal Boundary-aware action proposal as fundamental temporal action proposal. TBAP use the framework similar to TURN-TAP[2], we jointly predicts action proposals and re-fines the temporal boundaries by temporal coordinate regression. There are two salient aspects of TBAP : (1) boundary-aware proposal feature: We use our method similar to temporal pyramid pooling to concatenate unit features to the proposal feature; (2)ratio clip pyramid: we made our clip pyramid in ratio according to the ground truth distributing in training dataset.

Start-End Action Proposal (SEAP). To catch the boundary of potential proposals, we develop a network called Start-End Proposal Net, which consists of two networks. The Start net mainly for detecting the start boundary, is formed using an inner product layer, the input of which is features extracted by TSN. The End net has the same structure as the Start net, mainly for detecting the end boundary of actions. The outputs of both Start net and End net is the probability of boundary (start and end respectively) and background. We set a constant value as threshold to select high quality boundaries and filter out noise. Then, we group high quality start boundaries and end boundaries according to its position in the video to generate proposals.

Prop-SSAD. Prop-SSAD[3] is a proposal generate method which simultaneously conducts temporal action proposal and recognition,and has been used in last year challenge task3 winner method.

1.2.1 Proposal Fusion

We use normalized score to re-ranking different temporal action proposal generated from three methods . In addition, we analyze 4 kinds of proposal distribution weight (length ratio,length,start position,end position) from training dataset and use it in re-ranking proposals.

1.3 Temporal Action Localization

Since most videos in ActivityNet dataset only contain one action category, we use video-level action classification result of winner in ActivityNet challenge 2016 recognition task[4] as the category of temporal action proposals to get temporal action

localization result.

2 Experimental Results

Method	AR-AN
Prop-SSAD	61.52
TBAP	61.84
Fusion TBAP-SEAP	63.58
Fusion TBAP-SEAP-SSAD	65.38

Table 1. Proposal Results on validation set of ActivityNet.

References

1. Gao, J., Yang, Z. & Nevatia, R. Cascaded boundary regression for temporal action detection. *arXiv preprint arXiv:1705.01180* (2017).

2. Gao, J., Yang, Z., Sun, C., Chen, K. & Nevatia, R. Turn tap: Temporal unit regression network for temporal action proposals. *arXiv preprint arXiv:1703.06189* (2017).

3. Lin, T., Zhao, X. & Shou, Z. Single shot temporal action detection. In *Proceedings of the 2017 ACM on Multimedia Conference*, 988–996 (ACM, 2017).

4. Xiong, Y. *et al.* Cuhk & ethz & siat submission to activitynet challenge 2016. *arXiv preprint arXiv:1608.00797* (2016).

ActivityNet 2018: Temporal Action Proposal Challenge

Yuan Liu[†] Yongyi Tang[♮] Jingwen Wang[♯]
[‡]Southeast University [♮]Sun Yat-Sen University [♯]South China University of Technology

Abstract

This technical report presents an overview of our methods designed for the task of temporal action proposals in ActivityNet Challenge 2018. A three-stage workflow is particularly devised. In order to capture long time proposals, we use 7 anchor layers with different time resolution to detect proposals with various time length. We then use Temporal Actionness Grouping(TAG) method to modify the boundries of the proposals. Finally, a temporal convolutional network is proposed to rank the generated proposals, which can further boost the performances. Our approach achieves 64.93 on AUC on testing set.

1. Our Approach

Our framework working in three stages is designed to generate the temporal action proposals. In the following, we will introduce each component, namely the single shot action detector (SSAD) [1], temporal actionness grouping (TAG)[5], and proposal reranking, respectively.

1.1. SSAD

SSAD is inspired by YOLO [4] and SSD [2] network for object detection task. By using the multi anchor mechanism based on temporal convolutional layers, the network is able to detect action proposals with different time length. Specifically, the lower anchor layers are of smaller receptive field and higher time resolution when compared with the higher anchor layers. Thus, the lower anchor layers are used to detect shorter action proposals with the higher layers focusing on detecting longer action proposals. By the co-operation of several anchor layers, proposals with various time length will be detected. As for details, we use 7 anchors layers and each has 512 feature maps.

1.2. TAG

The core idea of TAG is acquiring a probability score for every snippet and then grouping them into region proposals with multiple thresholds. We train a multi-layer perceptron based on two hidden layers to give a score for each snippet. TAG , as a bottom-up model that relies on actionness grouping, is more boundary sensitive than SSAD and can give a boundary refinement to the proposals predicted by SSAD.

1.3. Proposal Reranking

We propose a rank model to refine the the probability scores of each proposals. SSAD will give a probability score which is the basis of position ranking. However, the probability score is not accurate and we propose a rank model to refine it. By giving a more reliable score for every predicted proposal, the performance (ie. area under the Average Recall vs. Average Number of Proposals per Video (AR-AN) curve with 100 proposals) will be improved.

1.3.1 Global representation

Due to the uneven distribution of the proposal time, we use pyramid pooling to acquire the global representation which is inspired by [6]. Specially, for a proposal with starting time s and ending time e, a series of snippets are included. In this work, we use a feature extractor first proposed in [3] to get the P3D feature p_t for each proposal. We use two stages to acquire proposal representation with different time resolution. For the first stage, the action proposal is divided into 5 intervals on average. For the i-th segment, it is denoted as $[s_{1i}, e_{1i}]$. The corresponding pooled feature is denoted as

$$u_1^i = \frac{1}{e_{1i} - s_{1i}} \sum_{t=s_{1i}}^{e_{1i}} p_t \tag{1}$$

By concatenating the pooled features across the 5 intervals of this stage, the representation is acquired, represented as

$$f_1 = [u_1^1, u_1^2, u_1^3, u_1^4, u_1^5] \tag{2}$$

As for the second stage, the action proposal is divided into 2 intervals on average and the corresponding feature representation will have larger receptive field, denoted as

$$f_2 = [u_2^1, u_2^2] \tag{3}$$

The global representation will be the concatenation of the two stages and is represented as $f = [f_1, f_2]$

1.3.2 classifier

The rank model use two types of classifiers, an activity classifier and a completeness classifier to rerank the sequence of proposals. The activity classifier is to give a discrimination between background proposals and the others. The completeness classifier is to predict whether the proposals are complete. As for the completeness filter, context information is included in the input representation which is crucial for the discrimination of completeness.

For both of the two classifiers, they need to output higher scores for positive instances and lower scores for negative instances. During training, suppose that we have a set of pairs $K_i = (p_i, n_i)$. Take the completeness classifier for example, p_i means a complete instance and n_i means a incomplete instance. Our goal is to train the two classifiers which assign higher scores for positive instances, which can be expressed as

$$f(p_i) \succ f(n_i), \qquad \forall (p_i, n_i) \in K \tag{4}$$

The ranking loss function is defined as:

$$min : \sum_{(p_i, n_i)} max(0, 1 - f(h_i) + f(n_i)) \tag{5}$$

2. Experiments results

2.1. Evaluation Metrics

As for the task of temporal action proposal, the area under the Average Recall (AR) vs. Average Number of Proposals (AN) per Video curve (AUC) is adopted as the evaluation metric, where AR is defined as the mean of all recall values using tIoU between 0.5 and 0.9 with a step size of 0.05. AN is defined as the total number of proposals divided by the number of videos in the testing subset.

2.2. Temporal Action proposal

The performance of TAG and SSAD on the validation set of Activitynet is shown in Table 1 and Table 2. The performances with different features, including inception_resnet, inceptionV4, I3d and P3d, vary significantly. It can also observed that with additionally incorporating the reranking model, the performances of both TAG and SSAD can be consistently improved.

TAG, regarded as a bottom-up model, is more sensitive to temporal boundaeries than SSAD. For each proposals t_i in TAG, we calculate its Iou with all proposals in SSAD. If the maximum Iou is higher than a threshold ϕ, we replace the corresponding proposals p_s in SSAD. As for the selection of threshold ϕ, we find 0.6 is most suitable for this problem which is shown in Table 3, the best performance is 66.01 by combining TAG and SSAD.

Table 1. The AUC of SSAD with multiple features on ActivityNet validation set for temporal proposals task.

Network	Features	Re_ranking	AUC
TAG	Inception_resnet		56.37
TAG	InceptionV4		56.9
TAG	I3d		58.243
TAG	P3d		57.16
TAG	Inception_resnet	$\surd$	56.92
TAG	InceptionV4	$\surd$	57.8
TAG	I3d	$\surd$	59.51
TAG	P3d	$\surd$	59.91
Fusion_all			60.61

Table 2. The AUC of TAG with multiple features on ActivityNet validation set for temporal proposals task.

Network	Features	Re_ranking	AUC
TAG	Inception_resnet		51.335
TAG	InceptionV4		50.8
TAG	I3d		53.298
TAG	P3d		56.21
TAG	Inception_resnet	$\surd$	53.635
TAG	InceptionV4	$\surd$	54.1
TAG	I3d	$\surd$	56.632
TAG	P3d	$\surd$	59.01
Fusion_all			62.07

Table 3. The AUC of boundary refinement based on TAG and SSAD on ActivityNet validation set for temporal proposals task. The threshold ϕ varies from 0.5 to 0.9.

Network	ϕ	AUC
TAG+SSAD	0.5	65.71
TAG+SSAD	0.6	66.01
TAG+SSAD	0.7	65.62
TAG+SSAD	0.8	65.10
TAG+SSAD	0.9	64.32

3. Conclusion

In ActivityNet Challenge 2018, we mainly focus on boundary refinement and reranking model to improve the performance of AUC based on multiple visual features. In the future work, we will further study how to make the network more sensitive to boundariesaOur approach achieves 64.93 on temporal action proposal task on the testing set.

References

[1] T. Lin, X. Zhao, and Z. Shou. Single shot temporal action detection. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 988–996. ACM, 2017.

[2] W. Liu, D. Anguelov, D. Erhan, et al. Ssd: Single shot multibox detector. In *European conference on computer vision (ECCV)*, pages 21–37. Springer, 2016.

[3] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5534–5542. IEEE, 2017.

[4] J. Redmon, S. Divvala, R. Girshick, et al. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 779–788, 2016.

[5] Y. Xiong, Y. Zhao, L. Wang, et al. A pursuit of temporal accuracy in general activity detection. *arXiv preprint arXiv:1703.02716*, 2017.

[6] Y. Zhao, Y. Xiong, L. Wang, et al. Temporal action detection with structured segment networks. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 8, 2017.

Best Vision Technologies Submission to ActivityNet Challenge 2018 —Task: Dense-Captioning Events in Videos

Yuan Liu and Moyini Yao
Best Vision Technologies Co., Ltd, Beijing, China
{yuan.liu, ting.yao}@y-cv.com

Abstract

This note describes the details of our solution to the dense-captioning events in videos task of ActivityNet Challenge 2018. Specifically, we solve this problem with a two-stage way, i.e., first temporal event proposal and then sentence generation. For temporal event proposal, we directly leverage the three-stage workflow in [13, 16]. For sentence generation, we capitalize on LSTM-based captioning framework with temporal attention mechanism (dubbed as LSTM-T). Moreover, the input visual sequence to the LSTM-based video captioning model is comprised of RGB and optical flow images. At inference, we adopt a late fusion scheme to fuse the two LSTM-based captioning models for sentence generation.

1. Sentence Generation Model

Inspired from the recent successes of LSTM based sequence models leveraged in image/video captioning [1, 3, 5, 6, 7, 10, 11, 12, 14, 15], we formulate our sentence generation model in an end-to-end fashion based on LSTM which encodes the input frame/optical flow sequence into a fixed dimensional vector via temporal attention mechanism and then decodes it to each target output word. An overview of our sentence generation model is illustrated in Figure 1.

In particular, given the input video with frame and optical flow sequences, each input frame/optical flow sequence ($\{\mathbf{v}_i^{(1)}\}_{i=1}^K$) is fed into a two-layer LSTM with attention mechanism. At each time step t, the attention LSTM decoder firstly collects the maximum contextual information by concatenating the input word w_t with the previous output of the second-layer LSTM unit $\mathbf{h}_{t-1}^2$ and the mean-pooled video-level representation $\overline{\mathbf{v}} = \frac{1}{K}\sum_{i=1}^{K}\mathbf{v}_i^{(1)}$, which will be set as the input of the first-layer LSTM unit. Hence the updating procedure for the first-layer LSTM unit is as

$$\mathbf{h}_t^1 = f_1\left(\left[\mathbf{h}_{t-1}^2, \mathbf{W}_s\mathbf{w}_t, \overline{\mathbf{v}}\right]\right), \tag{1}$$

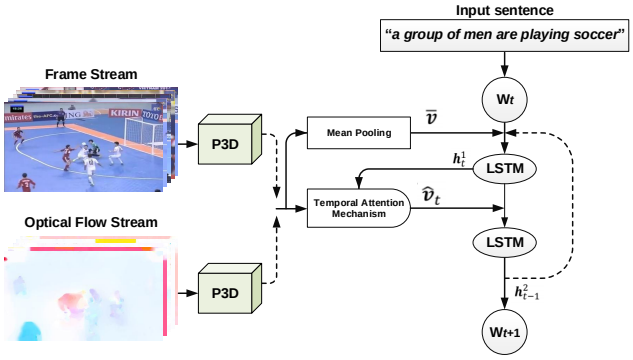


Figure 1. The sentence generation model in our system for dense-captioning events in videos task.

where $\mathbf{W}_s \in \mathbb{R}^{D_s^1 \times D_s}$ is the transformation matric for input word w_t, $\mathbf{h}_t^1 \in \mathbb{R}^{D_h}$ is the output of the first-layer LSTM unit, and f_1 is the updating function within the first-layer LSTM unit. Next, depending on the output $\mathbf{h}_t^1$ of the first-layer LSTM unit, a normalized attention distribution over all the frame/optical flow features is generated as:

$$\begin{aligned} a_{t,i} &= \mathbf{W}_a\left[\tanh\left(\mathbf{W}_f\mathbf{v}_i^{(1)} + \mathbf{W}_h\mathbf{h}_t^1\right)\right], \\ \lambda_t &= softmax\left(\mathbf{a}_t\right), \end{aligned} \tag{2}$$

where $a_{t,i}$ is the i-th element of $\mathbf{a}_t$, $\mathbf{W}_a \in \mathbb{R}^{1\times D_a}$, $\mathbf{W}_f \in \mathbb{R}^{D_a \times D_v}$ and $\mathbf{W}_h \in \mathbb{R}^{D_a \times D_h}$ are transformation matrices. $\lambda_t \in \mathbb{R}^K$ denotes the normalized attention distribution and its i-th element $\lambda_{t,i}$ is the attention probability of $\mathbf{v}_i^{(1)}$. Based on the attention distribution, we calculate the attended video-level representation $\hat{\mathbf{v}}_t = \sum_{i=1}^{K}\lambda_{t,i}\mathbf{v}_i^{(1)}$ by aggregating all the frame/optical flow features weighted with attention. We further concatenate the attended video-level feature $\hat{\mathbf{v}}_t$ with $\mathbf{h}_t^1$ and feed them into the second-layer LSTM unit, whose updating procedure is thus given as:

$$\mathbf{h}_t^2 = f_2\left(\left[\hat{\mathbf{v}}_t, \mathbf{h}_t^1\right]\right), \tag{3}$$

where f_2 is the updating function within the second-layer LSTM unit. The output of the second-layer LSTM unit $\mathbf{h}_t^2$ is

Table 1. Performance on ActivityNet captions validation set, where B@N, M, R and C are short for BLEU@N, METEOR, ROUGE-L and CIDEr-D scores. All values are reported as percentage (%).

Model	B@1	B@2	B@3	B@4	M	R	C
LSTM-T$_{frame}$	12.71	7.24	4.01	1.99	8.99	14.67	13.82
LSTM-T$_{opt}$	12.46	7.08	3.96	1.97	8.72	14.55	13.60
LSTM-T	13.19	7.75	4.48	2.31	9.26	15.18	14.97

leveraged to predict the next word w_{t+1} through a softmax layer. Note that the policy gradient optimization method with reinforcement learning [4, 9] is additionally leveraged to boost the sentence generation performances specific to METEOR metric.

2. Experiments

2.1. Features and Parameter Settings

Each word in the sentence is represented as "one-hot" vector (binary index vector in a vocabulary). For the input video representations, we take the output of 2048-way $pool5$ layer from P3D ResNet [8] pre-trained on Kinetics dataset [2] as frame/optical flow representation. The dimension of the hidden layer in each LSTM D_h is set as 1,000. The dimension of the hidden layer for measuring attention distribution D_a is set as 512.

2.2. Results

Two slightly different settings of our LSTM-T are named as LSTM-T$_{frame}$ and LSTM-T$_{opt}$ which are trained with only frame and optical flow sequence, respectively. Table 1 shows the performances of our models on ActivityNet captions validation set. The results clearly indicate that by utilizing both frame and optical flow sequences in a late fusion manner, our LSTM-T boosts up the performances.

3. Conclusions

In this challenge, we mainly focus on the dense-captioning events in videos task and present a system by leveraging the three-stage workflow for temporal event proposal and LSTM-based captioning model with temporal attention mechanism for sentence generation. One possible future research direction would be how to end-to-end formulate the whole dense-captioning events in videos system.

References

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

[2] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[3] Y. Li, T. Yao, Y. Pan, H. Chao, and T. Mei. Jointly localizing and describing events for dense video captioning. In *CVPR*, 2018.

[4] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy. Optimization of image description metrics using policy gradient methods. In *ICCV*, 2017.

[5] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, 2016.

[6] Y. Pan, Z. Qiu, T. Yao, H. Li, and T. Mei. Seeing bot. In *SIGIR*, 2017.

[7] Y. Pan, T. Yao, H. Li, and T. Mei. Video captioning with transferred semantic attributes. In *CVPR*, 2017.

[8] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 2017.

[9] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. *arXiv preprint arXiv:1612.00563*, 2016.

[10] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence-video to text. In *ICCV*, 2015.

[11] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[12] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *ICCV*, 2015.

[13] T. Yao, Y. Li, Z. Qiu, F. Long, Y. Pan, D. Li, and T. Mei. Msr asia msm at activitynet challenge 2017: Trimmed action recognition, temporal action proposals and dense-captioning events in videos. In *CVPR ActivityNet Challenge Workshop*, 2017.

[14] T. Yao, Y. Pan, Y. Li, and T. Mei. Incorporating copying mechanism in image captioning for learning novel objects. In *CVPR*, 2017.

[15] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei. Boosting image captioning with attributes. In *ICCV*, 2017.

[16] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, D. Lin, and X. Tang. Temporal action detection with structured segment networks. *arXiv preprint arXiv:1704.06228*, 2017.

ActivityNet 2018: Dense-Captioning Events in Videos Challenge

Bairui Wang[†] Yuan Liu[‡] Yang Feng[♭] Jingwen Wang[♮] Wei Zhang[†] Wenhao Jiang[♯] Lin Ma[♯]
[†]Shandong University [‡]Southeast University [♭]University of Rochester [♮]South China University of Technology [♯]Tencent AI Lab
bairuiwong@gmail.com

Abstract

In this paper, we describe the details of our approaches on the challenge of dense-captioning events in videos. Based on TA model [6], we employ a temporal encoder [7] and a reconstructor [3], and trained models by the REINFORCE algorithm [5]. As such the global and local semantic information, temporal relationships and backward flow information in video clips are fully captured. The evaluation metric, specifically the METEOR, is optimized directly, which contributes greatly to the performance improvements. Afterwards, we build a ensemble model trained with different features and re-rank [4] the final results with a fusion score computed by confidence score of predicted caption and its corresponding proposal score. Our approach achieves 8.1057 on METEOR on testing set.

1. Approaches

In this section we first describe our base model TA for video captioning briefly. Afterwards, the incorporated components are introduced in the following.

1.1. Model

Temporal Attention (TA) Captioning Model. The TA model [6] learns local temporal structure of videos from C3D features and yields the global structure by employing the soft-attention strategy which learns to select the most relevant temporal segments. Given the previous state h_{t-a} of RNN and video frame features $\mathbf{V} = \{v_1, v_2, ..., v_n\}$, the weight for each frame feature is learned as follow:

$$\begin{aligned} e_i^t &= w_T \tanh\left(W h_{t-1} + U v_i + b\right) \\ \alpha_i^t &= exp\{e_i^t\} / \sum_{j=1}^{n} exp\{e_i^t\} \end{aligned} \tag{1}$$

where the e_i^t and the α_i^t are relevance score and normalized weight for the i_{th} frame feature on t_{th} step, respectively. At last, the video feature is obtained by attentively weighting each frame feature:

$$\varphi_t(\mathbf{V}) = \sum_{i=1}^{n} \alpha_i^t v_i \tag{2}$$

where $\varphi_t(V)$ represents the obtained global video feature at t_{th} step. The decoder, realized by LSTM, relies on $\varphi_t(V)$ as well as previous hidden states and predicted word to generate the word at each time step.

Temporal Encoding. However, we find that soft-attention mechanism in TA only decides which frames are more important than the others in current time but does not care whether the order of input frame sequence is correct or not. Inspired by [7], we proposed to attentively encode the video frames. Specifically, we employ the LSTM as a temporal encoder working on the yielded video frame representation. The video frames are fed into LSTM, with the hidden state of each step are taken as the high-level semantic features. The soft-attention strategy performs on the high-level semantic features to generate the video global structure.

Reconstruction. In addition to the above information, we can also mine other information of the training data, such as forward and backward flow information between video and ground-truth description. Following [3], we build a reconstructor on top of the decoder. It takes the hidden state sequence of the decoder as input and reproduces the visual features of the video. Mean pooling is operating on the state sequence of the reconstructor to reconstruct the global semantic structure of the original video. The reconstruction loss is measured by the Euclidean distance between the original and reproduced video feature and participates in the training of the model as follows:

$$\mathcal{L} = \sum_{i=1}^{N} \Big(\underbrace{-\log P\left(\mathbf{S}^i | \mathbf{V}^i\right)}_{\text{encoder-decoder}} + \lambda \underbrace{\mathcal{L}_{rec}(\mathbf{V}^i, \mathbf{Z}^i)}_{\text{reconstructor}} \Big). \tag{3}$$

where $\mathbf{S} = \{s_1, s_2, ...s_m\}$ is sentence description, $\mathbf{Z} = \{z_1, z_2, ...z_m\}$ is hidden state sequence of reconstructor, and λ is a trade-off parameter for controlling the influence of reconstuctor.

REINFORCE Algorithm. Inspired by [2], we employed the REINFORCE algorithm with a baseline [5] and set the reward obtained by the current model under the inference algorithm used at validation time as the baseline, which called self critical. We use the REINFORCE to directly optimize the non-differentiable metrics, rather than the cross entropy loss. Moreover, the REINFORCE algorithm can help address the exposure bias [1] problem.

2. Experimental Results

2.1. Features

Besides the C3D features provided by the challenge organizers, we extract i3d features fine-tuned on Kinetics and ActivityNet, p3d features, inception-V4 features, resnet-152 features, and inception_resnet_v2 features for static images of ActivityNet dataset. These features can be used individually or fused together. Two fusion strategies are designed.

- **Simple Concatenation.** We simply concatenate the features together.
- **Dimensionality Reduction Concatenation.** We first reduce the dimension by full-connection layers and then concatenate them together.

2.2. Training

We set the hidden size of all LSTMs as 512, except for the reconstructor as same dimension as that of original features. We set the maximum sentence length for predicted sentence is 82, which is same as that in the training dataset.

We employ early stop on METEOR score during training all models. The entire training process is as follows:

- Firstly, we rely on the forward likelihood to train the encoder-decoder component by optimizing cross entropy loss, and got the model with best METEOR on the validation dataset.
- Secondly, the reconstructor and the backward reconstruction loss $\mathcal{L}_{rec}$ are introduced. We use the whole loss defined in Eq. (3) to jointly train the reconstructor and fine-tune the encoder-decoder. Model with best METEOR score can be received.
- Afterwards, use the REINFORCE algorithm to train the model from second step by directly optimizing the METEOR score.

2.3. Performance Evaluation

We first show the performance of original TA, TA with temporal encoder (TA_enc), TA with reconstructor (TA_rec) and TA with REINFORCE (TA_sc) on validation with ground-truth proposals in Table 1. It can be observed that

Table 1. Performance evaluation of models on the validation dataset with c3d feature in terms of METEOR scores (%).

Model	METEOR(with gt proposals)
TA	8.5197
TA_enc	8.7944
TA_rec	8.9328
TA_sc	13.0286

TA_enc outperforms TA, indicating that temporal encoding is useful to improve the captioning performance. TA_rec is trained based on TA_enc. With the backward flow information captured by reconstructor, the captioning performance can be further improved. Moreover, the performance can be significantly improved with the REINFORCE algorithm.

Table 2. Performance evaluation of models trained by REINFORCE on the validation dataset in terms of METEOR scores (%). Different features and fusion strategies are used. The 'i3ds' represents i3drgb and i3dflow, the '(unt)' means feature extracted from models trained with Imagenet and fine-tuned on Kinetics and '(t)' means features extracted from models fine-tuned on ActivityNed data.

num	Feature Type	METEOR(gt)	METEOR(ours)
1	c3d	13.0286	8.6508
2	p3d	12.8433	8.5878
3	i3d rgb (unt)	13.4688	9.1515
4	i3d flow (unt)	13.2782	8.9731
5	i3d rgb (t)	13.2756	9.0224
6	i3d flow (t)	13.2371	8.9721
7	inceptionV4	13.1697	8.8144
8	resnet152	13.0792	8.5952
9	inception_resnet_v2	12.9856	8.8193
10	i3ds(unt)	13.4765	9.0347
11	i3ds(unt) + c3d	12.9822	8.8571
12	i3ds (unt) + p3d	12.5905	8.4968
13	i3ds(t)	12.7720	8.966
14	i3ds (t) + i3ds (unt)	13.1894	8.9543
15	i3ds (t) + i3ds (unt)+ p3d	12.6332	8.5739
16	i3drgb (t) + i3drgb (unt)	12.9484	8.7323
17	i3dflow (t) + i3dflow (unt)	12.9702	8.7316
18	i3ds (unt)	13.0941	8.7802
19	i3ds (t)	12.7720	8.8857
20	i3ds (t) + i3ds (unt)	13.1895	8.6158
21	i3ds (t) + i3ds (unt) + p3d	12.6332	8.6305

Table 2 shows METEOR of models trained with different features and REINFORCE algorithm. Models 1 to 9 are trained with single features. And models 8 to 17 are trained with features by the simple concatenation strategy. Models 18 to 21 are trained with features by the dimensionality reduction concatenation strategy. The results are performed with the ground-truth proposal (gt) as well as the proposals we get from the Temporal Action Proposals task of ActivityNet2018 Challenge. We find that performances of models trained by fused features are slightly better than those of models trained with single features.

Finally, we fuse the models the METEOR scores of which on validation dataset with our proposal larger than 8.8% together. Table 3 shows that ensemble model presents a significant better performance. After generating the sen-

Table 3. Performance evaluation of model ensemble on the validation dataset in terms of METEOR scores (%).

model Type	METEOR(our proposals)
model_ensemble	9.3005
model_ensemble(reranking)	9.4002

tence, we compute a ranking score [4] for the proposal-sentence pair as follows:

$$r = \lambda_{sent} * con_i + \lambda_{prop} * p(proposal_i) \quad (4)$$

For the i_{th} proposal, its proposal confidence score $p(proposal_i)$ and the confidence score con_i of its associated sentence is generated. We obtain the ranking score r by summing con_i and $p(proposal_i)$ with the trade-off parameters λ_{sent} and λ_{prop}, and re-rank the results. The ranking process can help to remove the results with good sentence but bad proposal or good proposal but bad sentence.

2.4. Submission Results

We submit the captions predicted by the model_ensemble(reranking) with temporal proposals. The METEOR score on the test server is 8.1057.

3. Conclusion

In this report, we employ a temporal encoder and a reconstructor equipped with the TA model for video captioning. The REINFORCE algorithm is utilized to train our video captioning model with the optimization on the METEOR score. During the inference, the ensemble and joint ranking techniques are used. The METEOR score achieves 8.1057 on the test server for the dense-captioning events in videos challenge.

References

[1] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

[2] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. In *CVPR*, volume 1, page 3, 2017.

[3] B. Wang, L. Ma, W. Zhang, and W. Liu. Reconstruction network for video captioning. *arXiv preprint arXiv:1803.11438*, 2018.

[4] J. Wang, W. Jiang, L. Ma, W. Liu, and Y. Xu. Bidirectional attentive fusion with context gating for dense video captioning. *arXiv preprint arXiv:1804.00100*, 2018.

[5] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer, 1992.

[6] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *Computer Vision (ICCV), 2015 IEEE International Conference on*. IEEE, 2015.

[7] L. Yu, M. Bansal, and T. L. Berg. Hierarchically-attentive rnn for album summarization and storytelling. *arXiv preprint arXiv:1708.02977*, 2017.

RUC+CMU: System Report for Dense Captioning Events in Videos

Shizhe Chen[1], Yuqing Song[1], Yida Zhao[1], Jiarong Qiu[1], Qin Jin[1*] and Alexander Hauptmann[2]
[1] Renmin University of China, [2] Carnegie Mellon University
{cszhe1, syuqing, zyiday, jiarong_qiu, qjin}@ruc.edu.cn, alex@cs.cmu.edu

Abstract

This notebook paper presents our system in the ActivityNet Dense Captioning in Video task (task 3). Temporal proposal generation and caption generation are both important to the dense captioning task. Therefore, we propose a proposal ranking model to employ a set of effective feature representations for proposal generation, and ensemble a series of caption models enhanced with context information to generate captions robustly on predicted proposals. Our approach achieves the state-of-the-art performance on the dense video captioning task with 8.529 METEOR score on the challenge testing set.

1. Task Introduction

Most natural videos contain multiple events. Instead of generating a single sentence to describe the overall video content, the dense video captioning task aims to localize the event and generate a series of sentence to describe each event. This task is more challenging than the single sentence video captioning task, which requires to generate good temporal event proposals, consider the correlations of different events in the video and so on.

2. Proposed Approach

The framework of our approach is presented in Figure 1, which consists of four components: 1) segment feature extraction; 2) proposal generation; 2) caption generation; and 4) re-ranking. In this section, we introduce each component of the framework in details.

2.1. Segment Feature Extraction

We divide the video clip into non-overlapping segments and extract features for each segment. The length of the segment is set to be 64 frames in our work. Since the video contains multi-modal information, we first extract three types of deep features from different modalities, which are: 1) image modality: Resnet features [3] pretrained on the ImageNet dataset; 2) motion modality: I3D features [1] pretrained on the Kinetics dataset; and 3) audio modality: VGGish features [4] pretrained on the Youtube8M dataset.

As shown in previous works [6], the context information plays an important role in generating proper captions for an event proposal. Therefore, we utilize a bidirectional LSTM to capture the context information and extract the hidden states of LSTM as our context feature. The LSTM employs the aforementioned three types of deep features as input, and is trained to predict concepts in groundtruth captions in each step. In such a way, the LSTM learns the bidirectional context for each segment to generate captions.

After the feature extraction, the video is represented as a sequence of segment-level features.

2.2. Proposal Generation

We adopt a two-stage pipeline to generate temporal proposals. Firstly, a heuristic sliding window method is exploited to generate a series of candidate proposals for each video. Then, we train a proposal ranking model to select proposals that are of high tiou (temporal intersection over union) with groundtruth proposals.

Candidate Proposal Generation

In order to generate candidate proposals with high recalls, we apply the sliding window approach on the video clip. Assuming w is the length of the window, we slide the window over the clip with the shift of $w/4$. The window lengths are generated according to lthe length distribution of groudtruth proposals and the length of the video. We first cluster proportions of the groundtruth proposal in the video into K centers $\{w_1^p, \cdots, w_K^p\}$. Then we set the window lengths for each video to be $w_k = w_k^p \cdot l$ for $k = 1, \cdots, K$, where l is the length of the video.

Proposal Ranking Model

The proposal ranking model is trained to filter out inappropriate candidate proposals. We consider a good temporal proposal to satisfy the following conditions: 1) the event in the proposal is meaningful; 2) the event in the proposal is different from its context; 3) the boundaries of the proposal contain variance; and 4) the location of the proposal is satisfied with groundtruth distributions. Therefore, we propose

*Corresponding author.

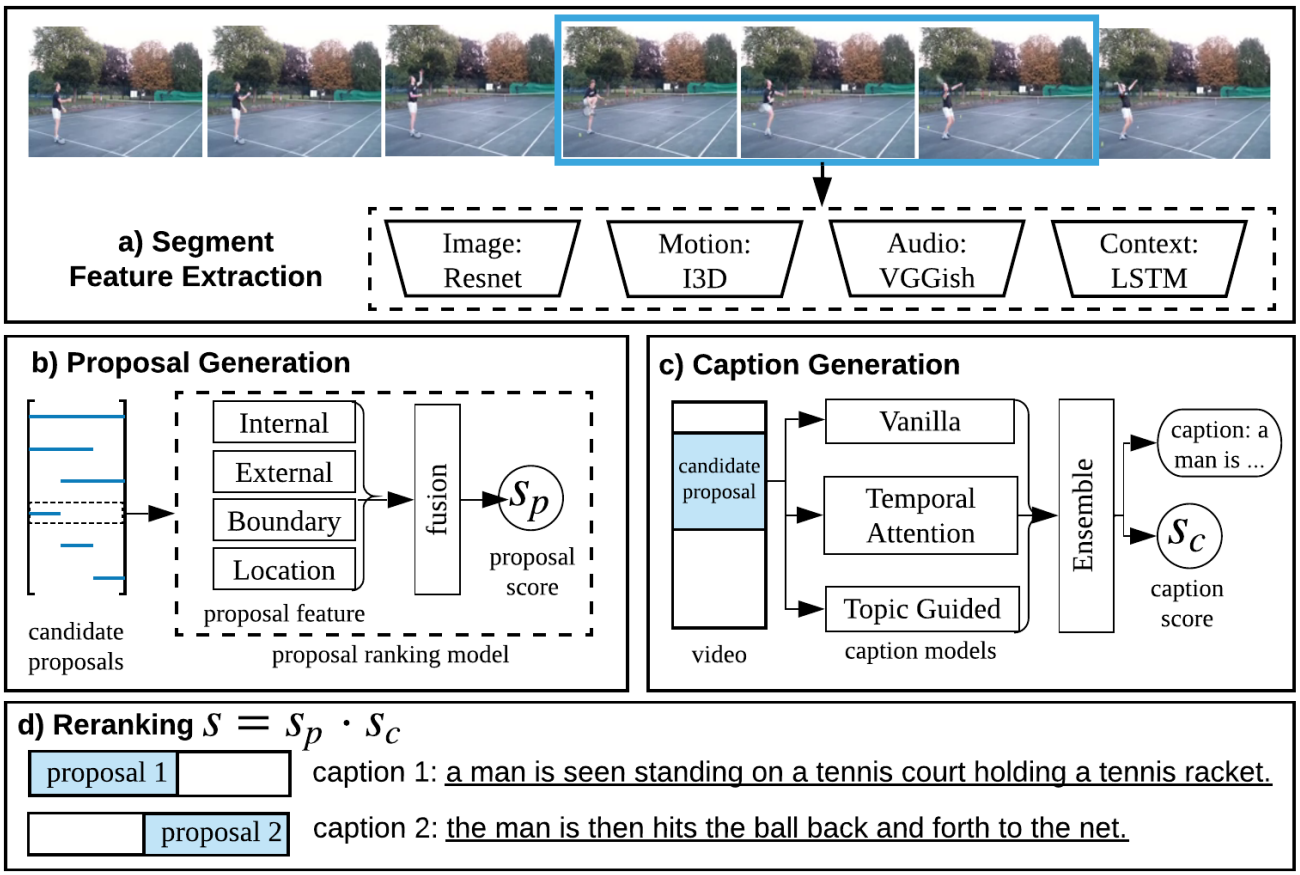


Figure 1. Framework of our proposal approach, which consists of four components: 1) segment feature extraction to transfer the video into a sequence of multimodal features; 2) proposal generation which contains a proposal ranking model to select good event proposals; 3) caption generation which employs various caption models to generate accurate event descriptions; and 4) re-ranking to select event captions with both high proposal and caption score.

four different features to satisfy the conditions above: 1) internal feature: mean pooling of segment features in the proposal to represent events in the proposal; 2) external feature: mean pooling of segment features in the context to represent contextual events; 3) boundary feature: the difference of the segment feature near the proposal boundary to represent the boundary variance; and 4) the proportion of the location and duration of the proposal. We utilize a two-layers feed-forward neural network to fuse these features and predict the proposal score s_p of the proposal. During training, candidate proposals with tiou above 0.7 are as positive samples and tiou less than 0.5 are as negative samples.

2.3. Caption Generation

In order to generate accurate and diverse video captions, we employ three different caption models and ensemble them to generate the caption for each event proposal.

Vanilla Caption Model [5] is the baseline model for the video captioning task. It consists of a multimodal video encoder and a LSTM language decoder. Since the context is vital to generate consistent captions for the proposal, we enhance the encoder with the LSTM context features [8].

Temporal Attention Caption Model [9] improves over the vanilla caption model via paying attention to relevant segments in the video to generate each word. To incorporate the context, we also enhance the encoder in the attention model with the contexts of the boundaries.

Topic Guided Caption Model [2] utilizes the video topics to guide the caption model to generate topic-aware captions. Since there are 200 manual labeled categories in the ActivityNet dataset, we directly use these categories as our topics. We train a topic predictor which is a single-layer feed-forward neural network to predict the category probabilities of each proposal. As the size of the dataset is not large, we adopt the Topic Concatenation in Decoder version in [2] to guide the caption generation which requires fewer parameters than TGM in [2].

We firstly use the cross entropy loss to pretrain all

the caption models, which optimizes the likelihood of the groundtruth captions. But such training approach suffers from the exposure bias and evaluation mismatch problems. Therefore, we employ the self-critical reinforcement learning [7] to further train our caption models, which is the state-of-the-art approach in image captioning and alleviates the above two problems. CIDEr and METEOR are weighted as our reinforcement reward.

Caption Model Ensemble aims to make use of various caption models. We ensemble the word prediction of each model at every step. Beam search with beam size of 5 is used to generate the final caption with probability score s_c.

2.4. Re-ranking

Since both the proposal quality and the caption quality influence the evaluation of dense captions, we re-rank the captions of different proposals by $s = s_p \cdot s_c$. The top 10 captions with their proposal are selected.

3. Experimental Results

3.1. Experimental Settings

Dataset: The ActivityNet Dense Caption dataset [6] is used in our work. We follow the official split with 10,009 videos for training, 4,917 videos for validation and the remaining 5,044 videos for testing. The groundtruth of the testing videos are unknown. For the final submission, we enlarge our training set with part of validation set to future improve the performance, which contains 14,009 videos for training and 917 videos for validation.

Evaluation Metrics: We employ the precision and recalls to evaluate the performance of proposals. To evaluate the captions, we first evaluate the performance of the caption using the groundtruth proposal. And then we use the same metric as [6] to evaluate the captions of predicted proposals, which computes the caption performance for proposals with tiou 0.3, 0.5 and 0.7 with the groundtruth.

3.2. Evaluation of Proposals

Table 1 presents the performance of our proposal generation approach. For the sliding window candidate proposal generation, we use 20 clusters to generate sliding window, which leads to 241 proposals for each video. We can see that the heuristic sliding window approach achieves remarkable recall (0.98 on average), while the precision of the proposal is quite low. After applying the proposal ranking model, we select proposals that contain proposal score $s_p > 0.5$ which results in 53 proposals on average for each video. The precision is significantly improved (0.71 vs 0.28) with minor recall decrease, which demonstrates the effectiveness of our proposal ranking model.

Table 1. Performance of the proposal generation approach. P and R are short for precision and recall.

	#props	metric	0.3	0.5	0.7	avg
sliding window	241	P	0.45	0.27	0.12	0.28
		R	0.99	0.99	0.95	0.98
proposal ranking	53	P	0.97	0.77	0.38	0.71
		R	0.91	0.85	0.76	0.84

Table 2. Performance of difference caption models.

proposal	model	Bleu4	Meteor	CIDEr
groundtruth	vanilla	3.62	13.37	52.36
	attention	3.69	13.21	53.45
	topic guided	3.46	13.71	51.53
	ensemble	**3.97**	**13.75**	**56.45**
predicted	ensemble	4.00	12.44	31.10

Table 3. Performance of the submitted models.

	Bleu4	Meteor	CIDEr
val_small	3.92	12.67	31.92
testing	-	8.529	-

3.3. Evaluation of Captions

Table 2 shows the caption performance using groundtruth proposals. We can see that the performance of different models are competitive with each other, and the ensemble of these models achieves the best performance consistently on different caption metrics. For the predicted proposals, the performance is dropped a little due to the imperfect proposal, which shows the robustness of our caption model on imperfect proposals. The significant decrease of CIDEr score mainly results from the more proposals in the predicted version than the groundtruth, which makes the tf-idf statistics different.

3.4. Submission

For the final submission, we train our caption models on the bigger training set and utilize the smaller validation set to select models. The performance of the submitted model is presented in Table 3. More training data brings small improvement, and our model achieves 8.529 METEOR score on the testing set.

4. Conclusion

In this work, we propose a system with four components to generate dense captions in videos, which achieves significant improvements on the dense video captioning task. Our results show that it is important to utilize context-related features for both the proposal generation and caption generation. In the future, we will explore to unify the system in an end-to-end way to improve the proposal module with captions and generate more diverse caption for the events.

References

[1] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733. IEEE, 2017.

[2] S. Chen, J. Chen, and Q. Jin. Generating video descriptions with topic guidance. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pages 5–13. ACM, 2017.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, et al. Cnn architectures for large-scale audio classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 131–135. IEEE, 2017.

[5] Q. Jin, S. Chen, J. Chen, and A. Hauptmann. Knowing yourself: Improving video caption via in-depth recap. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 1906–1911. ACM, 2017.

[6] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 1, page 6, 2017.

[7] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. In *CVPR*, volume 1, page 3, 2017.

[8] J. Wang, W. Jiang, L. Ma, W. Liu, and Y. Xu. Bidirectional attentive fusion with context gating for dense video captioning. *arXiv preprint arXiv:1804.00100*, 2018.

[9] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE international conference on computer vision*, pages 4507–4515, 2015.

TDU&AIST Submission for ActivityNet Challenge 2018 in Video Caption Task

Tenga Wakamiya[1]* Takumu Ikeya[1]* Akio Nakamura[1] Kensho Hara[2] Hirokatsu Kataoka[2]
Tokyo Denki University[1]
National Institute of Advanced Industrial Science and Technology (AIST)[2]
{wakamiya.t, ikeya.t}@is.fr.dendai.ac.jp, nkmr-a@cck.dendai.ac.jp
{kensho.hara, hirokatsu.kataoka}@aist.go.jp

Abstract

In the report, we introduce our video caption approach for the ActivityNet Challenge in conjunction with CVPR 2018. Based on the 3D-ResNet with 34-layer [1, 2] and LSTM-based Sentence Generator [5], our captioner generates a suitable sentence along an input video. The captioning model is trained with the training-set on ActivityNet database. In the experimental section, we show our rate on the test-set with evaluation server. Finally, we achieved to put our name on the leaderboard![1]

1. Introduction

The task of finding a de facto standard for video recognition has advanced with both hand-crafted and deeply learned feature representations. In the recent DNN-based video recognition, we are focusing on 3D convolutional networks such as C3D [4] and 3D-ResNets [1, 2].

On one hand, video caption which includes time duration seems to be very difficult issue in the current vision-based algorithm. The open problem is composed by two problem, namely (i) video representation in order to generate an appropriate sentence, and (ii) temporal segmentation to fix an event duration. We believe that the video representation problem is more important. Therefore we apply a sophisticated video representation 3D-ResNet with layer-34 for video caption. To generate a video caption, we apply a standard sentence generator LSTM based on the Google's Show and Tell algorithm [5].

*denotes equal contribution

[1]Two bachelor students have tried very challenging task, namely "can CV-research beginners achieve the ActivityNet Challenge in two months?" Although our rate is far from competitive performance, we succeeded to list our team on the leaderboard.

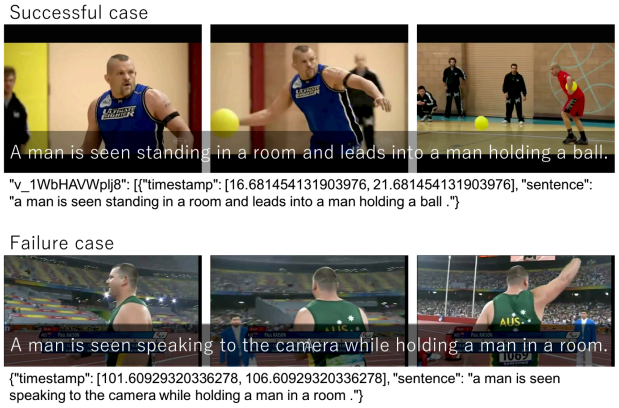


Figure 1. The successful (top) and failure (bottom) cases in ActivityNet Challenge with video caption.

2. Proposed approach: 3D-ResNet-34 + LSTM

We simply combine 3D-ResNet-34 with LSTM. To train/test the LSTM, the layer after global average pooling (2,048-d vector) is inserted from 3D-ResNet. The 3D-ResNet-34 is pretrained by Kinetics dataset [6] and the 3D-ResNet-34+LSTM is trained by ActivityNet caption [3] with end-to-end training manner.

3. Result on video caption task

Our performance value with METEOR is 0.6266. The score is far from top-ranked captioners from other teams. The result is coming from fewer proposals per a video. Our temporal proposals with start- and end-time are only 2 per a video. In the future, we would like to evaluate the video captioner with e.g. over 100 proposals in video. Moreover, we must implement an improved temporal proposals and more sophisticated models such as 3D-ResNet-{50, 101, 152}, 3D-ResNeXt-101.

References

[1] K. Hara, H. Kataoka, and Y. Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. International Conference on Computer Vision Workshop (ICCVW), 2017. 1

[2] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet? IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018. 1

[3] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles. Dense-captioning events in videos. In *International Conference on Computer Vision (ICCV)*, 2017. 1

[4] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. International Conference on Computer Vision (ICCV), 2015. 1

[5] Vinyals, O. and Toshev, A. and Bengio, S. and Erhan, D. Show and tell: A neural image caption generator. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. 1

[6] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman and Andrew Zisserman. The kinetics human action video dataset. arXiv:1705.06950, 2017. 1

Cascade Convolutional Attention Networks for Action Recognition-Task A:Trimmed Activity Recognition (Kinetics)

Zhikang Fu, Lei Zhao, Xiao Liu
Challenger X team, Cloud Vision Dept., Meitu

1. Introduction

We present the design of attention blocks for action recognition on convolutional networks and propose a cascade convolutional attention network for action recognition.

The center loss is adopted for learning a center for deep features of each class and penalizing the distances between the deep features and their corresponding class centers.

We implement a multi-stream framework to utilize the rich multi-modal information in videos for human action recognition. Specifically, we train four convolutional neural networks whose inputs are RGB images, stacked optical flow, human pose information and audio respectively in each video.

2. Proposed Approach

2.1. Multimodal Feature Extraction

A video can be decomposed into visual and acoustic components. The visual component can be further divided into spatial and temporal parts. The optical flow are extracted with TVL1 optical flow algorithm[4]. The human pose information is calculated by Openpose [1].

The audio is divided into 2s frames. The MFCC features are extracted. The resulting frame can be seen as a 348 X12 image that form the input of a Inception-ResNet-v2 image classification model.

2.2. Attention Block

In fields of action recognition, discriminative spatial temporal features are key factors to define the action in a video. Thus, we aim to design attention blocks that can capture salient data both spatially in each frame and temporally across frames.

In implementation, the proposed attention block can be formulated as follows. Let $\boldsymbol{S} \in R^{C\times H\times W}$ denote the input feature maps, where C is the number of feature channels, and H and W represent the filter map size. The feature vector $s_{i,j} \in R^C$ consists of the element at spatial location (i, j) of each feature map input.

As shown in Figure 1, the convolutional layers in the attention block produce a feature map, with each element $\boldsymbol{s}^{'}_{i,j}$ at location (i, j) calculated as in Equation (1).

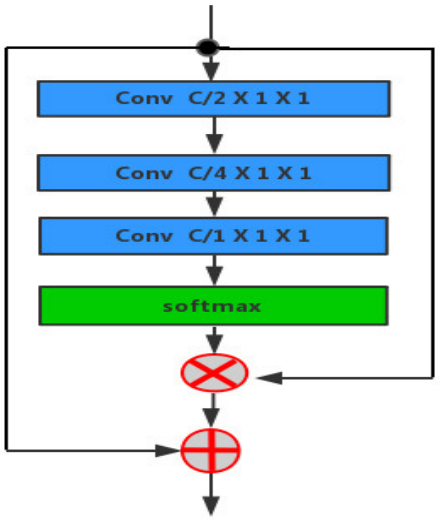


Figure 1: The architecture of attention block.

$$s^{'}_{i,j} = convs(s_{i,j}; \boldsymbol{W}_c), \tag{1}$$

where $convs(.)$ represents a series of convolutional operations that calculates the convoluted feature maps on the input and $\boldsymbol{W}_c$ represents the weight parameters to be learned.

The attention weight map $\boldsymbol{\alpha}$ is defined as $\boldsymbol{\alpha} = \{\alpha_{i,j}\}$, where $\alpha_{i,j}$ is then produced with Equation (2).

$$\alpha_{i,j} = softmax(s^{'}_{i,j}), \tag{2}$$

The re-weighted feature map $\boldsymbol{s}^{\alpha} = \{s^{\alpha}_{i,j}\}$ is computed by element-wise product of the attention map and the input feature maps S. The final output feature map $\boldsymbol{S}^{\alpha}$ is produced by adding a shortcut connection from the input feature map $\boldsymbol{S}$, as shown in Equation (3) and (4),

$$s^{\alpha}_{i,j} = \alpha_{i,j} \odot s_{i,j} \tag{3}$$

$$\boldsymbol{S}^{\alpha} = tile(s^{\alpha}) + \boldsymbol{S} \tag{4}$$

where $\odot$ and$+$ respectively denote element-wise product and add operations.The *tile* operation duplicates the attention map to produce C feature maps.

The re-weighted feature maps will be fed into subsequent classification layer. During training, the final action classification loss is back propagated to the attention block to guide the learning of block weights $\boldsymbol{W}_c$ so that the attention map $\boldsymbol{\alpha}$ can pick up salient regions.

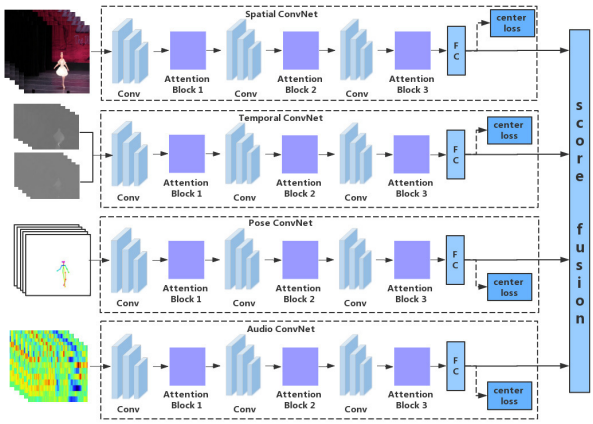


Figure 2: The architecture of our proposed cascade attention network on a four-stream ConvNet.

2.3. Cascade Attention Networks

Our four-stream model is constructed by four individual spatial, temporal, pose and audio stream. Each stream models different type of information in videos respectively and independently. Cascade attention blocks on a four stream ConvNet where attention blocks are embedded between convolutional layers progressively from low layers to high layers. Figure 2 illustrates the Cascade Convolutional Attention Networks.

Besides, the joint supervision of softmax loss and center loss [3] to train the CNNs for discriminative feature learning. formulation is given in Equation (5), (6), (7).

$$l = l_s + \alpha l_c \tag{5}$$

$$l_s = -\sum_{i=1}^{m} \log \frac{e^{W_{yi}^T x_i + b_{yi}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} \tag{6}$$

$$l_c = \frac{1}{2} \sum_{i=1}^{m} \| x_i - c_{y_j} \|_2^2 \tag{7}$$

where x_i denotes the i th deep feature,belonging to the y_ith class, The c_{y_i} denotes the y_ith class center of deep features.

Last, Grid Search algorithm is employed to search the optimal weights of RGB, optical flow, pose and audio networks. Last result is calculated by four weighted results.

3. Experiment Results

Except for using multiple feature models, we tried different ConvNet architectures such as Inception-BN, Inception-v3, Inception-ResNet-V2, ResNext-50 and SE-ResNext to train the RGB, optical flow, skeleton and audio separately, and then we ensembled all models to get the final result. All the RGB models and audio models are initialized with weights pretrained on ImageNet. In order to utilize the ImageNet RGB models to initialize the motion and pose model, we use cross-modality pretraining method [2] that averages the weights across RGB channels and replicates this average by the channel number of the target network.Table 1 summarizes our results on the Kinetics validation dataset.

Model	Top-1 Accuracy(%)	Top-5 Accuracy(%)
RGB	77.5%	90.5%
Flow	54.4%	75.6%
Audio	21.3%	38.7%
pose	45.6%	58.6%
Ensemble	80.5%	92.3%

Table 1: Kinetics validation results.

References

[1] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[2] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36. Springer, 2016.

[3] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pages 499–515. Springer, 2016.

[4] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l1 optical flow. *Lecture Notes in Computer Science*, 4713(5):214–223, 2007.

Qiniu Submission to ActivityNet Challenge 2018

Zhang Xiaoteng, Bao Yixin, Zhang Feiyun, Hu Kai, Wang Yicheng,
Zhu Liang, He Qinzhu, Lin Yining, Shao Jie and Peng Yao
Qiniu AtLab
Shanghai, China
shaojie@qiniu.com

Abstract

In this paper, we introduce our submissions for the tasks of trimmed activity recognition (Kinetics)[8] and trimmed event recognition (Moments in Time)[9] for Activitynet Challenge 2018. In the two tasks, non-local neural networks and temporal segment networks are implemented as our base models. Multi-modal cues such as RGB image, optical flow and acoustic signal have also been used in our method. We also propose new non-local-based models for further improvement on the recognition accuracy. The final submissions after ensembling the models achieve 83.5% top-1 accuracy and 96.8% top-5 accuracy on the Kinetics validation sets, 35.81% top-1 accuracy and 62.59% top-5 accuracy on the MIT validation sets.

1. Introduction

Activity Recognition in videos has drawn increasing attention from the research community in recent years. The state-of-the-art benchmark datasets such as ActivityNet, Kinetics, Moments in Times have contributed to the progress in video understanding.

In Activitynet Challenge 2018, we mainly focused on two trimmed video recognition tasks based on Kinetics and Moments in Times datasets respectively. The Kinetic dataset consists of approximately 500,000 video clips, and covers 600 human action classes. Each clip lasts around 10 seconds and is labeled with a single class. Similarly, the Moments in Times dataset is also a trimmed dataset, including a collection of 339 classes of one million labeled 3 second videos. The videos not only involve people, but also describe animals, objects or natural phenomena, which are more complex and ambiguous than the videos in Kinetics.

To recognize actions and events in videos, recent approaches based on deep convolution neural networks have achieved state-of-the-art results. To address the challenge, our solution follows the strategy of non-local neural network and temporal segment network. Particularly, we learn models with multi-modality information of the videos, including RGB, optical flow and audio. We find that these models are complementary with each other. Our final result is an ensemble of these models, and achieves 83.5% top-1 accuracy and 96.8% top-5 accuracy on the Kinetics validation set, 35.81% top-1 accuracy and 62.59% top-5 accuracy on the MIT validation sets.

2. Our Methods

2.1. Temporal Segment Networks

One of our base model is temporal segment network (TSN)[11]. TSN models long-term temporal information by evenly sampling fixed number of clips from the entire videos. Each sampled clips contain one or several frames / flow stacks, and produce the prediction separately. The video-level prediction is given by the averaged softmax scores of all clips.

We experiment with several state-of-the-art network architectures, such as ResNet, ResNeXt, Inception[10], Inception-ResNet, SENet[7], DPN[2]. These models are pretrained on ImageNet, and have good initial weights for further training. Table 1 and 2 show our TSN results on Kinetics and Moments in Times dataset.

Models	Top-1 acc(RGB)	Top-1 acc(Flow)
DPN107	75.95	69.60
ResNext101	75.43	None
SE-ResNet152	73.88	None
InceptionV4	73.51	68.76
ResNet152	72.04	67.13
InceptionV3	68.52	64.08

Table 1. Performance of TSN on Kinetics

2.2. Acoustic Model

While most motions can be recognized from visual information, sound contains information in another dimen-

Models	Top-1 acc(RGB)	Top-1 acc(Flow)
DPN107	31.06	None
ResNet152	30.21	None
ResNet269	None	18.53*
ResNet101	None	22.82

Table 2. Performance of TSN on Moments in Times dataset. *: Due to time limit, the training of these models was not finished.

sion. We use audio channels as complementary information to visual information to recognize certain classes, especially for the actions with better distinguishability on sound, whistling and barking for example.

We use the raw audio as input into the pre-trained VGGish model[6][4], and extract $n \times 128$ dimension features to do classification (n is the length of audio). Besides, we extract MFCC features from raw audio and train with SE-ResNet-50 (Squeeze-and-Excitation Network) and ResNet-50 (Deep residual network[5]). After ensembling with visual models, we achieve 0.7% improvement in top 1 error rate.

Table 3 shows our acoustic results on on Kinetics and Moments in Times dataset. Figure 1 shows 15 classes with best top 1 accuracy in MIT validation dataset and2 a shows 25 classes with best top 1 accuracy in Kinetics validation dataset.

Models	Kinetics	MIT
MFCC+ SENet-50	7.73	16.8
VGGish	7.83	17.12
Audio Ensemble	8.83	19.02

Table 3. Performance of acoustic models on Kinetics and MIT dataset.

2.3. Non-local Neural Networks

Non-local Neural Networks[12] extract long-term temporal information which have demonstrated the significance of non-local modeling for the tasks of video classification, object detection and so on.

Notation Image data and feature map data are generally three-dimensional: channel, height and width (in practice there is one more dimension: batch). They are represented as C-dimensional vectors with 2-dimensional index

$$\boldsymbol{X} = \{\boldsymbol{x}_i | i = (h, w) \in \mathbb{D}^2, \boldsymbol{x}_i \in \mathbb{R}^C\} \tag{1}$$

where $\mathbb{D}^2 = \{1, 2, \cdots, H\} \times \{1, 2, \cdots, W\}$. Video data get one more dimension time and are represented as C-dimensional vectors with 3-dimensional index

$$\boldsymbol{X} = \{\boldsymbol{x}_i | i = (t, h, w) \in \mathbb{D}^3, \boldsymbol{x}_i \in \mathbb{R}^C\} \tag{2}$$

where $\mathbb{D}^3 = \{1, 2, \cdots, T\} \times \mathbb{D}^2$.

Non-local Operation Non-local operation define a generic non-local operation in deep neural networks as:

$$\boldsymbol{y}_i = \sum_{j \in \mathbb{D}^3} f(\boldsymbol{x}_i, \boldsymbol{x}_j) g(\boldsymbol{x}_j). \tag{3}$$

Here function f representing the relation between position i and j. Many visions of function f such as $f(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left[\theta(\boldsymbol{x}_i)^T \varphi(\boldsymbol{x}_j)\right]$ are discussed, but performed almost the same.

As done in[1][3], a 2D $k \times k$ kernel can be inflated as a 3D $t \times k \times k$ kernel that spans t frames, in our experiments we used 32 frames. So this kernel can be initialized from 2D models(pretrained on Imagenet), each of the t planes in the $t \times k \times k$ kernel is initizlized by pretrained $k \times k$ weights, rescaled by 1/t. Each video we sample 64 consecutive frames from the original full-length video and then dropping every other frame. The non-local operation computes the response at a position as a weighted sum of the features at all positions with Embedding Gaussian. We used 5 non-local blocks added to i3d baseline. Table 4 shows our non-local results on Kinetics and Moments in Times dataset.

Models	Kinetics	MIT
Res50 baseline	78.63	30.83
Res50 non-local	80.80	32.96
Res101 baseline	79.58	31.33
Res101 non-local	81.96	33.69

Table 4. Performance of nonlocal NN on Kinetics and MIT dataset.

3. Relation-driven Models

We are interested in two questions. Firstly, *non-local* operations would be important for relation learning, but *global* operations may be unnecessary. If position i is far away from j, then $f(\boldsymbol{x}_i, \boldsymbol{x}_j) \approx 0$. Second quesion is that an unsupervised function may not be able to learn relations.

Mask Non-local To answer the first question, we compared the performance of non-local opeations and mask non-local opeations:

$$\boldsymbol{y}_i = \sum_{j \in \mathbb{D}^3} \mathbb{I}_{\mathbb{D}_i}(j) f(\boldsymbol{x}_i, \boldsymbol{x}_j) g(\boldsymbol{x}_j). \tag{4}$$

Here $\mathbb{D}_i$ is the $\delta-$ neighbourhood of $i = (t_i, h_i, w_i)$. Say:

$$\mathbb{D}_i = [t_i - \delta_t, t_i + \delta_t] \times [h_i - \delta_h, h_i + \delta_h] \times [w_i - \delta_w, w_i + \delta_w]. \tag{5}$$

$\mathbb{I}_{\mathbb{D}_i}(j)$ is the mask function. Say:

$$\mathbb{I}_A(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}. \tag{6}$$

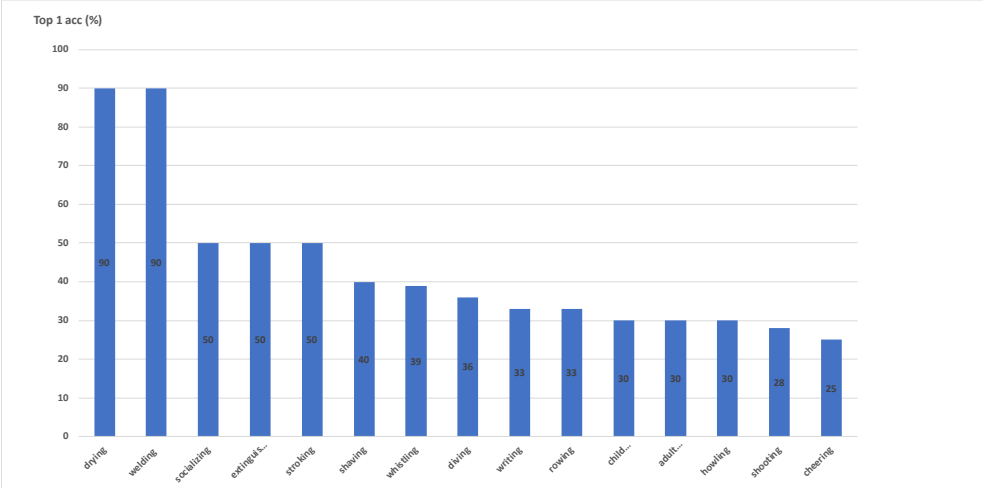


Figure 1. Acoustic Model: 15 classes with best top 1 accuracy in MIT validation dataset

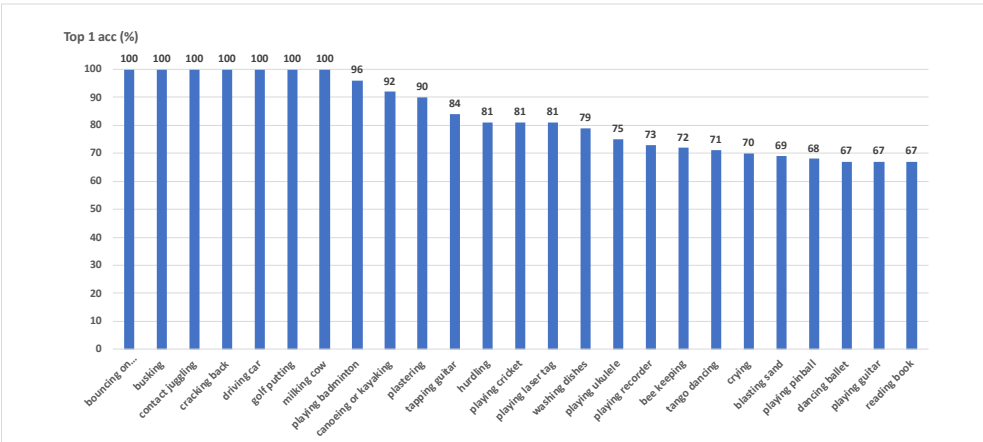


Figure 2. Acoustic Model: 25 classes with best top 1 accuracy in Kinetics validation dataset.

δ_t	δ_h	δ_w	top-1 acc
$+\infty$	$+\infty$	$+\infty$	80.80
$+\infty$	$\frac{3}{7}H$	$\frac{3}{7}W$	81.26
$+\infty$	$\frac{3}{28}H$	$\frac{3}{28}W$	80.63
$\frac{1}{2}T$	$\frac{3}{7}H$	$\frac{3}{7}W$	81.65
$\frac{1}{2}T$	$\frac{3}{28}H$	$\frac{3}{28}W$	80.93

Table 5. Perfomance for different settings of δ neighbourhood.

Table 5 shows mask nonlocal's performance on Kinetics. $+\infty$ means non-local operation in the dimention. Note that the first setting is the non-local baseline.

Learning Relations in Video Common convolution layers use invariant kernels for feature extraction at all positions in the feature map. It's limited for learning relations between different positions on the feature map. Nonlocal operations compute a feature-map-wise relation matrix to represent the kernel so that different positions get different but related feature extractions. The problem is that an unsupervised function may not be able for relations learning. We proposed a new model to learn the relation patten.

The network contains a network-in-network with a $(2t_0+1)\times(2h_0+1)\times(2w_0+1)$ size receptive field. The network-in-network computes a $(2t_1+1)\times(2h_1+1)\times(2w_1+1)$-dimensional relation vector $r^{(i)}$ for any position $i=(t,h,w)$ at the feature map ($t_1<t_0<\frac{1}{2}T$, $h_1<h_0<\frac{1}{2}H$ and $w_1<w_0<\frac{1}{2}W$).

The learnable relation vector $r^{(i)}$ represent the relation between position i and its neighbourhood

$$\begin{aligned}\mathbb{D}_i &= [t_i \pm t_1]\times[h_i \pm h_1]\times[w_i \pm w_1],\\ \boldsymbol{y}_i &= \sum_{j\in\mathbb{D}_i} r_j^{(i)} g(\boldsymbol{x}_j).\end{aligned} \tag{7}$$

Here $r_j^{(i)}$ is the j^{th} element of $r^{(i)}$. Figure 3 shows our network structure. Note that, by using mask non-local's initialization, our network can get better results than what table 5 shows. But due to time limit and training from scratch, we

haven't finished the experiments.

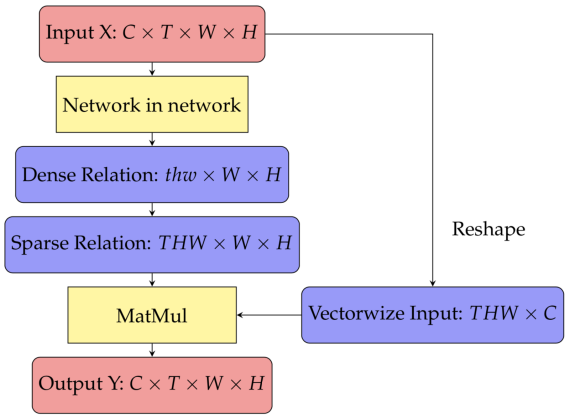


Figure 3. Our network structure for learning relations

Reference

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733. IEEE, 2017.

[2] Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. In *Advances in Neural Information Processing Systems*, pages 4470–4478, 2017.

[3] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In *Advances in neural information processing systems*, pages 3468–3476, 2016.

[4] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 776–780. IEEE, 2017.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 131–135. IEEE, 2017.

[7] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 2017.

[8] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[9] Mathew Monfort, Bolei Zhou, Sarah Adel Bargal, Tom Yan, Alex Andonian, Kandan Ramakrishnan, Lisa Brown, Quanfu Fan, Dan Gutfruend, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding.

[10] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[11] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *arXiv preprint arXiv:1705.02953*, 2017.

[12] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. *CVPR*, 2018.

Exploiting Spatial-Temporal Modelling and Multi-Modal Fusion for Human Action Recognition

Dongliang He*, Fu Li, Qijie Zhao, Xiang Long, Yi Fu, Shilei Wen

Baidu Research

Abstract

In this report, our approach to tackling the task of ActivityNet 2018 Kinetics-600 challenge is described in detail. Though spatial-temporal modelling methods, which adopt either such end-to-end framework as I3D [1] or two-stage frameworks (i.e., CNN+RNN), have been proposed in existing state-of-the-arts for this task, video modelling is far from being well solved. In this challenge, we propose *spatial-temporal network* (StNet) for better joint spatial-temporal modelling and comprehensively video understanding. Besides, given that multi-modal information is contained in video source, we manage to integrate both early-fusion and later-fusion strategy of multi-modal information via our proposed *improved temporal Xception network* (iTXN) for video understanding. Our StNet RGB single model achieves 78.99% top-1 precision in the Kinetics-600 validation set and that of our improved temporal Xception network which integrates RGB, flow and audio modalities is up to 82.35%. After model ensemble, we achieve top-1 precision as high as 85.0% on the validation set.

1 Introduction

The main challenge lies in extracting discriminative spatial-temporal descriptors from video sources for human action recognition task. CNN+RNN architecture for video sequence modelling [2, 3] and purely ConvNet-based video recognition [4, 5, 6, 7, 8, 1, 9] are two major research directions. Despite considerable progress has been made since several years ago, action recognition from video is far from being well solved.

For the CNN+RNN solutions, the feed-forward CNN part is used for spatial modelling, meanwhile the temporal modelling part, e.g., LSTM [10] or GRU [11], makes end-to-end optimization more difficult due to its recurrent architecture. Taking feature sequence extracted from a video as input, there are many other sequence modelling frameworks or feature encoding methods aiming at better temporal coding for video classification. In [12], fast-forward LSTM (FF-LSTM) and temporal Xception network are proposed for effective sequence modelling and considerable performance gain is observed against traditional RNN models in terms of video recognition accuracy. NetVLAD [13], ActionVLAD [14] and Attention Clusters [15] are recently proposed to integrate local features for action recognition and good results are achieved by these encoding methods. Nevertheless, separately training CNN and RNN parts is harmful for integrated spatial-temporal representation learning.

ConvNets-based solutions for action recognition can be generally categorized into 2D ConvNet and 3D ConvNet. Among these solutions, 2D or 3D two-stream architectures achieve state-of-the-art recognition performance. 2D two-stream architectures [4, 7] extract classification scores from evenly sampled RGB frames and optical flow fields. Final prediction is obtained by simply averaging the classification scores. In this way, temporal dynamics are barely explored due to poor temporal modelling. As a remedy for the aforementioned drawback, multiple 3D ConvNet models are invented for end-to-end spatial-temporal modelling such as T-ResNet [6], P3D [9], ECO [16], ARTNet [17] and S3D [18]. Among these 3D ConvNet frameworks, state-of-the-art solution is non-local neural network [19] which is based on I3D [1] for video modelling and leverages the spatial-temporal nonlocal relationships therein. However, 3D CNN is computational costly and training 3D CNN models inflated from deeper network suffers from performance drop due to batch size reduction.

In this challenge, we propose a novel framework called Spatial-temporal Network (StNet) to jointly model spatial-temporal correlations for video understanding. StNet first models local spatial-temporal cor-

*Corresponding author: hedongliang01@baidu.com

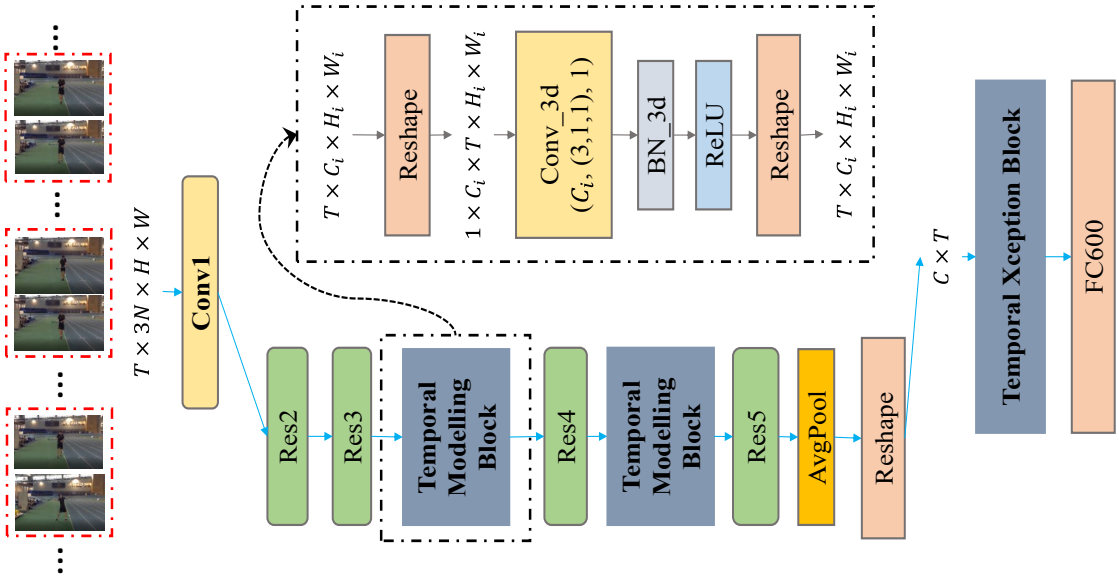


Figure 1: Illustration of constructing StNet based on ResNet [20] backbone. The input to StNet is a $T \times 3N \times H \times W$ tensor. Local spatial-temporal patterns are modelled via 2D Convolution. 3D convolutions are inserted right after the Res3 and Res4 blocks for long term temporal dynamics modelling. The setting of 3D convolution (# Output Channel, (temporal kernel size, height kernel size, width kernel size), # groups) is (C_i, (3,1,1), 1).

relation by applying 2D convolution over a $3N$-channel *super image* which is formed by sampling N successive RGB frames from a video and concatenating them in the the channel dimension. As for long range temporal dynamics, StNet treats 2D feature maps of uniformly sampled T *super images* as 3D feature maps whose temporal dimension is T and relies on 3D convolution with temporal kernel size of 3 and spatial kernel size of 1 to capture long range temporal dependency. With our proposed StNet, both local spatial-temporal relationship and long range temporal dynamics can be modelled in an end-to-end fashion. In addition, large number of convolution kernel parameters is avoided because we can model local spatial-temporal with 2D convolution and spatial kernel size of 3D convolution in StNet is set to 1.

Video source contains such multi-modal information as appearance information in the RGB frames, motion information among successive video frames and acoustic information in its audio signal. Existing works have proved that fusing multi-modal information is helpful [7, 15, 12]. In this challenge, we also utilize multiple modalities to boost the recognition performance. We improve our formerly proposed temporal Xception network [12] and enable it to integrate both early-fusion and later-fusion features of multi-modal information. This model is referred to as improved temporal Xception network (iTXN) in the following .

2 Spatial-Temporal Modelling

The proposed StNet can be constructed from existing state-of-the-art 2D CNN frameworks, such as ResNet [20], Inception-Resnet [21] and so on. Taking ResNet as example, Fig.1 illustrates how we can build StNet. Similar to TSN [7], we choose to model long range temporal dynamics by temporal snippets sampling rather than inputing the whole video sequence. One of the differences from TSN is that we sample T temporal segments which consists of N contiguous RGB frames rather than one single frame. These N frames are stacked to form a *super image* whose channel size is $3N$, so the input to the network is a tensor of size $T \times 3N \times H \times W$. We choose to insert two temporal modelling blocks right after the Res3 and Res4 block. The temporal modelling blocks are designed to capture the long-range temporal dynamics inside a video sequence and they can be implemented easily by leveraging Conv3d-BN3d-ReLU. Note that existing 2D CNN framework is powerful enough for spatial modelling, so we set both height kernel size and width kernel size of 3D convolution as 1 to save model

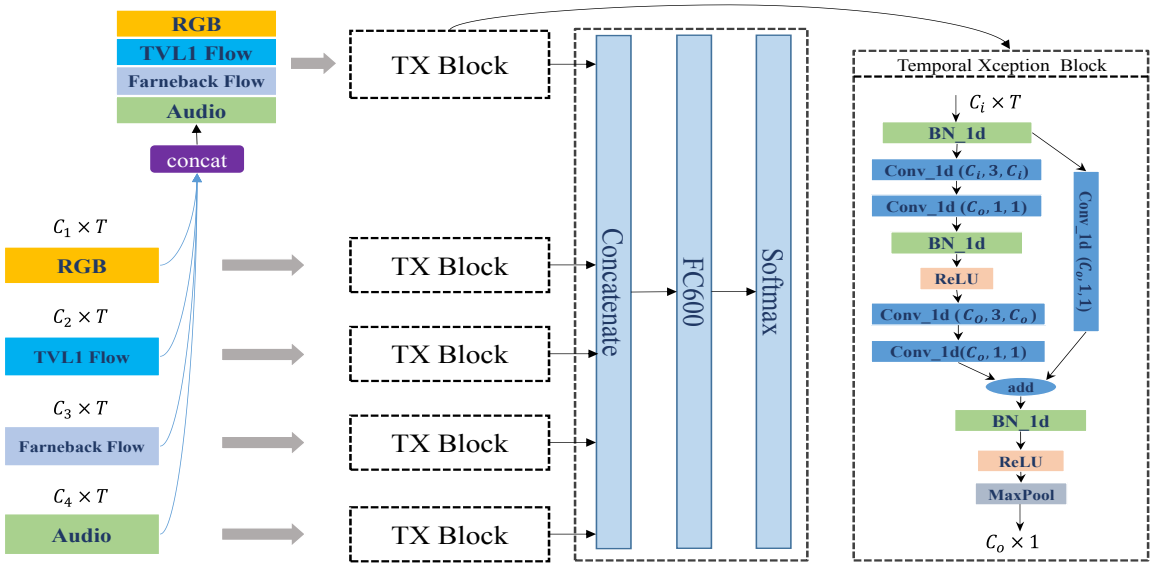


Figure 2: Block diagram of our proposed improved temporal Xception network (iTXN) framework for multi-modality integration. It is built based upon the temporal Xception network [12]. RGB, TV_L1 flow, Farneback flow and audio feature sequences are encoded individually for later-fusion and encoded jointly for early fusion with temporal Xception block, respectively. Numbers in bracket of temporal Xception block denote (# Output Channel, kernel size, # group) of Conv_1d layer.

parameters while the temporal kernel size is empirically set to be 3. As an augmentation, we append a temporal Xception block [12] to the global average pooling layer for further temporal modelling. Details about temporal Xception block can be found in the most right block of Fig.2.

To build StNet from other 2D CNN frameworks such as InceptionResnet V2 [21], ResNeXt [22] and SENet [23] is quite similar to what we have done with ResNet, therefore, we do not elaborate all such details here. In our current setting, N is set to 5, T is 7 in the training phase and 25 in the testing phase. As can be seen, StNet is an end-to-end framework for joint spatial-temporal modelling. A large majority of its parameters can be initialized from its 2D CNN counterpart. The initialization of the rest parameters following the below rules: 1) weights of Conv1 can be initialized following what the authors have done in I3D [1]; 2) parameters of 1D or 3D BatchNorm layers are initialized to be identity mapping; 3) biases of 1D or 3D Conv are initially set to be zeros and weights are all set to $1/(3 \times C_i)$, where C_i is input channel size.

3 Multi-Modal Fusion

Videos consist of multiple modalities. For instances, appearance information is contained in RGB frames, motion information is implicitly shown by the gradually change of frames along time and audio can provide acoustic information. For a video recognition system, utilizing such multi-modal information effectively is beneficial for performance improvement. Existing works [15, 12] have evidenced this point.

In this piece of work, we also follow the common practice to boost our recognition performance by integrating multi-modal information, i.e., appearance, motion and audio. Appearance can be explored from RGB frames with existing 2D/3D solution as well as our proposed StNet. To better utilize motion information, we extract optical flows from video sequences not only with the TV_L1 algorithm [24] but also with the Farneback algorithm [25]. As for audio information, we simply follow what have been done in [26, 12].

Fusing multi-modal information have been extensively researched in the literature. Early-fusion and later-fusion are the most common methods. In this paper, we propose to combine early-fusion and later-fusion in one single framework. As is shown in Fig.2, pre-

extracted features of RGB, TV_L1 flow, Farneback flow and audio are concatenated along with the channel dimension and its output is fed into a temporal Xception block for early fusion. These four feature modalities are also encoded with temporal Xception block individually. Afterwards, the early-fusion feature vector are concatenated with the individually encoded features of the four modality for classification.

4 Experiments

In this section, we report some experimental results to verify the effectiveness of our proposed frameworks. All the base RGB, flow and audio models evaluated in the following subsection are pre-trained on the Kinetics-400 training set and finetuned on the Kinetics-600 training set. All the results reported below are evaluated on the Kinetics-600 validation set.

4.1 Spatial-Temporal Modelling

To show the effectiveness of the proposed StNet, we have trained StNet with InceptionResnet V2 [21] and SE-ResNeXt 101 [23, 22] and a series of baseline RGB models, denoted as StNet-IRv2 and StNet-se101 respectively. As we know, the state-of-the-art 2D CNN models for action recognition is TSN [7], and we implemented TSN with InceptionResnet V2 and SE-ResNeXt 152 backbone networks. In the following context, we denote these two models as TSN-IRv2 and TSN-se152 respectively. We also introduced VLAD encoding + SVM on the TSN-IRv2 Conv2d_7b feature. Nonlocal neural network is state-of-the-art 3D CNN model for video classification, so we also finetuned nonlocal-net as a baseline model with the codes released by the authors.

Table 1: Performance comparison among StNet and baseline RGB models.

Model	Prec@1
TSN-IRv2 (T=50, cropsize=331)	76.16%
TSN-se152 (T=50, cropsize=256)	76.22%
TSN-IRv2 + VLAD + SVM	75.6%
Nonlocal Net (30crops, 32 frames/crop)	78.6%
StNet-se101 (T=25, cropsize=256)	76.08%
StNet-IRv2 (T=25, cropsize=331)	**78.99%**

Evaluation results are presented in Tabel.1. We can see from this table that StNet-IRv2 outperforms TSN-IRv2 by up to 2.83% in top-1 precision and it also achieves better performance than nonlocal-net. Please note that our StNet-se101 performs comparable with TSN-se152, which also evidences the superiority of the StNet framework.

4.2 Multi-Modal Fusion

In this work, we exploit not only RGB information, but also TV_L1 flow [24], Farneback flow [25] and audio information [26] extracted from video sources. The recognition performances with each individual modality are listed in Table.2. For multi-modality fusion, StNet-IRv2 RGB feature, TSN-IRv2 TV_L1 flow feature, TSN-se152 Farneback flow feature and TSN-VGG audio feature are used for better complementarity.

Table 2: Recognition performance of each individual modality.

Modality	Prec@1
TSN-IRv2 TV_L1	65.1%
TSN-IRv2 Farneback flow	69.3%
TSN-se152 Farneback flow	71.3%
StNet-IRv2 RGB	78.99
TSN-VGG audio	23%

To evaluate iTXN which is designed for multi-modal fusion, we compared it with several baselines: AttentionClusters [15], Fast-Forward LSTM and temporal Xception network which are proposed in [12]. The results are shown in the Table.3. From this table, we can see that iTXN is a good framework for integrating multiple modalities.

Our final results are obtained by ensembling multiple single modality models and several multi-modal models by gradient boosting decision tree (GBDT) [27]. After model ensemble, we finally achieve top-1 and top-5 precision of 85.0% and 96.9% on the validation set.

Table 3: Recognition performance of multi-modal fusion and model ensemble.

Model	Prec@1	Prec@5
temporal Xception network	81.8%	95.6%
Fast-Forward LSTM	81.6%	95.1%
AttentionClusters	82.3%	**96.0%**
iTXN	**82.4%**	95.8%
Model Ensemble	**85.0%**	**96.9%**

5 Conclusion

In this challenge, we proposed a novel StNet end-to-end framework to jointly model spatial-temporal patterns in videos for human action recognition. In order to better integrate multi-modal information which is naturally contained in video sources, we improved temporal Xception network to combines both early-fusion and later-fusion of multiple modalities. Experiment results have evidenced the effectiveness of the proposed StNet and iTXN.

References

[1] Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (July 2017)

[2] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 2625–2634

[3] Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: Deep networks for video classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 4694–4702

[4] Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems. (2014) 568–576

[5] Feichtenhofer, C., Pinz, A., Wildes, R.: Spatiotemporal residual networks for video action recognition. In: Advances in Neural Information Processing Systems. (2016) 3468–3476

[6] Feichtenhofer, C., Pinz, A., Wildes, R.P.: Temporal residual networks for dynamic scene recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 4728–4737

[7] Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: European Conference on Computer Vision, Springer (2016) 20–36

[8] Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE international conference on computer vision. (2015) 4489–4497

[9] Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with pseudo-3d residual networks. In: The IEEE International Conference on Computer Vision (ICCV). (Oct 2017)

[10] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8) (1997) 1735–1780

[11] Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259 (2014)

[12] Bian, Y., Gan, C., Liu, X., Li, F., Long, X., Li, Y., Qi, H., Zhou, J., Wen, S., Lin, Y.: Revisiting the effectiveness of off-the-shelf temporal modeling approaches for large-scale video classification. arXiv preprint arXiv:1708.03805 (2017)

[13] Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 5297–5307

[14] Girdhar, R., Ramanan, D., Gupta, A., Sivic, J., Russell, B.: Actionvlad: Learning spatio-temporal aggregation for action classification. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (July 2017)

[15] Long, X., Gan, C., de Melo, G., Wu, J., Liu, X., Wen, S.: Attention clusters: Purely attention based local feature integration for video classification. arXiv preprint arXiv:1711.09550 (2017)

[16] Zolfaghari, M., Singh, K., Brox, T.: Eco: Efficient convolutional network for online video understanding. arXiv preprint arXiv:1804.09066 (2018)

[17] Wang, L., Li, W., Li, W., Van Gool, L.: Appearance-and-relation networks for video classification. arXiv preprint arXiv:1711.09125 (2017)

[18] Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning for video understanding. arXiv preprint arXiv:1712.04851 (2017)

[19] Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. arXiv preprint arXiv:1711.07971 (2017)

[20] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778

[21] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI. (2017) 4278–4284

[22] Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, IEEE (2017) 5987–5995

[23] Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. arXiv preprint arXiv:1709.01507 (2017)

[24] Pérez, J.S., Meinhardt-Llopis, E., Facciolo, G.: Tv-l1 optical flow estimation. Image Processing On Line **2013** (2013) 137–150

[25] Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. Image analysis (2003) 363–370

[26] Hershey, S., Chaudhuri, S., Ellis, D.P., Gemmeke, J.F., Jansen, A., Moore, R.C., Plakal, M., Platt, D., Saurous, R.A., Seybold, B., et al.: Cnn architectures for large-scale audio classification. In: Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on, IEEE (2017) 131–135

[27] Friedman, J.H.: Stochastic gradient boosting. Computational Statistics & Data Analysis **38**(4) (2002) 367–378

Samsung & SIAT Submission to ActivityNet Challenge 2018

Wenhao Wu[1,2*], Wenbo Chen[2*], Shifeng Chen[1], Zhenbo Luo[2]

[1] Shenzhen Institutes of Advanced Technology, CAS, China
[2]Samsung R&D Institute of China - Beijing, China

{wh.wu, shifeng.chen}@siat.ac.cn, zb.luo@samsung.com

Abstract

This paper describes the method for the Samsung & SIAT submission to the trimmed activity recognition (Kinetics) tasks of the ActivityNet Large Scale Activity Recognition Challenge 2018. Motivated by [11], We integrate short-term temporal information with 3D Pooling models and long-term temporal information with temporal segment networks [10]. We also utilize multi-modal information, including audio and visual streams presenting in the videos. We pre-train models on Kinectics-400 dataset [4], then finetune them on Kinetics-600 dataset [14]. Our system finally obtains an averaged top-1 and top-5 error percentage of 14.977% on the test set.

1. Introduction

Trimmed action recognition, as its importance of understand human behaviors in videos, becomes a hot and basic topic in computer vision.

Benchmarks and related competitions have made great contributions to the action recognition research, such as HMDB-51 [7], UCF-101 [6], and so on. Especially the ActivityNet series challenges [2] and related datasets attract more and more research teams from the academic and industry. The latest version of Kinetics dataset, the Kinetics-600 [14] containing 500,000 trimmed video with 600 action categories, is released this year. It is an approximate super-set of the initial Kinetics-400 dataset [4] released in 2017. The actions cover a broad range of classes including human-object interactions such as playing instruments, and human-human interactions such as shaking hands, hugging, and so on.

Deep learning based frameworks become the main stream of action recognition in recent years. Among these frameworks, Two-stream [5], C3D [8], TSN [10], have achieved impressive results on the benchmark datasets. However, how to learn the spatiotemporal structure from videos is still remained a challenging task. One reason is mainly due to the computational resources needed for the task. The other reason is due to the lack of large and robust datasets. As a result, previous method transfer image-level object recognition representation to video-level action recognition representation.

In this report, we focus on learning video-based representation using Kinetics-600 dataset [14]. Different from image-level recognition, temporal structure and motion representation are essential for action recognition. We use temporal segment networks [10] as our base model, and increase the number of segments to model more long-term temporal information. We use 3D pooling units to model the short-term temporal information. Videos are naturally multimodal because a video can be decomposed into visual and acoustic components. Besides visual component such as RGB frames and stacked optical flow fields, we observe that acoustic signal coming along with the visual component provides complementary information. Combining all of the visual, and acoustic models, we attain a high recognition accuracy (average error of 14.977% on Kinetics-600 testing set).

The remaining part of this report is organized as follows. Section 2 present the details for trimmed action recognition. Section 3 concludes this work.

2. Methodology

In this year's challenge, the trimmed action recognition task in conducted on the Kinetics-600 dataset [14]. The dataset consists of approximately 500,000 video clips, and covers 600 human action classes with at least 600 video clips for each action class. Each clip lasts around 10 seconds and is labeled with a single class. Our efforts on this task are focused on how to utilize temporal information, and muti-

*Work done while as intern in Machine Learning Lab, Samsung R&D Institute of China - Beijing.

stream CNNs, to learn the better 2D-CNN features for video-based action recognition.

2.1. 3D Pooling with Temporal Segment Network

We follow the pipeline of temporal segment networks (TSN) [10] to model the long-term temporal information. However, in the original TSN [10], the underlying ConvNet models are using 2D inputs. To further exploit short-term temporal variation between neighboring frames, we propose to inflate the original 2D model into a 3D version. The number of input frames in each snippet is changed to 8. Specifically, in order to reduce computation, we inflate pooling layers to 3-dimensional instead of using 3-dimensional convolutional. Then using TSN, we divide every 10-second video to a fixed number of segments, such as 3, 5 and 7. We found that using 3D pooling with TSN can get a better boost of performance compared with original TSN models. The results are illustrated in Table 1. During training, one short snippet is sampled from each segment, which forms a sparse snippet sampling scheme. The snippet-wise prediction is then aggregated using average pooling. During testing, we follow the standard procedure of using 25 frames uniformly extracted from the testing videos and average the predictions. Due to Kinetics-600 is an approximate super-set of Kinetics-400, first we experiment with several normal network architectures such as BN-Inception [3] and Inception-V3 [13] on Kinetics-400 [4]. And these models pre-trained from ImageNet [1]. Then we finetune the Kinetics-400 models on Kinetics-600 with more segments. The results are illustrated in Tabel2.

Table 1. Performance comparison of 3D-pool models and original TSN models on Kinetics-400 val set [4]

Models	RGB		Flow	
	Top-1	Top-5	Top-1	Top-5
BN Inception	72.68%	90.10%	64.55%	85.72%
Inception V3	73.85%	91.04%	65.20%	86.63%
TSN	69.03%	89.12%	62.81%	83.65%

Table 2. Performance of different 3D-pool models on Kinetics-600 val set [14].

Modality	base model	Top-1	Top-5
RGB	Inception BN	72.16%	91.02%
FLOW	Inception BN	65.72%	86.79%
RGB	Inception V3	73.79%	92.10%
FLOW	Inception V3	66.59%	87.30%

2.2. Acoustic Information

Audio also provides a lot of important information for video classification. For example, occlusions seriously decrease the recognition accuracy of RGB or optical flow features based methods, but does not affect acoustic features.

We use a CNN-based [12] audio action recognition system to extract acoustic features. We transform the raw audio signal to log-mel spectrogram as input of CNN model. Audio preprocessing is as follows:

1) Resample audio to 16 kHz mono, then compute spectrogram using magnitudes of the Short-Time Fourier Transform with a window size of 25 ms, a window hop of 10 ms, and a periodic Hann window.
2) Compute mel spectrogram by mapping the spectrogram to 64 mel bins and stabilize log mel spectrogram by applying log where an offset is used to avoid logarithm of zero.
3) These features are then framed into non-overlapping examples of 0.96 seconds, where each example covers 64 mel bands and 96 frames of 10 ms each.

We also attempt another scale spectrogram with parameters 2.5ms for window size, 1ms for window hop and 9.6ms per frames. These examples are then fed into the CNN model to extract embedding.

The CNN architectures are Inception-ResNet-v2 [9] and BN Inception [3], initialized by ImageNet [1] pre-trained parameters. We also train the acoustic models with multiscale spectrogram using temporal segment network framework [10]. The results of them are summarized in Table 3. We find large scale spectrogram always performs better, and increase the number of segmentation not always improve accuracy. Finally, we confirm that audio features and visual features can complement each other in an ensemble way.

Finally, we choose the averaging fusion approach to ensemble muti-stream models as it is a simple and fast approach. The result is summarized in Table4.

Table 3. Results of audio CNN on Kinetics600 [14] val set

Network	Scale	TSN segment	Top-1	Top-5
Inception-ResNet-v2	960*	1	0.226	0.389
Inception-ResNet-v2	960	3	0.231	0.396
Inception-ResNet-v2	960	10	0.232	0.401
BN Inception	960	1	0.171	-
BN Inception	960	3	0.187	-
BN Inception	960	10	0.193	-
BN Inception	96	3	0.096	-
BN Inception	96	10	0.116	-
Ensemble	-	-	0.254	-

*960 is large scale spectrogram which is extracted every 960 ms, 96 is small scale spectrogram which is extracted every 96 ms.

Table 4. Results on Kinetics-600 [14] test set. The avg. error is the average of top-1 and top-k error.

Models	avg. error
model ensemble	0.14977

3. Conclusions

This paper reports our team's solution to the task of trimmed action recognition. We proposed a three-stream CNN to exploit the spatial, motion and acoustic information

of the trimmed videos. We presented the importance of 3D spatiotemporal models for visual component. We exploited audio information as the complementary to visual information, improving performance in an ensemble way.

References

[1] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[2] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015.

[3] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.

[4] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[5] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[6] K. Soomro, A. Roshan Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*, 2012.

[7] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2556–2563, 2011.

[8] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 4489–4497, 2015. 1, 2

[9] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284, 2017.

[10] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36, 2016.

[11] Y. Zhao, B. Zhang, Z. Wu, S. Yang, L.Zhou, S.Yan, L.Wang, Y.Xiong, D.Lin, Y.Qiao, X.Tang, CHUK & ETHZ & SIAT submission to ActivityNet Challenge 2017. *arXiv preprint arXiv:1710.08011*, 2017.

[12] Y. Xiong, L. Wang, Z. Wang, B. Zhang, H. Song, W. Li, D. Lin, Y. Qiao, L. Van Gool, and X. Tang. Cuhk & ethz & siat submission to activitynet challenge 2016. *arXiv preprint arXiv:1608.00797*, 2016.

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[14] https://deepmind.com/research/open-source/open-source-datasets/kinetics

Alibaba-Venus at ActivityNet Challenge 2018 - Task B Spatio-temporal Action Localization (AVA)

Hao Zhang, Liu Liu, Xiao-Wei Zhao and Yang Liu
Alibaba Group, Hangzhou, China
{zh153861, diana.ll, zhiquan.zxw, panjun.ly}@alibaba-inc.com

Abstract

This technical report details the framework used by Alibaba-Venus team for the task of spatio-temporal action localization (AVA) in the ActivityNet challenge 2018. Our framework basically follows the common infrastructure, i.e., "actor detector" plus "action classifier", as proposed in [2], with some modifications on the choices of object-detector and 3D net backbone. Specifically, we adopt RFB-Net, a computational efficient SSD variant, as "actor detector" and Pseudo-3D Residual Net (P3D) / Inflated 3D ConvNet (I3D) as "action classifier". For simplicity, we name models according to their 3D-net backbones: P3D-RFBNet and I3D-RFBNet. During the challenge, we test the P/I3D-RFBNet on different modalities (i.e., RGB and optical-flow), with different fusion and testing-augmentation strategies. Experimental results show that P/I3D-RFBNet achieve promising performances on both RGB and optical-flow modalities, and benefit from fusion/testing-augmentation strategies.

1. Introduction

Human action/activity classification has been intensively studied in the past five years. The proposal of two-stream 3D convolutional neural networks, such as C3D [7], I3D [1], P3D [5] and etc., greatly boost the development of video-level activities classification, laying a good base for even finer-grained level activities detection: atomic action of human instance. In 2017, google releases a carefully annotated action dataset named "Atomic Visual Actions" (AVA) [2], which annotates human instances in videos with bounding boxes and action labels, targeting at fine-grained action detection. Unlike promising results obtained on video-level activity classification, the new atomic action detection is more challenging, since it requires much finer inspections.

For this year's spatio-temporal action localization challenge (AVA), we propose P/I3D-RFBNet, following the common infrastructure: "actor detector" plus "action classifier" in [2], with some modifications. Specifically, we replace Faster-RCNN with RFBNet [3] as actor detector, since RFBNet achieves a good trade-off between detection speed and accuracy. For action recognition, we incorporate 3D-net, RoI-align layer and multi-layer perceptron as action classifier. Equipped with P/I3D-RFBNet, we mainly explore the following aspects during the challenge.

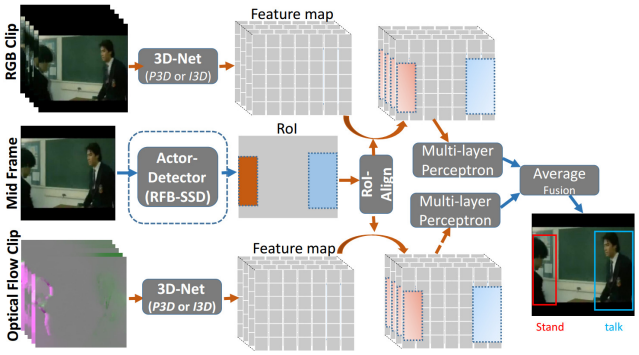


Figure 1. Overview of P/I3D-RFBNet. A mid-frame and its wrapping clip are separately fed into actor detector and action classifier. The former serves for actor locating, whereas, the latter serves for action classifying (the figure is best viewed in color).

- RFBNet as actor detector: Unlike baselines [6], we chose the lightweight RFBNet, instead of the computationally heavy Faster-RCNN, as actor detector. Experimental results show that the RFBNet even demonstrates a superior detection performance than the Faster-RCNN on AVA dataset.

- P/I3D-RFBNet on RGB/optical-flow modalities: We separately train P/I3D-RFBNet models on RGB and optical-flow modalities. We observe that P/I3D-RFBNet achieve promising performances on both modalities.

- Modalities' ensemble: We adopt average-fusion strategy to ensemble models trained on different modalities. Specifically, predicted scores by different models on the same actor bounding boxes are averaged. Ex-

perimental results show that the ensemble strategy introduces considerable improvements.

- Testing augmentation: Actor action might exceed the duration of fixed-length clip, we propose a testing augmentation method named *shift-pred* to ease the problem. The main idea of *shift-pred* is slightly jittering video-clip forward or backward in temporal dimension before feeding it into the neural network. We fuse results obtained with/without jittering to get a better performance.

The remainings of this paper are organized as below. In the methodology part, we briefly introduce modules in our infrastructure. Then we present experimental results for AVA challenge. Finally, we conclude this report in the last section.

2. Methodology

As shown in Figure 1, our model is composed of two modules (separated by dased line): actor detector and action classifier. To ensure a large batch size, we train the two modules in a two-step style. Specifically, actor detector is firstly fine-tuned on AVA-v2 dataset to detect human beings. Then, we fixed parameters in actor detector and only train action classifier. Also, we separately train action classifier on RGB and optical-flow modalities. Prediction scores of RGB and optical-flow modalities on the same bounding boxes are averaged. We will elaborate the two modules as below.

2.1. Actor Detector

We adopt RFBNet [3], a variant of SSD [4], as actor detector. The RFBNet shares the following advantages: real-time computational speed and competitive detection accuracy as Faster-RCNN++ [1]. We adopt the optimal settings, i.e., RFBNet512-E, in training human detector. On AVA-v2 validation set, RFBNet achieves 93.1% mAP@0.5, which is higher than 75.3% by Faster-RCNN in [2].

2.2. Action Classifier

Human action classifier is composed of three submodules: 3D net, RoI-align layer, and mutli-layer perceptron. Specifically, given RoIs, RoI-algin layer fetches sub feature-maps out of 3D feature-map and then feeds them into multi-layer perceptron for action categorization. Following common settings, our multi-layer perceptron contains three fully connected (fc) layers, with two of them followed by relu activations. To avoid overfitting, dropout layer is also launched before the first two fc layers.

[1] https://github.com/ruinmessi/RFBNet

3. Experiments

3.1. Settings

Experimental settings regarding frame representation, hyperparameters of P/I3D-RFBNet are elaborated in this section.

Frame Representation: We extract frames out of videos at 25 fps, regardless of their original frame rate, and then resize all frames in 512×512 without keeping their aspect ratio. Same fps and resolution as RGB modality, we extract TVL1 optical flow for each video using toolkit provided in [2]. Denoting x_i as the i-th frame to be tested, we collect a clip $c_i = \{x_{i-m}, x_{i-m+1}, \ldots, x_{i+n}\}$ wrapping x_i. The frame-clip pair $\{x_i, c_i\}$ serves as input for actor-detector/action-classifier module. For test witout shift-pred, we set $(m = 7, n = 8)$, whereas, for two tests with shift-pred, we set $(m = 4, n = 11)$ and $(m = 11, n = 4)$.

Hyperparameters of P/I3D-RFBNet: When forwarding 3D net, each clip c_i can be represented by $f_i \in R^{C \times T \times H \times W}$ feature-map.Average-pooling operator is applied to squeeze the temporal channel, transfering f_i into $\hat{f}_i \in R^{C \times H \times W}$. For RoI-align layer, we experimentally set RoI size to be 16×16 for I3D-RFBNet, and 7×7 for P3D-RFBNet.

3.2. Results

Table 1. AVA-v2: Performances of modalities and their ensembles on validation and test set (frame-mAP@0.5 in percentage).

	AVA-v2 (Val) %	AVA-v2 (Test) %
Faster-RCNN (RGB, ResNet-101)[2]	11.25	-
I3D-RFBNet (RGB)	15.59	-
I3D-RFBNet (OptFlow)	15.70	-
P3D-RFBNet (RGB)	15.69	-
P3D-RFBNet (OptFlow)	13.63	-
I3D+P3D-RFBNet (RGB)	17.33	15.30
I3D+P3D-RFBNet (OptFlow)	16.10	
I3D+P3D-RFBNet (RGB + OptFlow)	19.09	-
I3D+P3D-RFBNet (RGB + OptFlow) + Shift-Pred	**19.37**	**17.67**

We conduct experiments on different modalities, cross-modalities fusion, and testing augmentation, the results are presented in Table 1. We observe that: (1). P/I3D-RFBNet show promising results on both RGB/optical-flow modalities, verifying their robustnesses. (2). Fusion of P3D-RFBNet and I3D-RFBNet shows considerable improvement, indicating 3D net backbones are complementary to each other. (3). Multi-modalities fusion brings further improvements. (4). Testing augmentation, i.e., shift-pred, also show its effectiveness.

[2] https://github.com/yjxiong/dense_flow

4. Conclusion

We have present P/I3D-RFBNet for ActivityNet challenge 2018, the P/I3D-RFBNet shows their effectiveness in detecting actors' actions on RGB/optical-flow modalities. Also, we find some useful engineering tricks, such as shift-pred in the exploration.

References

[1] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733. IEEE, 2017.

[2] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. *arXiv preprint arXiv:1705.08421*, 2017.

[3] S. Liu, D. Huang, and Y. Wang. Receptive field block net for accurate and fast object detection. *arXiv preprint arXiv:1711.07767*, 2017.

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[5] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5534–5542. IEEE, 2017.

[6] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[7] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

Human Centric Spatio-Temporal Action Localization

Jianwen Jiang[1], Yu Cao[2], Lin Song[3], Shiwei Zhang[4] Yunkai Li[5], Ziyao Xu[5], Qian Wu[6], Chuang Gan[1*], Chi Zhang[5*], Gang Yu[5*]
[1]Tsinghua University, jjw17@mails.tsinghua.edu.cn, ganchuang1990@gmail.com
[2]Beihang University, cqcy1208@buaa.edu.cn
[3]Xian Jiaotong University, stevengrove@xtu.xjtu.edu.cn
[4]Huazhong University of Science and Technology, swzhang@hust.edu.cn
[5]Megvii Inc. (Face++), {liyunkai, xuziyao, zhangchi, yugang}@megvii.com
[6]Zhejiang University, wq1601@zju.edu.cn

***Abstract*—This paper describes our solution for the spatio-temporal action localization of ActivityNet AVA challenge. Our system is consisted of three components: a human detector, an action classification module and an actor-target relation network. We first apply a region proposal network (RPN) to detect human in the videos, since AVA mainly contains human-centric action categories. Then we conduct the action classification by adopting the ROI pooling operation on the human regions. In order to capture the human-object relationships, we further design an actor-target relation network, which is achieved with a non-local operation between the ROI and its surrounding regions. Our method obtains 25.63% and 25.75% in terms of mean average precision (mAP) on the validation set of the two tracks, and 20.56% and 20.78% on the testing set.**

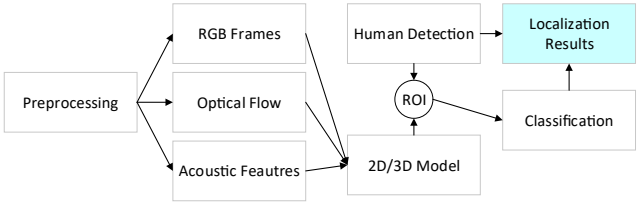


Fig. 1. The designed framework in our method. We split the spatio-temporal action localization into two subtasks, including human detection and action classification. Given the detections, we mainly focus on extracting multi vision cues, such as appearance information, motion information, and acoustic features. By applying ROI pooling, we can integrate the results from different models.

I. Introduction

Spatial-temporal action recognition and localization has received significant research attention in the computer vision communities [3], [28], [30], [31] due to its enormous applications such as public security, event recognition and video retrieval. There are some publicly available datasets such as UCF-Sports [16], J-HMDB [6] and UCF101 [21], [8], which have made great contribution to improve the performance for the task of action recognition and detection. Based on these benchmarks, there are a few promising deep model based methods, including TS (two streams) framework [18], C3D [22], TSN [25], p3d [14] and Artnet [23] for action recognition. These methods mainly try to extract different vision cues, such as short video clips [14], [22], [23], motion information [18] and long-range video clips [25]. Meanwhile, recent object detection methods, such as faster-RCNN [15], light head RCNN [9] and megdet [12], also make significant process for the general object detection.

Recently, some detection methods such as ACT [7], online method [20], multi-region faster-RCNN [13], achieve impressive results on the public datasets in the detection frameworks. In this challenge, the AVA dataset [4] is more challenging and we aim to apply different clues to extract video representation. In this report, we mainly adopt three modalities, including appearance, motion and audio information. Noting that, the audio feature is only applied in the full track. To conduct action detection, we design our method in the Faster-RCNN [15] framework. To better fit the framework on the action localization, we propose to apply a good pretrained human detectors as the RPN module, shown in Fig. I. Following the RPN, we train the action classification network in an end-to-end manner. Moreover, we design an actor-target relation (ATR) network to extract correlation between the actors and the corresponding targets. For this purpose, we conduct non-local operation between the ROI and its surrounding regions. The applied base models mainly focus on short- or long-term input clips, including i3d [1] with non-local module [26], C3D [22], and TSN [25].

For the RPN module, we apply FPN model [10] because of its high recall and precision. Given the proposal regions, we apply ROI pooling [15] to extract features and classify each proposals. After that, a posterior fusion strategy is used to give the final predictions of action categories of every corresponding target. Attributed to the structure of the designed model, we obtain 10% gain than the baseline method [4]. We show the overview in the Fig. I.

The remaining sections are organized as follows. Section II presents the details of our method. In section III, we also present some experimental results. Finally, this report concludes in section IV.

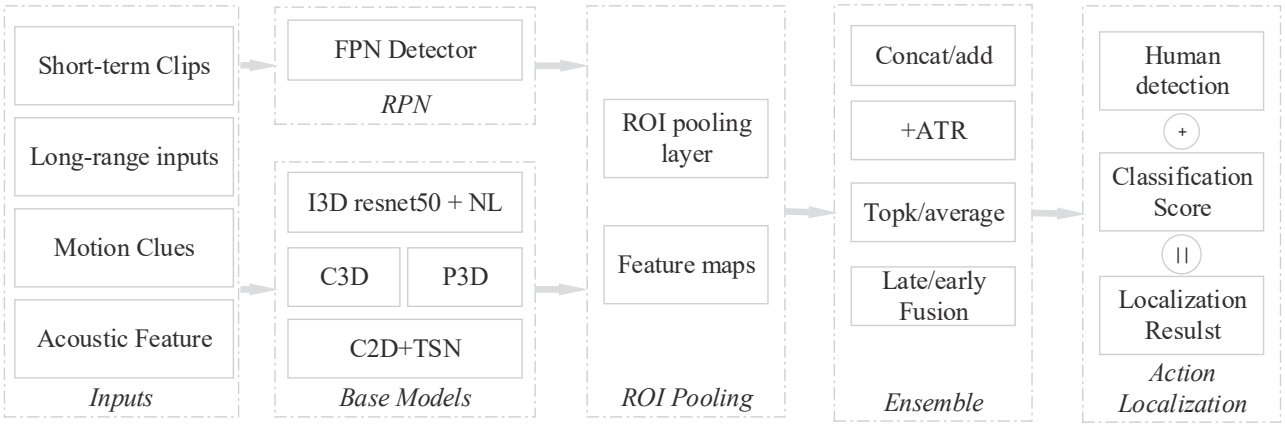


Fig. 2. The overview of our method. First, we explore different vision cues, which are respectively fed into RPN and feature extractors. Then we apply ROI pooling operation based on the proposal regions and the corresponding feature maps. After that, we explore different integration strategies on the applied models. Finally, we calculate the location results by considering the classification results and proposal regions.

II. The proposed Method

In this section, we first introduce the utilized multiple clues. Then we present the framework of action localization and classification in both tasks of AVA challenges.

A. Multi clues

Action localization is a complex task and is very challenging. We explore several modalities for this task, including short-term clips, long-range temporal structure, motion information and acoustic features.

Short-term clips. Inspired by most 3D CNNs, such as C3D [22], P3D [14] and I3D [1], we apply several continuous clips as input to extract short-term video representation. As shown in the benchmark [4], long clip and large input size is helpful to improve the performance. Therefore, we explore the applied inputs with different length and size to better understand their influence to the final action classification results.

In the AVA dataset [4], the action instances are sparsely annotated per second. Therefore, we extract one clip in a short time interval to predict the target actions. In our method, we apply the I3D Resnet [26], and P3D [14] to conduct the video representation. All the models are pretrained in Kinetics [8] in advance.

Long-range sampling. TSN [25] is proved to be a powerful method of long-range temporal structure modeling. Similar to TSN [25], we apply uniform sampling strategy to sample n frames for the model to learn. In our method, we find it effective to apply the sampling in the traditional 2D CNN. Therefore, we just adopt 2D model in this framework, such as resnext [27], resnet [5] and artnet18 [24].

Before the ROI pooling module [15], we integrate the frame-level feature by using average pooling scheme along the time axis.

Motion clues. To better extract micro motion information between two consecutive frames, we calculate optical flow to be used as a modality of input for the deep models. We first compute the horizontal and vertical motion maps, which construct the two independent channels. For the third channel, we simply apply point-wise multiplication between these two maps.

In this paper, we extract optical flow by applying TV-L1 [29] method which is integrated in the Opencv tools. Moreover, we also explore different methods of optical flow, such as Farneback [2], to add variety to the modality.

Acoustic features. Acoustic information is also discriminative for some actions, such as "play musical instrument", "sing". Therefore, we try to extract acoustic feature to improve our video representation. Similar to CNN based audio classification task [17], we divide the videos into frames every 1s, following which Fourier transformation and histogram integration are adopted. Given the new frames, we apply a VGG16 [19] model to conduct action classification based on a pre-trained model on the Kinetics dataset.

B. Action Localization

In this section, we mainly introduce RPN module and classification module.

RPN module. The goal of these two tasks is to localize human centric spatio-temporal action, hence we hold the point that the RPN module should have good performance on human detection. In our method, we apply feature pyramid networks (FPN) [10] to reach this goal, for its better performance. The FPN detector is first pretrained on MSCOCO dataset [11] and then is fine tuned on the AVA dataset [4]. By this means, we can obtain 96.5% recall and 81.6% accuracy on the evaluation set on an Intersection over Union(IoU) threshold of 0.5.

Action classification. As aforementioned, we design our method in the faster RCNN [15] framework. We apply the ROI pooling [15] strategy based on the proposal regions and the corresponding base models. We locate the ROI pooling layer after the last feature maps, followed by a classification branch. For the classification network, sigmoid function is used as in [4]. Finally, the output of the classification branch is used as the classification probability prediction results of the corresponding proposal boxes.

TABLE I
RESULTS ON VALIDATION SET.

Model	Input	Modality	Operation	mAP (%)
Faster-RCNN [4]	(3, 40(RGB)+40(Flow), 360, 400)	RGB + Flow	-	16.2
i3d resnet50 + NL	(3, 20, 224, 224)	RGB	-	19.33
	(3, 20, 224, 224)	RGB	ATR	20.01
	(3, 40, 224, 224)	RGB	40 clips	19.37
	(3, 20, 360, 400)	RGB	(360,400) size	19.86
	(3, 20(RGB)+20(Flow), 224, 224)	RGB + Flow	add	21.66
P3D199	(3, 20(RGB)+20(Flow), 224, 224)	RGB + Flow	-	17.87
resnet152	(3, 20, 224, 224)	RGB	TSN	14.68
artnet18	(3, 20, 224, 224)	RGB	-	16.67
Vgg16	-	Audio	-	6.5
Ensemble(Vison Only)				**25.63**
Ensemble (Full)				**25.75**

To further improve the performance, we also explore following several different strategies: *(i)* we concatenate or add the feature maps from different networks; *(ii)* we simply average the scores before or after sigmoid function; *(iii)* top-k fusion scheme are adopted for the ensemble process; *(iv)* we concatenate the features of RGB and Flow streams on the fully connected layer. *(v)* ROI align method is also explored.

In our method, we integrate all the model to calculate the results of our human detections. In the experiments, we find that apply $k = n//2$ (n is the number of the total applied models), and fusion before sigmoid function can lead to better results.

C. Extract Actor-Target Relationship

In the AVA dataset [4], we observed that the annotation boxes mainly contain the human but lose much attention on the targets, such as "grab (a person)" and "hug (a person)". We speculate the performance could be further improved by incorporating the the Actor-Target relationship (ATR).

Inspired by the successful application of the non-local [26] network on the action recognition, we adopt the non-local operation to extract ATR. Particularly, we conduct non-local operation between the ROI feature and features outside the bounding boxes. By this means, we can learn the discriminative relationship related to the actors. Experiments also show that this structure effectively captures the motion by bridging between people and the interactive objects through space and time domain.

D. Training

In this section, we present some details of our method during training stage. We train our network end-to-end with invariant 0.001 learning rate. For each model, we train about 5 epoches. We train our model on the 8 P40 GPUs for each experiments and the batch size is 16. When fusing different models, we freeze the base model before ROI pooling layer.

III. EXPERIMENT RESULTS

In this section, we respectively report our performance on the validation and testing set in the Table II-A and Table III. In the Table II-A, we show the results with different 2D/3D models. All the 3D models are first pretrained on Kinetics [8], and all the 2D models are pretrained on the Imagenet. Extracting the ATR can obtain about 0.68%, which means it is indeed helpful to learning the relationship for action classification. Finally, our method obtain 25.63% and 25.75% in terms of mAP on the two tasks.

In the Table III, our method get 20.56% and 20.78% mAP on the testing set. We find that there is about a gap of 5% between validation and testing set, we think the reason may be that there are different number of videos of the two sets.

TABLE II
RESULTS ON TESTING SET.

Tasks	mAP(%)
Computer Vision ONLY	20.55
Full	20.77

IV. CONCLUSION

In the Activitynet-AVA Challenge 2018, we propose a new framework for the human centric spatio-temporal action localization. We design our method under the faster-RCNN framework, but propose to apply a good human detector as the RPN module. Meanwhile, we apply non-local operation between the proposal regions and their surrounding regions to extract actor-target relation (ATR). Moreover, we also explore different integration strategies to extract multi vision cues. By this means, we achieve significant improvement again the baseline method. In the future, we will explore the correlation between different actions and learn this correlation in the deep models.

REFERENCES

[1] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 4724–4733. IEEE, 2017.

[2] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer, 2003.

[3] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. G. Hauptmann. Devnet: A deep event network for multimedia event detection and evidence recounting. In *CVPR*, pages 2568–2577, 2015.

[4] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. *arXiv preprint arXiv:1705.08421*, 2017.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *ICCV*, pages 3192–3199. IEEE, 2013.

[7] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid. Action tubelet detector for spatio-temporal action localization. In *ICCV*, 2017.

[8] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[9] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun. Light-head r-cnn: In defense of two-stage object detector. *arXiv preprint arXiv:1711.07264*, 2017.

[10] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017.

[11] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.

[12] C. Peng, T. Xiao, Z. Li, Y. Jiang, X. Zhang, K. Jia, G. Yu, and J. Sun. Megdet: A large mini-batch object detector. *arXiv preprint arXiv:1711.07240*, 2017.

[13] X. Peng and C. Schmid. Multi-region two-stream r-cnn for action detection. In *ECCV*, pages 744–759. Springer, 2016.

[14] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 2017.

[15] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[16] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, pages 1–8. IEEE, 2008.

[17] e. S. Hershey, S. Chaudhuri. Cnn architectures for large-scale audio classificatio. *arXiv preprint arXiv:1609.09430*, 2017.

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

[19] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR, abs/1409.1556*, 2014.

[20] G. Singh, S. Saha, M. Sapienza, P. Torr, and F. Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *ICCV*, pages 3637–3646, 2017.

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

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

[23] L. Wang, W. Li, W. Li, and L. Van Gool. Appearance-and-relation networks for video classification. *arXiv preprint arXiv:1711.09125*, 2017.

[24] L. Wang, W. Li, W. Li, and L. Van Gool. Appearance-and-relation networks for video classification. *CVPR*, 2018.

[25] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool. Temporal segment networks: Towards good practices for deep action recognition. *ECCV*, 22(1):20–36, 2016.

[26] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. *arXiv preprint arXiv:1711.07971*, 2017.

[27] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 5987–5995. IEEE, 2017.

[28] G. Yu and J. Yuan. Fast action proposals for human action detection and search. In *CVPR*, pages 1302–1311, 2015.

[29] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint Pattern Recognition Symposium*, pages 214–223. Springer, 2007.

[30] S. Zhang, C. Gao, F. Chen, S. Luo, and N. Sang. Group sparse-based mid-level representation for action recognition. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(4):660–672, 2017.

[31] S. Zhang, C. Gao, J. Zhang, F. Chen, and N. Sang. Discriminative part selection for human action recognition. *IEEE Transactions on Multimedia*, 20(4):769–780, 2018.

A Simple Kinetics-600 Baseline for AVA

João Carreira[†], Carl Doersch[†], Rohit Girdhar[†‡], Andrew Zisserman[†*]

[†]DeepMind, [‡]Carnegie Mellon University, [*]Oxford University

Abstract. We introduce a simple baseline for action localization on the AVA dataset, that showcases the value of pretraining video models on the Kinetics-600 dataset. The model builds upon the faster-RCNN bounding box detection framework, adapted to operate on pure spatiotemporal features – in our case produced exclusively by an I3D model pretrained on Kinetics-600. This model obtains 21.6% average AP on the validation set of AVA v2.1, up from 14.5% of the best RGB spatiotemporal model used in the original AVA paper (which was pretrained on Kinetics-400 and ImageNet), and up from 11.3% of the publicly available baseline using a ResNet-101 image feature extractor, that was pretrained on ImageNet.

1 Introduction

Despite considerable advances in the ability to estimate position and pose for people and objects, the computer vision community lacks models that can describe what people are doing at even short-time scales. This has been highlighted by new datasets such as Charades [1] and AVA [2], where the goal is to recognize the set of actions people are doing in each frame of example videos – e.g. one person may be standing and talking while holding an object in one moment, then it puts the object back and sits down on a chair. The winning system of the Charades challenge 2017 obtained just around 21% accuracy on this per-frame classification task. On AVA the task is even harder as there may be multiple people and the task requires also localizing the people and describing their actions individually – a strong baseline gets just under 15% on this task [2]. The top approaches in both cases used I3D models trained on ImageNet [3] and the Kinetics-400 dataset [4].

Recently the Kinetics-400 dataset was expanded into Kinetics-600 [5], introducing 50% new classes and approximatey 60% new training videos. An open question is how much impact this size increase has in the quality of the models when transferring from Kinetics to different tasks and datasets. In this paper we aim to answer this question using the new AVA dataset. The resulting model was an entry to the 2018 AVA challenge.

2 Model and Approach

Our model is inspired by I3D [6] and Faster R-CNN [7]. We start from labeled frames in the AVA dataset, and extract a short video clip, typically 64 frames,

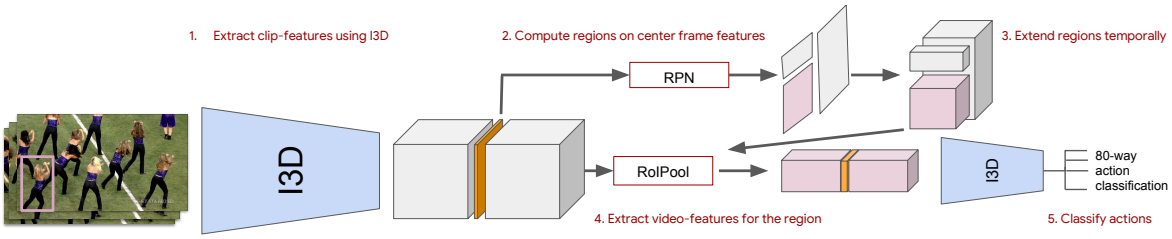


Fig. 1. Network architecture. We build upon I3D and Faster R-CNN architectures. A video clip is passed through the first few blocks of I3D to get a video representation. The center frame representation then is used to predict potential 'person' regions using a region proposal network (RPN). The proposals are extended in time by replicating, and used to extract a feature map for the region using RoIPool. The feature map is then classified into the different actions using two I3D blocks.

around that keyframe. We pass this clip through I3D blocks upto `Mixed_4f`, which are pre-trained on the Kinetics-600 dataset for action classification. The feature map is then sliced to get the representation corresponding to the center frame (the keyframe where the action labels are defined). This is passed through the standard region proposal network (RPN) [7] to extract box proposals for persons in the image. We keep the top 300 region proposals for the next step: extracting features for each region that feed into a classifier.

Since the RPN-detected regions corresponding to just the center frame are 2D, we extend them in time by replicating them to the match the temporal dimension of the intermediate feature map, following the procedure for the original AVA algorithm [2]. We then extract an intermediate feature map for each proposal using the RoIPool operation, applied independently at each time step, and concatenated in time dimension to get a 4-D region feature map for each region. This feature map is then passed through the last two blocks of the I3D model (up to `Mixed_5c`, and classified into each of the 80 action classes. The box classification is treated as a non-exclusive problem, so probabilities are obtained through an independent sigmoid for each class. We also apply bounding box regression to each selected box following Faster-RCNN [7], except that our regression is independent of category (since the bounding box should capture the person regardless of the action). Finally we post-process the predictions from the network using non-maximal supression (NMS), which is applied independently for each class. We keep the top-scoring 300 class-specific boxes (note that the same box may be repeated with multiple different classes in this final list) and drop the rest.

3 Experiments

We trained the model on the training set using a synchronized distributed setting with 11 V100 GPUs. We used batches of 3 videos with 64 frames each, and augmented the data with left-right flipping and spatial cropping. We trained the model for 500k steps using SGD with momentum and cosine learning rate

annealing. Before submitting to the challenge evaluation server, we finetuned the model further on the union of the train and validation sets. We tried both freezing batch norm layers and finetuning them with little difference in performance.

Results of our model on the validation set are compared with resuts from the models in the AVA paper in table 1. The RGB-only baseline [2] used the same I3D feature extractor that we used but pretrained on ImageNet then Kinetics-400, whereas our model was just pretrained on the larger Kinetics-600. This baseline differs from ours in a few other ways: 1) it used a ResNet-50 for computing proposals and I3D for computing features for the classification stage, whereas we only use the same I3D features for both things; 2) our model preserves the spatiotemporal nature of the I3D features all the way to the final classification layer, whereas theirs performs global average pooling in time of the I3D features right after ROI-pooling; 3) we opted for action-independent proposal boxes.

The RGB+Flow baseline is similar but also uses flow inputs and a Flow-I3D model, also pretrained on Kinetics-400. The ResNet-101 baseline corresponds to a traditional Faster-RCNN object detector system applied to human action classes instead of objects, using just a single frame as input to the model.

Our model achieves a significant improvement of nearly 40% over the best baseline (RGB+Flow), while using just RGB and just one pretrained model instead 3 separate ones. The results suggest that simplicity, coupled with a large pre-training dataset for action recognition, are helpful for action detection. This is reasonable considering that many AVA categories have very few examples, and so overfitting is a serious problem.

4 Conclusion

We have presented an action localization model that aims to densely classify the actions of multiple people in video using the Faster-RCNN framework with spatiotemporal features from an I3D model pretrained on the Kinetics-600 dataset. We show a large improvement over the state-of-the-art on the AVA dataset, but at 21.6% AP, performance is still far from what would be practical for applications. More work remains to be done to understand what are the current modelling problems and how to fix them. In the meanwhile, continuing to grow datasets such as Kinetics should help.

Table 1. Validation set results.

Proposed model	**RGB+Flow [2]**	**RGB-only [2]**	**ResNet-based model [8]**
21.6	15.6	14.5	11.3

References

1. Sigurdsson, G.A., Varol, G., Wang, X., Farhadi, A., Laptev, I., Gupta, A.: Hollywood in homes: Crowdsourcing data collection for activity understanding. In: Proceedings of the European Conference on Computer Vision (ECCV). (2016)
2. Gu, C., Sun, C., Ross, D.A., Vondrick, C., Pantofaru, C., Li, Y., Vijayanarasimhan, S., Toderici, G., Ricco, S., Sukthankar, R., Schmid, C., Malik, J.: Ava: A video dataset of spatio-temporally localized atomic visual actions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018)
3. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision (IJCV) **115**(3) (2015) 211–252
4. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
5. : The kinetics-600 human action video dataset. `http://https://deepmind.com/research/open-source/open-source-datasets/kinetics/` Accessed: 2018-06-10.
6. Carreira, J., Zisserman, A.: 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)
7. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NIPS). (2015)
8. : Ava v2.1 faster rcnn resnet-101 baseline. `https://research.google.com/ava/download.html` Accessed: 2018-06-10.

Two-Stream baseline for AVA dataset: challenge 2018

Gurkirt Singh Fabio Cuzzolin
Oxford Brookes University
gurkirt.singh-2015@brookes.ac.uk

Abstract

In this work, we present a two-stream spatial action detection approach based on a previous state-of-the-art approach to spatial-temporal action detection. We present a two-stream baseline for AVA dataset based on [9]. Our aim is to find out how well a 2D pipeline can perform on AVA and reproduce results presented in [3]. We submitted this approach to ActvityNet challenge 2018 for Task AVA#1 and AVA#2.

1. Methodology

Our approach is based on the work of Singh *et al*. [9], because of its simplicity. There are few changes from Singh *et al*. [9]. Firstly, we replace VGG [6] as base network with Resnet101 [4] as a base network. Secondly, we adopt single-stage object detection approach based on feature pyramid network(FPN) [7] rather than SSD [8]. Lastly, we link detections in time by following the online linking approach described in [9], except temporal labelling. We encourage the reader to read the original paper [9] for more details about the pipeline.

1.1. Inputs

Appearance stream takes 3 channel 600×600 rgb frame as input. Motion stream takes 5 optical frames [1] as input, resulting in a 5x3 channel optical flow image, with $flow_x$ and $flow_y$ being channels and the third channel is the magnitude of $flow_x$ and $flow_y$.

1.2. Resnet101

We use Resnet101 [4] as base network with FPN [7] in [9] pipeline.

1.3. Pre-training on Kinetics Dataset

We trained a Resnet101 model on Kinetics-600 [5] for frame classification task up to 100K iterations. We use this model to start the training on AVA dataset for both flow and appearance stream.

1.4. Loss function

We use standard L1-loss formulation [2] for bounding box regrerssion units. For classification units, we tried binary cross entropy loss with sigmoid activation without much success, so, we switched to softmax loss with online-negative-hard-minging strategy explained in [8]. basically, the loss function is identical to the one described in [8].

1.5. Class Balance

We found that balance across classes is important to achieve given results in this work. We sampled at least 2000 frames for each class and maximum up to 4000 frames per classes. This provides the stabilisation during training and leads to improvements to 5% in final results on the validation set.

1.6. Temporally Consistent Spatial Detection

We link detections in time by following the online linking approach described in [9], resulting in an arbitrary number of bounding boxes linked together called *action-paths*. To generate action-paths, we link detections from 6 frames per second. For evaluation, we need to generate frame-level detections at the rate of one frame per seconds on given timestamps. The detections on every central frame are gathered from action-paths after action-path scores are smoothed with 10 frame sliding window.

2. Experimental Results

We report results for two settings in table 1 for AVA dataset used in ActvityNet challenge 2018 for Task AVA#1 and AVA#2 at intersection-over-union threshold (IOU δ) of 0.5.

Method	Train-Sets	Test-Set	Inputs	$\delta = 0.5$
Gu *et al*. [3]	Train	Val	40-RGB+40-Flow	15.6%
Ours	Train	Val	1-RGB	14.2%
Ours	Train+Val	Test	1-RGB+5-Flow	–%

Table 1: Activity detection performance on validation and testing set. Quantity δ is the spatial Intersection over Union (IoU) threshold.

References

[1] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. *Proc. European Conf. Computer Vision*, 2004. 1

[2] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 1

[3] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. *arXiv preprint arXiv:1705.08421*, 2017. 1

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[5] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1

[6] K. Kulkarni, G. Evangelidis, J. Cech, and R. Horaud. Continuous action recognition based on sequence alignment. *International Journal of Computer Vision*, 112(1):90–114, 2015. 1

[7] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017. 1

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1

[9] G. Singh, S. Saha, M. Sapienza, P. Torr, and F. Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *Proc. Int. Conf. Computer Vision*, 2017. 1

Alibaba-Venus at ActivityNet Challenge 2018 - Task C Trimmed Event Recognition (Moments in Time)

Chen Chen, Xueyong Wei, Xiaowei Zhao and Yang Liu
Alibaba Group, Hangzhou, China
{chenen.cc, xueyong.wxy, zhiquan.zxw, panjun.ly}@alibaba-inc.com

ABSTRACT

In this paper, we present a solution to Moments in Time (MIT) [1] Challenge. Current methods for trimmed video recognition often utilize inflated 3D (I3D) [2] to capture spatial-temporal features. First, we explore off-the-shelf structures like non-local [3], I3D, TRN [4] and their variants. After a plenty of experiments, we find that for MIT, a strong 2D convolution backbone following temporal relation network performs better than I3D network. We then add attention module based on TRN to learn a weight for each relation so that the model can capture the important moment better. We also design uniform sampling over videos and relation restriction policy to further enhance testing performance.

1 INTRODUCTION

Video understanding is a challenging task in computer vision and has significant attention during these years with more and more large-scale video datasets. Compared with image classification, video classification needs to model temporal information and more modalities can be extracted in videos like acoustic, motion, ASR etc. Multi-modalities are mutual complement to each other in many cases.

The recent challenge "Moments in Time Challenge" provides a platform to explore new approaches for short video understanding. The dataset has 339 categories which cover dynamic events unfolding within three seconds. The training/validation/test set has 802264/33900/67800 trimmed videos respectively. The evaluation metric is the average of top1 and top5 accuracy. The organizers provide raw videos and a preprocessed version which normalize videos to resolution 256x256 at 30fps. Participants are allowed to utilize any modality.

2 APPROACH

2.1 Modality Preparation

2.1.1 Visual image preprocessing.

We use preprocessed videos officially provided with resolution 256x256 and 30fps. We extract frames to jpeg format with best quality by using FFmpeg. After checking hundreds of videos, we found a lot of videos have vertical/horizontal black borders like movie style. We remove the black borders by some OpenCV tool and rescale it back to the resolution 256x256. We train/test models by using videos with and without black borders respectively.

2.1.2 Motion Features.

We use traditional TVL1 features which is implemented in OpenCV. It costs 2 weeks to extract motion features for all the MIT video data in a 2 gpu (M40) machine. Horizontal and vertical components are saved as gray image files and we concatenate them to an image with 2 channels during training.

2.1.3 Acoustic Features.

Audio contains a lot of information that helps to classify videos. We extract audio feature by a VGG like acoustic model trained on AudioSet [5] which consists of 632 audio event classes and over 2 million labeled 10-second sound clips. The process is the same as that in Youtube-8M, Google has released the extraction code in tensorflow model github.

2.2 Network Architecture

In this section, we describe all the networks involved.

2.2.1 NetVLAD aggregation with acoustic feature.

Acoustic feature pre-trained on AudioSet for each video has a dimension of 3x128. We use NetVLAD as that in [6] to aggregate acoustic features through time. It learns VLAD encoding followed by fully connect, mixture of experts and context gating.

2.2.2 Non local network.

We use off-the-shelf non local network, and train it with settings of both 32 and 64 sampled frames. The implementation of non-local network decodes video file during training, so we only do experiments on RGB modality.

2.2.3 Inflated 3d model.

I3D and its variant has achieved state of the art performance on datasets like kinetics. It's natural to apply it here in MIT dataset. We use two backbones. One is the origin Inception-V1 pre-trained on kinetics. The other backbone is Inception-V3 inflated ourselves. We inflate the convolution kernel of size 3x3, 5x5, 3x1, 1x3, 7x1, 1x7 into 3x3x3, 3x5x5, 3x3x1, 3x1x3, 3x7x1 and 3x1x7. We drop every other frames, the input video data dimension is 45x224x224. The spatial size is randomly cropped from a scaled video whose shorter side is randomly sampled in [240, 256]. We also randomly flip the whole video horizontally as an augmentation. We use 8 P100 cards to train this model, the batch size is 32. In testing, we use multi-crops (4 corners and center crop together with horizontal flipping) and average to get the final score.

2.2.4 Temporal Relation Network.

TRN achieves advanced performance on three video datasets, Something-Something, Jester, and Charades. These datasets all depends on temporal relational reasoning and MIT has similar character. We employ InceptionV3, InceptionResnetV2 and SENet-154 [7] as backbones for MultiScale TRN and build attention module based on squeeze & excitation module to learn the weighted relations leveraging the global relation distribution instead of simply accumulating them. In testing, we uniformly sample frames over whole video and utilize multi-crops. Also, we analyze the impact of different relations and select them explicitly. We find the following restriction will improve the performance

slightly. In 2-frames relation, the minimum relation sampling distance should be 2. In 3-frames relation, the distance should be in range [2, 3]. In 4-frames relation, the distance should be in range [2, 4].

We also try to combine I3D and TRN together. First, we split the video frames into 5 segments, each segment has 18 frames. Then, we apply 3D convolution model to each segment and will get a representation vector. Finally, TRN builds the relationship between the 5 segments. We apply this model to both RGB and Flow with Inception-V1 backbone, and we rescale the input resolution to 184x184 to reduce the complexity. The batch size is 64.

2.3 Ensemble

We use class-wise weighted ensemble. We calculate average precisions for each model and then normalize the weight for each class through models. After this operation, the ensemble model will take the different ability for each single model on each class into consideration. For example, when dealing with "clapping", acoustic model will have a predominant weight. In the final submission, we ensemble 13 models and the result is showed in next section**.**

3 EXPERIMENT

3.1 Experiment Results

We test on 3 modalities with different models. The input resolution is 224x224 except the case in I3D (184x184). We use multi-crop testing in all cases. Details are listed in Table 1.

We notice that in MIT dataset, 2D convolution following temporal relations works better than 3D convolution networks including I3D, non-local network and their variants. In temporal relation testing, uniform sampling policy over the whole videos works well. We use 8 segments here (90 frames) and average the score of 11 uniformly sampled clips. With the test enhancement, the baseline performance greatly improves from 28.61/54.65 to 29.67/55.74. The backbone is also of great importance, we compare InceptionV3, Inception Resnet V2, and SENet-154. SENet-154 is the best backbone in cost of high complexity and long training time. We spend 6 days to train SENet-154 TRN model. Actually, we also try Nasnet but fail to get a good result due to small batch size (only 8). Attentional TRN and restricting distance between consecutive sampled relations also help which means that more effective relations are selected. . The best single RGB model (32.21/59.05) is attentional temporal relation network with backbone senet154, and test using uniform sampling, multi-crop and manually restricted relation policy. Our acoustic model using AudioSet pre-trained features following netVLAD aggregation layer is better than baseline SoundNet metric. The final class-wise weighted ensemble consists of 13 models listed in the table which achieves top1/top5 (%) 36.23/64.56 on validation set. Since the ensemble weights depend on validation set, it makes more sense to check it on test set. We verify it on test server and find the weighted ensemble is better than average ensemble by a considerable margin about 0.2.

Table 1: *Experimental results on Validation Set (model with * is used in ensemble. Test enhancement means uniform sampling and multi-crop. ATRN is attentional temporal relation network)*

Models	Modality	Backbone	Top1/Top5
Non-local 32 frames *	RGB	Resnet50	27.04/54.02
Non-local 64 frames	RGB	Resnet50	26.44/53.11
I3D *	RGB	InceptionV3	27.62/53.89
I3D + TRN *	RGB	InceptionV1	28.25/54.83
I3D + TRN *	Flow	InceptionV1	18.00/39.17
TRN without test enhancement	RGB	InceptionV3	28.61/54.65
TRN without test enhancement, with relation restricted	RGB	InceptionV3	28.82/54.72
TRN *	RGB	InceptionV3	29.67/55.74
TRN black borders removed *	RGB	InceptionV3	29.59/55.86
TRN	Flow	InceptionV3	16.55/37.04
TRN *	RGB	InResnetV2	29.33/56.57
TRN (without test enhancement)	RGB	SENet-154	31.10/58.08
TRN	RGB	SENet-154	31.89/58.82
ATRN *	RGB	SENet-154	32.09/58.91
ATRN black borders removed *	RGB	SENet-154	31.97/59.26
ATRN relation restricted *	RGB	SENet-154	32.21/59.05
ATRN 512 dim bottleneck	RGB	SENet-154	31.63/58.92
ATRN vertical flipped input *	RGB	SENet-154	31.32/58.84
ATRN	Flow	SENet-154	17.92/39.60
netVLAD 64clusters *	Audio	VGG	9.00/19.51
netVLAD 128clusters *	Audio	VGG	8.90/20.23
13 models ensemble	None	None	36.23/64.56

4 CONCLUSIONS

In summary, we have tried off-the-shelf models for video recognition. To our surprise, temporal relation on top of 2d convolution works better than inflated 3d models in Moments in Time. It may be the case that MIT dataset is more complicated than traditional trimmed activity datasets like kinetics in terms of 1) Events are not limited to human related, more objects and scenes are involved, deep 2d convolution networks have stronger representation ability. 2) Big inner-class difference, for example, "fencing" has two totally different meanings which makes it harder to train 3d models. Furthermore, based on TRN, we add

attention module on relations, try stronger backbone and design effective uniform sampling test which greatly improves the performance.

REFERENCES

[1] Mathew Monfort, Bolei Zhou, Sarah Adel Bargal, Alex Andonian, Tom Yan, Kandan Ramakrishnan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, Aude Oliva. Moments in Time Dataset: one million videos for event understanding. 2018.

[2] Xiaolong Wang and Ross Girshick and Abhinav Gupta and Kaiming He. Non-local Neural Networks. In CVPR 2018.

[3] Joao Carreira and Andrew Zisserman. Quo Vaids, Action Recognition? A new model and the kinetics dataset. In CVPR 2017.

[4] B. Zhou, A. Andonian, and A. Torralba. Temporal Relational Reasoning in Videos. arXiv:1711.08496, 2017.

[5] Jort F. Gemmeke and Daniel P.W. Ellis and Dylan Freedman and Aren Jansen and Wade Lawrence and R. Channing Moore and Manoj Plakal and Marvin Ritter. Audio Set: An ontology and human-labeled dataset for audio events. In ICASSP 2017.

[6] Antoine Miech, Ivan Laptev and Josef Sivic. Learnable pooling with context gating for video classification. arXiv preprint arXiv:1706:06905, 2017.

[7] Jie Hu, Li Shen, Gang Sun. Squeeze and Excitation Networks. arXiv preprint arXiv:1709.01507, 2017.

Trimmed Event Recognition (Moments in Time): Submission to ActivityNet Challenge 2018

Dongyang Cai
caidongyang_sx@qiyi.com

Abstract

In this paper, a brief description is provided of the method used for the task of trimmed event recognition (Moments in Time). A set of TRN models were used to train video classification models for the 200 action categories of the Moment in Time Mini Database, and the P3D feature is also used to further enhance the model diversity, finally we propose a simple yet effective method to combine different modalities together for action prediction.

1. Introduction

The Moments in Time Dataset [1], is a large-scale human-annotated collection of one million short videos corresponding to dynamic events unfolding within three seconds, each video is tagged with one action or activity label. Modeling the spatial-audio-temporal dynamics even for actions occurring in 3 second videos poses many challenges: meaningful events do not include only people, but also objects, animals, and natural phenomena; visual and auditory events can be symmetrical or not in time. Here, with limited computation resources, we will use the Moments in Time Mini dataset, which is a subset of Moments in Time with 100k videos provided in the training set and involves 200 action categories, for model training and action prediction. As we note the temporal relational reasoning is very important for this task [2], we train a set of Temporal Relation Network (TRN) models firstly, also the recently proposed P3D method [3] is found useful to enhance the model diversity, finally we propose a simple yet effective method to combine those methods above to identify the event labels depicted in a 3 second video.

2.Method Description

2.1 TRN

Temporal relational reasoning is critical for activity recognition, forming the building blocks for describing the steps of an event. A single activity can consist of several temporal relations at both short-term and long-term timescales, the ability to model such relations is very important for activity recognition. The Temporal Relation Network (TRN) proposed by Bolei Zhou et al [2] is designed to learn and reason about temporal dependencies between video frames at multiple time scales. It is an effective and interpretable network which is able to learn intuitive and interpretable visual common sense knowledge in videos. The networks used for extracting image features is very important for visual recognition tasks, here we use an 8 segment multi-scale TRN with an inceptionV3 base and Inception with Batch Normalization (BN-Inception) base separately, and then train the TRN-equipped network with different data augmentation scheme with each base network, we found training a set of TRN networks with fusing them together bring action prediction improvement.

2.2 P3D

Pseudo-3D Residual Net (P3D ResNet) architecture proposed by Qiu, Zhaofan et al [3], aims to learn spatio-temporal video representation in deep networks, it simplifies 3D convolutions with 2D filters on spatial dimension plus 1D temporal connections. Experiments on five datasets in the context of video action recognition, action similarity and scene recognition also demonstrate the effectiveness and generalization of spatio-temporal video representation produced by P3D ResNet. Here, to enhance our model diversity, we adopted P3D ResNet to learn feature representation of the Moments in Time Mini Dataset, and utilized the learned features for this video classification task.

2.3 Weak classifiers

In this part, we use the idea of AdaBoost[4] to generate our own classifier. First, we rank the 200 classes according to their accuracy on the validation dataset. Then, we choose 50 classes with the lowest accuracy and increase their sample weight for future training. What is more, we calculate all the training data and get the confusion matrix of the 200 classes, for those confusing categories, we trained weak classifiers to classify them especially. For all the training samples which was classified wrong before, their weight will be also increased for another weak classifier. By this method, we achieved improvement on the accuracy of testing data.

2.4 Ensemble

We propose a simple yet effective model ensemble method to enhance the action prediction ability of our final classification model. Firstly, we will calculate the classification accuracy of each model referred above on the validation dataset, then we assign a weight to each model according to its classification accuracy, model with high accuracy embracing a higher weight. Given a test video, we firstly predict its top 5 labels with each model. We will give a label weight to a predicted label according to its number of occurrences across models. For one model, we will multiply confidence score of each predicted label with the model weight referred above, and then add up the resulting value of the same predicted label across models with its label weight. Finally, we will rank the predicted labels according to their label scores above in descending order and get the top 5 labels for action prediction of the test video. Our experimental results below will show the effectiveness of our method.

3.Experimental Results

The Moments in Time Mini Dataset contains 100000 training videos, 10000 validation videos and 20000 testing videos. Each video is in one of 200 categories. Table 1 summarizes our results on the Moments in Time Mini validation dataset.

Model	Top-1 Accuracy (%)	Top-5 Accuracy (%)
TRN	26.1%	48.5%
P3D	14.7%	33.4%
Weak classifier	28.3%	52.2%
TRN+P3D+ Weak classifier	31.7%	56,9%

Table 1. Moments in Time Mini validation results.

4.Conclusion

The recently proposed TRN method is effective for recognizing daily activities with learning intuitive and interpretable visual common sense knowledge in videos. Also, the P3D feature is used to enhance model diversity. We utilize those methods and propose a simple yet effective method to combine different modalities together for action prediction of the Moment in Time Mini Dataset, our experimental results show its effectiveness.

5.Acknowledgement

This work was finished during an internship work in IQIYI, many thanks to Jie Liu, Tao Wang, and Xiaoning Liu for helpful comments and discussion.

References

[1] Monfort, Mathew, et al. "Moments in Time Dataset: one million videos for event understanding." *arXiv preprint arXiv:1801.03150* (2018).

[2] Zhou, Bolei, Alex Andonian, and Antonio Torralba. "Temporal Relational Reasoning in Videos." *arXiv preprint arXiv:1711.08496* (2017).

[3] Qiu, Zhaofan, Ting Yao, and Tao Mei. "Learning spatio-temporal representation with pseudo-3d residual networks." *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017.

[4] Freund, Yoav, and Robert E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting." *Journal of computer and system sciences* 55.1 (1997): 119-139.

CMU-AML Submission to Moments in Time Challenge 2018

Po Yao Huang
School of Computer Science
Carnegie Mellon University
poyaoh@andrew.cmu.edu

Xiaojun Chang
SCS, Carnegie Mellon University
Hangzhou Anmeilong Intelligence Co., Ltd.
cxj273@gmail.com

Alexander G. Hauptmann
School of Computer Science
Carnegie Melon University
alex@cs.cmu.edu

Abstract

*In this report, we describe our solution for Moments in Time Challenge 2018. We employed both visual and audio features in the submission. For visual features, we utilize the preprocessed RGB and optical flow data for training or fine-tuning 2D (*e.g. Temporal Segment Network (TSN) *and 3D (*e.g. *Inflated 3D ConvNets (I3D). For audio features, we use raw waveforms as the input modality and fine-tune the feature extracted from the last pooling layer of SoundNet. We achieve 31.56% in terms of Top-1 accuracy and 59.75% in terms of Top-5 accuracy on the validation set.*

1. Introduction

The last decodes have witnessed the success of deep learning in image understanding tasks, *i.e.* classification [8], segmentation [11], and *etc*. Researchers have demonstrated the superiority of state-of-the-art Convolutional Neural Networks (CNN) [9, 3] against traditional algorithms with hand-crafted features. Inspired by the progress, CNNs have been widely employed to improve the performance of video understanding tasks. Compared to image understanding tasks, temporal information of videos can boost the performance of video classification. Additionally, auditory soundtracks provides an additional clue for video analysis.

We cannot obtain discriminate models without large-scale labeled dataset, such as ImageNet [2] and ActivityNet [4]. Recently, the MIT-IBM Watson AI Lab has released a large-scale Moments dataset [7] to help AI systems recognize and understand actions and events in videos. This dataset contains a collection of one million labeled 3 second videos, involving people, animals, objects or natural phenomena, that capture the gist of a dynamic scene. The Moments in Time Challenge 2018 is based on this dataset.

2. Our Approach

In this section, we describe the features and models we used for the challenge. We use the standard split defined in the original paper where 802,244 training video and 37,800 validation video are available.

2.1. Features

Visual Features: All the videos are first resized to 340×256 under 30 fps. We rescale raw RGB values into $\big[-1, 1\big]$. We also computed optical flow with the TVL1 algorithm and rescale the value into $\big[-1, 1\big]$.

We utilize the preprocessed RGB and optical flow data for training or fine-tuning 2D(e.g. TSN) and 3D(e.g. I3D [1]) models. In order to fit the relatively shorter but constant period for the targeted Moments in Time dataset, we use dilated frames (with a fixed M network input size with step size $\lfloor N/(M-1) \rfloor$, where N is the frame size) as inputs instead of consecutive frames.

To leverage the knowledge from other dataset, we also use existing models pre-trained on external large datasets such as Kinetics. In practice, we use the RGB and Optical flow models pre-trained on ImageNet and as the feature extractors to extract the features reside in last pooling layer as new video representation. Specifically, we sample the center frames of a video as the input and store a $(7, 1024)$ vector for each video. This approach is equivalent to fine-tuning the layer above last pooling layer of a model.

Audio Features: We average the two channels and re-

sample the audio into 22,050 Hz .wav files. For videos without audio channel, we fill a 3-second silent audio for them. We extract the conv7 layer of the soundnet model, which is pretrained over 2,000,000 unlabeled videos. Then we feed the features into a 10-layer DenseNet [5] with the output layer changed to predict moment categories.

2.2. Models

Fine-tune all models For this challenge, we fine-tune 2D(spatial) and 3D(spatial-temporal) models with additional layers.

For 2D models, built upon TSN, we add an additional cutout layer. Each sampled (340×256) frame will be randomly cut out with a (90×90) region. We also tried other augmentation techniques such as mix-up but found cut-out is the most feasible one.

For 3D models, we use I3D models with ResNet 50 (R50) as its backbone. In addition to cutout augmentation layer we add an addition non-local layer to capture the interaction between spatial-temporal units. As in [10], we add 10 non-local blocks to R50.

Considering the size of the target dataset, we choose to use network with 8 frame inputs. Empirically we found that ImageNet pre-trained I3D model with non-local networks are prone to overfit for Moment in time dataset in comparison to 3D models without non-local networks. A better choice is to use ImageNet-Kinetics pre-trained models where we observed preferred behaviors.

Fine-tune last models As described before, we utilize the ImageNet-Kinectics pre-trained I3D models as the spatial-temporal feature extractor. Take (7,1024) features as the input, we randomly sample and average 2 frames then feed to the classification network.

For the classification, we choose the mixture-of-residual expert (MoRE) network as proposed in[6] with 4 experts with two-layer network (each layer with 2048 neurons) with residual links as the classification model for RGB and optical flow features. We found that with pre-extracted feature the network are prone to overfit and therefore apply a high dropout rate (0.8) and append an input batch-normalization later to train the model.

2.3. Training and Inference Details

For training finetune-all models, we use 3-Titan XP GPUs with batch size 24 and standard momentum SGD. With limited resource and time we train each 3D models for 20 epochs. The learning rate is set 0.005 and decayed by 0.1 at 10, 16, 18 epochs respectively. It take roughly 4 days to train a model. For inferencing, we sample 5 inputs (each with 8 frams) from a video then mean-pool the predictions as the video-level prediction.

Training finetune-last models are comparably cost-effective. We use one Titan XP GPU with batch size 512 and Adam optimizer and train for 80 epochs. The learning rate is set 0.001 and decayed by 0.1 at 30, 50, 70 epochs. At testing phase, we loop every frame of a video and generate frame-wise prediction then mean-pool the results.

2.4. Evaluation metric

Following the stand of the Moments challenge, we employ top-k accuracy as the evaluation metric. For each video, the system will generate k labels $l_j, j = 1 \dots k$. The ground truth label for the video is g. The error of the algorithm for that video would be:

$$e = \min_j d(l_j, g), \tag{1}$$

where $d(x, y) = 0$ if $x = y$ and 1 otherwise. The overall error score for an algorithm is the average error over all videos. We use $k = 1$ and $k = 5$.

2.5. Fusion

In this report, we fuse multiple features for video classification. We learn the optimal weights for different features on the validation set. Then we apply these weights on the testing set, and get the results for the final submission.

3. Results

In this section, we first evaluate the performance of the individual feature on the validation set. The performance are shown in Table 1. From the experimental results we can observe that I3D with non-local network have better performance than I3D without non-local network. For example, the performance of I3D with NLN improves the performance of I3D without NLN from 28.96 to 29.48 in terms of top-1 accuracy. However, we observe that I3D with NLN is prone to overfit. For example, when we use the ImageNet pretrained I3D with NLN, we get only 25.94 in terms of top-1 accuracy. This demonstrates the necessity of using ImageNet and Kinects pre-trained network to avoid overfitting.

After that, we learn the optimal weights for individual features on the validation set. The weights of utilized features are shown in Table 2. With these weights, we have obtained 31.56 in terms of Top-1 accuracy and 59.75 in terms of Top-5 accuracy on the validation set, respectively.

4. Conclusion

In this report, we have presented our solution to the Moments in Time Challenge 2018. We found that Inflated 3D ConvNets (I3D) with non-local networks has the best single model performance. However, we found that ImageNet pre-trained I3D model with non-local networks are prone to overfit for the challenge dataset. Hence, we choose to use ImageNet-Kinects pretrained models where we observed preferred performances.

Table 1. Performance evaluation of different features on the validation set.

Type	Model	Pre-Trained	val Top-1	val Top-5	Final Fusion
TSN-Spatial	Baseline	I+K	24.07	48.98	T
TRN-Multiscale	Baseline	I+K	21.02	43.27	T
Audio	SoundNet	U	6.83	15.41	T
2D-RGB	Last-RGB	I+K	19.84	41.75	T
2D-OF	Last-Optical Flow	I+K	18.49	39.64	T
3D-RGB	All-vanilla-I3D (8 frames)	I	28.12	56.04	F
3D-RGB	All-I3D (8 frames)	I	28.96	56.45	T
3D-RGB	All-I3D-NLN (8 frames)	I	25.94	52.60	F
3D-RGB	All-I3D-NLN (8 frames)	I+K	25.75	53.31	F
3D-RGB	All-I3D-NLN (16 frames)	I+K	29.48	57.37	T

Table 2. Weights of different features and the fusion results on the validation set.

Type	Model	Weights
TSN-Spatial	Baseline	2
TRN-Multiscale	Baseline	2
Audio	SoundNet	1
2D-RGB	Last-RGB	2
2D-OF	Last-Optical Flow	1
3D-RGB	All-I3D (8 frames)	8
3D-RGB	All-I3D-NLN (8 frames)	3
3D-RGB	All-I3D-NLN (16 frames)	12
Fusion Result	Top-1: 31.56	Top-5: 59.75

References

[1] J. Carreira and A. Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4724–4733, 2017. 1

[2] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255, 2009. 1

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016. 1

[4] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 961–970, 2015. 1

[5] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2261–2269, 2017. 2

[6] P.-Y. Huang, Y. Yuan, Z. Lan, L. Jiang, and A. G. Hauptmann. Video representation learning and latent concept mining for large-scale multi-label video classification. *CoRR*, abs/1707.01408, 2017. 2

[7] M. Monfort, B. Zhou, S. A. Bargal, A. Andonian, T. Yan, K. Ramakrishnan, L. M. Brown, Q. Fan, D. Gutfreund, C. Vondrick, and A. Oliva. Moments in time dataset: one million videos for event understanding. *CoRR*, abs/1801.03150, 2018. 1

[8] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 1

[9] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826, 2016. 1

[10] X. Wang, R. B. Girshick, A. Gupta, and K. He. Non-local neural networks. *CoRR*, abs/1711.07971, 2017. 2

[11] Y. Zhang, Z. Qiu, T. Yao, D. Liu, and T. Mei. Fully convolutional adaptation networks for semantic segmentation. *CoRR*, abs/1804.08286, 2018. 1

Team DEEP-HRI Moments in Time Challenge 2018 Technical Report

Chao Li, Zhi Hou, Jiaxu Chen, Yingjia Bu, Jiqiang Zhou, Qiaoyong Zhong, Di Xie and Shiliang Pu
Hikvision Research Institute

Abstract

Video-based action recognition is challenging as spatial and temporal reasonings are involved jointly. We propose a novel multi-view convolutional architecture, which performs 2D convolution along three orthogonal views of volumetric video data. With weight sharing, it is capable of encoding spatio-temporal feature of video clips efficiently, and achieves superior performance over state-of-the-art spatio-temporal feature learning architectures. Furthermore, we also explore the auditory modality, which is complementary to visual clues. Our final submission to the Moments in Time challenge 2018 is an ensemble of several visual RGB and audio models, achieving a top-1 accuracy of 38.7% and top-5 66.9% on the validation set.

1 Introduction

The task of video-based action recognition requires proper modelling of both visual appearance and motion pattern. Recently, a significant effort has been devoted to spatio-temporal feature learning from video clips. Since the success of convolutional neural networks (CNN) in 2D image recognition [1], 3D convolution is a natural adaption for volumetric video data [2]. However, in C3D [2], significantly more (e.g. 2×) parameters than its 2D counterpart are introduced, which makes the model difficult to train and prone to overfitting. This issue is particularly critical when the training data size is limited. P3D [3] and (2+1)D [4] attempted to address the issue by decomposing a 3D convolution into a 2D convolution along the spatial dimension and a 1D convolution along the temporal dimension. We argue that the "unequal" treatment of spatial and temporal features is undesirable. On the contrary, we propose Multi-View CNN (MV-CNN), which performs feature extraction along the spatial and temporal dimensions in a consistent way. The details of MV-CNN are described in Section 2.1. To further improve the overall accuracy, we train non-local networks [5] for model ensemble.

2 Method

In this work, we explore multiple modalities for categorizing the action occurring in a video. Our visual RGB model is based on an ensemble of the proposed MV-CNN and other state-of-the-art spatio-temporal feature learning models. We also tried optical flow, but found that it do not contribute to the final accuracy after ensemble. However, we do exploit audio-based action recognition, which is complementary to visual signal.

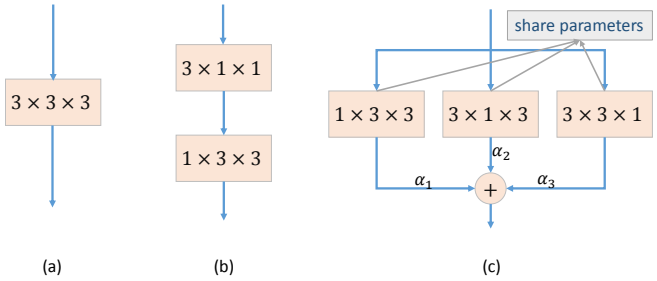


Figure 1: Comparison of MV-CNN to common spatio-temporal feature learning architectures. (a) C3D. (b) (1+2)D. (c) the proposed MV-CNN.

2.1 Multi-View CNN

A video clip can be represented as a 3D array of dimension $T \times H \times W$, where T, H and W are number of frames, frame height and frame width respectively. Taking kernel size of 3 as an example, Figure 1 compares the proposed MV-CNN to common convolutional architectures. In C3D, a 3D $3 \times 3 \times 3$ convolution is utilized to extract spatial (H and W) and temporal (T) features jointly. In the (1+2)D configuration, a 1D $3 \times 1 \times 1$ convolution is utilized to aggregate temporal feature, followed by a 2D $1 \times 3 \times 3$ convolution for spatial feature. While in the proposed MV-CNN, we perform 2D 3×3 convolutions along three views of the $T \times H \times W$ volumetric data, i.e. $T \times H$, $T \times W$ and $H \times W$ separately. The three orthogonal views are conceptually similar to the three anatomical planes of human body, namely sagittal, coronal and transverse. Notably, the parameters of the three-view convolutions are shared, such that the number of parameters is kept the same as single-view 2D convolution. The three resulting feature maps are further aggregated with weighted average. The weights are also learned during training in an end-to-end manner. To facilitate training, we initialize the 2D convolutional kernels with a ImageNet [6] pretrained model.

For each model, to obtain better generalization on the test

set, the Stochastic Weight Averaging (SWA) scheme [7] is adopted. Several model variants of the same network are trained with cycle learning rate and subsequently form an ensemble.

2.2 Auditory Modality

Complementary to visual signal, sound conveys important information for action recognition. Therefore, in our method, audio streams extracted from videos are exploited for the task of action categorization. In audio processing, log-mel spectrum is a powerful hand-tuned feature, exhibiting locality in both time and frequency domains [8]. In ResNet-34 [9], the 2D log-mel feature is cast into an image, and a 34-layer ResNet is applied for audio classification. While M34-res [10] and EnvNet [11] attempted to learn semantic feature from the 1D raw audio waveforms in an end-to-end way. We train the three state-of-the-art models on the Moments in Time dataset. Notably, we adapt EnvNet [11] with residual connections, and henceforth refer to the variant as EnvNet+ResNet.

3 Experiments

The Moments in Time dataset [12] contains 802245 training videos and 39900 validation videos. Excluding the videos without audio track, the auditory dataset contains 450k training segments and 20k validation segments. In total 339 action categories are annotated. In all experiments, our models are trained on the provided Moments in Time training data only. Apart from ImageNet, no other video datasets are used for pretraining.

For the visual RGB model, during training, we select 64 continuous frames from a video and then sample 8 frames by dropping the 7 frames in between. The spatial size is 224×224 pixels, randomly cropped from a scaled video whose shorter side is randomly sampled between 256 and 320 pixels. During inference, following [5] we perform spatially fully convolutional inference on videos whose shorter side is rescaled to 256 pixels. While for the temporal domain, we sample 6 clips evenly from a full-length video and compute softmax scores on them individually. The final prediction is the averaged softmax scores of all clips.

In this work, we use ResNet-101 [13], Inception-v4 and Inception-ResNet-v2 [14] as the backbone models, which are pretrained on ImageNet. The proposed MV-CNN along with C3D and non-local (NL) models are trained to form an ensemble. The top-1 and top-5 accuracies of individual models as well as their ensemble are shown in Table 1. For Inception-ResNet-v2, MV-CNN obtains 35.6% top-1 and 63.6% top-5 accuracy, leading to 0.5% and 0.3% accuracy gain compared with the C3D baseline. It is worth noting that with MV-CNN, more significant performance gain can be obtained on smaller sized datasets like UCF-101 [15]. On large-scale datasets like Moments in Time, the performance gain saturates, which is reasonable as increasing data size could be more effective than algorithmic innovations. With an ensemble of visual RGB models alone, we achieve a top-1 accuracy of 37.7% and top-5 65.9%.

For the training of audio models, all the sound data are downsampled to a frequency of 16kHz. For M34-res, we train two models for audio section lengths of 1s and 3s separately. Then their scores are averaged. This multi-scale training and inference scheme improves the robustness against audio length. The performances of the three audio models are summarized in Table 1. With an ensemble of audio models alone, we obtain 17.6% top-1 and 31.1% top-5 accuracy. With an ensemble of visual RGB and audio models, we achieve a top-1 accuracy of 38.7% and top-5 66.9%.

Table 1: Accuracy on the validation set of the Momements in Time dataset. Performances of both individual visual and audio models and their ensemble are shown.

Model	Modality	Accuracy (%)	
		top1	top5
ResNet-101-C3D	RGB	33.6	61.2
ResNet-101-NL	RGB	32.8	60.8
Inception-v4-C3D	RGB	34.3	62.0
Inception-ResNet-v2-C3D	RGB	35.1	63.3
Inception-ResNet-v2-NL	RGB	34.8	63.3
Inception-ResNet-v2-MV	RGB	35.6	63.6
Ensemble	RGB	37.7	65.9
ResNet-34	Audio	13.8	23.6
M34-res	Audio	14.8	27.4
EnvNet+ResNet	Audio	13.2	25.9
Ensemble	Audio	17.6	31.1
Ensemble	RGB+Audio	38.7	66.9

4 Conclusions

In our submission to the Moments in Time challenge 2018, we explore multiple modalities for the task of video-based action recognition. Particularly, we propose a novel multi-view convolutional architecture, which achieves superior performance over the C3D baseline with significantly less number of parameters. A more thorough and systematic evaluation of the architecture is left for future work.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[2] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri, "Learning spatiotemporal features with 3d convolutional networks," pp. 4489–4497, 2014.

[3] Zhaofan Qiu, Ting Yao, and Tao Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," pp. 5534–5542, 2017.

[4] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann Lecun, and Manohar Paluri, "A closer look at spatiotemporal convolutions for action recognition," 2017.

[5] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He, "Non-local neural networks," 2017.

[6] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[7] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson, "Averaging weights leads to wider optima and better generalization," 2018.

[8] Ossama Abdel-Hamid, Abdel Rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[9] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, and Bryan Seybold, "Cnn architectures for large-scale audio classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 131–135.

[10] Wei Dai, Chia Dai, Shuhui Qu, Juncheng Li, and Samarjit Das, "Very deep convolutional neural networks for raw waveforms," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 421–425.

[11] Yuji Tokozume and Tatsuya Harada, "Learning environmental sounds with end-to-end convolutional neural network," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 2721–2725.

[12] Mathew Monfort, Bolei Zhou, Sarah Adel Bargal, Tom Yan, Alex Andonian, Kandan Ramakrishnan, Lisa Brown, Quanfu Fan, Dan Gutfruend, Carl Vondrick, et al., "Moments in time dataset: one million videos for event understanding," .

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," pp. 770–778, 2015.

[14] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," 2016.

[15] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

Method Summary

Team Name: HERO_AN

Inspired by the effectiveness of sparse sample, we combine the advantages of TRN and TSN to discriminate the action in this action challenge. Specifically, TRN has the ability of reasoning the temporal relation of video frames. However, the sparse sample strategy used in TRN would discard essential motion information if it samples in major interval or minor interval coincidentally. Besides, due to the high computation complexity, the efficiency would degrade dramatically. For TSN, although it achieves desirable performance, it just averages the predictions of three segments, without the reasoning ability. Different from TRN, TSN could capture frame information at every segments.

Based on above analyses, we propose a comprehensive sparse sample strategy. We divide video frames into four segments averagely. Then, we randomly sample four frames in each segment, which means we get 16 frames from a video. The resulted 16 frames are fed into ResNet-34 pretrained on ImageNet to extract the feature representation of each frame. These representations are concatenated into a feature vector and classified by two fully-connected layers.

That is all we used in the submission.

AIST Submission to ActivityNet Challenge 2018

Kensho Hara, Hirokatsu Kataoka, Yutaka Satoh
National Institute of Advanced Industrial Science and Technology (AIST)
Tsukuba, Ibaraki, Japan
{kensho.hara, hirokatsu.kataoka, yu.satou}@aist.go.jp

Abstract

In this paper, we introduce our method for ActivityNet Challenge 2018 Task C (Moments in Time). We used a 3D convolutional neural network (CNN) pretrained on Kinetics-400, and finetuned it on Moments in Time. We experimentally evaluated the performance of our method.

1. Introduction

We focus on the trimmed event recognition task (Task C) in ActivityNet Challenge 2018. We use a 3D convolutional neural network (CNN) pretrained on Kinetics-400 [1] to recognize events. In our previous work [2], we trained various 3D architectures on Kinetics-400 and released them[1]. We use the pretrained ResNeXt-101 for the recognition on the Moments in Time dataset.

2. Implementation

We use stochastic gradient descent with momentum to train the network and randomly generate training samples from videos in training data in order to perform data augmentation. First, we select a temporal position in a video by uniform sampling in order to generate a training sample. A 16-frame clip is then generated around the selected temporal position. If the video is shorter than 16 frames, then we loop it as many times as necessary. Next, we randomly select a spatial position from the 4 corners or the center. In addition to the spatial position, we also select a spatial scale of the sample in order to perform multi-scale cropping. The scale is selected from $\left\{1, \frac{1}{2^{1/4}}, \frac{1}{\sqrt{2}}, \frac{1}{2^{3/4}}, \frac{1}{2}\right\}$. Scale 1 means that the sample width and height are the same as the short side length of the frame, and scale 0.5 means that the sample is half the size of the short side length. The sample aspect ratio is 1 and the sample is spatio-temporally cropped at the positions, scale, and aspect ratio. We spatially resize the sample at 112×112 pixels. The size of each sample is 3 channels $\times$ 16 frames $\times$ 112 pixels $\times$ 112 pixels, and each sample is horizontally flipped with 50% probability. We also perform mean subtraction, which means that we subtract the mean values of ActivityNet from the sample for each color channel. All generated samples retain the same class labels as their original videos.

In our training, we use cross-entropy losses and back-propagate their gradients. The training parameters include a weight decay of 0.001 and 0.9 for momentum. When finetuning the network, we start from learning rate 0.01, and divide it by 10 after the validation loss saturates.

[1] https://github.com/kenshohara/3D-ResNets-PyTorch

Table 1: Accuracies on the Moments in Time validation set. *Average* is averaged accuracy over *Top-1* and *Top-5*.

Method	Top-1	Top-5	Average
ResNeXt-101	28.5	53.9	41.2

3. Experiments

Table 1 shows the results on the Moments in Time validation set. We recognized the videos of the test set using this network, and submitted the results.

4. Conclusion

In this paper, we described the submission for the Task C of ActivityNet Challenge 2018.

References

[1] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The Kinetics human action video dataset," *arXiv preprint*, vol. arXiv:1705.06950, 2017.

[2] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and Imagenet?," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Multi-Modal Fusion for Moment in Time Video Classification

Hu-Cheng Lee* Sebastian Agethen* Chih-Yu Lin Hsin-Yu Hsu
Pin-Chun Hsu Zhe-Yu Liu Hsin-Li Chu Winston Hsu
National Taiwan University
{r05922174, d01944015, r05922109, r06922087,b03901023}@ntu.edu.tw
j2325138@gmail.com, {b05505004,whsu}@ntu.edu.tw

Abstract

Action recognition in videos remains a challenging problem in the machine learning community. Particularly challenging is the differing degree of intra-class variation between actions: While background information is enough to distinguish certain classes, many others are abstract and require fine-grained knowledge for discrimination. To approach this problem, in this work we evaluate different modalities on the recently published Moments in Time dataset, a collection of one million videos of short length.

1. Introduction

There are hundreds of thousands of activities occurring around us in our daily life. Most of these activities are not only restricted to one person or a single motion, but involve many types of actors in different environments, at different scales, and with many different modalities. If we want to solve problems that are relevant to our real world, it is necessary to develop models that scale to the level of complexity and abstract reasoning that a human processes on a daily basis. We propose a new approach to tackle these challenges. To evaluate our work, we use the *Moments in Time Dataset* [7].

Moments in Time Dataset is a large-scale human-annotated collection of one million short videos corresponding to dynamic events unfolding within three seconds and has a significant intra-class variation among the categories.

The dataset poses a number of challenges that we need to conquer. First, the videos have a diverse set of actors, including people, objects, animals and natural phenomena. Second, the recognition may depend on the social context of ownership and the type of place. For example, picking up an object, and carrying it away while running can be categorized as stealing, saving or delivering, depending on the ownership of the object or the location where the action occurs. Third, the temporal aspect: the same set of frames in a reverse order can actually depict a different action, consider for example `opening` vs. `closing`. Since we want to build a true video understanding model, we need to be able to recognize events across agent classes. In other words, it is necessary to recognize these transformations in a way that will allow them to discriminate between different actions, yet generalize to other agents and settings within the same action.

In this work, we investigate the fusion of features of different modalities. In Section 2, we outline each modality. In Section 3, we discuss the fusion methods, and provide preliminary results on the *Moments in Time Mini* validation set. Finally, Section 4 discusses analytic insights into the dataset based on a simple RGB baseline.

2. Methodology

We investigated a number of modalities of interest for action recognition. We first discuss each modality, and then examine both early and late fusion of these modalities.

2.1. RGB and optical flow

In action recognition, we consider two essential visual concepts: appearances and motions. Most action recognition work uses RGB frame and optical flow as the visual representation respectively. In order to fully utilize the visual contents from videos, a practical approach, introduced by [9], models short temporal snapshots of videos by averaging the predictions from a single RGB frame and a stack of 10 externally computed optical flow frames, which is also known as *Two-stream ConvNets* method.

Temporal Segment Networks There have been many improvements over the basic two-stream architecture, and one of the most well-known method is *Temporal Segment Networks* (TSN) [11]. Instead of working on single frames or frame stacks, TSN operate on a sequence of short snippets

*Equal contribution

sparsely sampled from the entire video. Each snippet in this sequence will generate its own preliminary prediction of the action classes. Then a consensus among the snippets will be derived as the video-level prediction. We use the same settings as the original work for our prediction.

Temporal Relational Reasoning *Temporal Relational Reasoning Network* (TRN) [12] can learn and discover possible temporal relations at multiple time scales. TRN is a general and extensible module that can be used in a plug-and-play fashion with any existing CNN architecture. We also use the same settings for our prediction.

2.2. Sound

Sound is a valueable modality in action recognition. It can not only complement visual observations, but also help add information where vision is not available, i.e., unseen or occluded surroundings.

Feature extraction We use two pretrained models for audio feature extraction: *Audio Event Net* (AENet) [10] and *VGGish* pretrained on AudioSet[4].

To ensure that our sound features are useful for the fusion tasks, we ignore those videos with no audio channels or channels that are muted. We use wav file format with 16kHz sampling rate, 16bit, monoral channel; the codec is PCM S16 LE.

In AENet, the dimensions of extracted features are $(N, 1024)$, where N equals to the total length in seconds. On the other hand, we used the VGGish to save those features into $(N, 3, 128)$ embeddings. It took about 12 hours to extract features for each from the mini training set with one K80 GPU.

We trained 200 linear SVM binary classifiers for each class using the extracted AENet and VGGish features respectively. Besides, we did not perform any preprocessing on the extracted AENet features while we flattened the extracted VGGish features to dimension $(N, 384)$ before we fed them into the SVM classifiers for training and testing. We got distances to the 200 separating hyperplanes after feeding each testing sample into the 200 binary classifiers and use these distances to do classification.

Table 1. Numbers of videos with and without sound

Videos	With sound	Without sound	Total
Training	55,933	44,067	100,000
Validation	6,286	3,714	10,000
Testing	12,776	7,224	20,000

Feature generation We found out that not all the videos have sound track. The detailed number of videos with and without sound is listed in Table 1. We can see half of videos

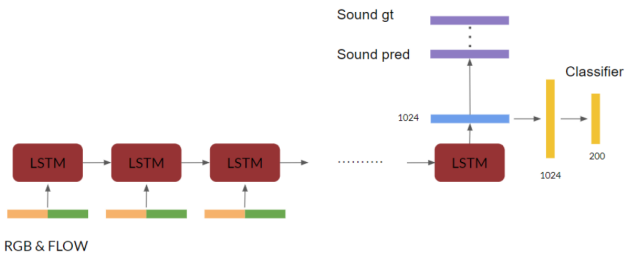


Figure 1. Sound generation with LSTM.

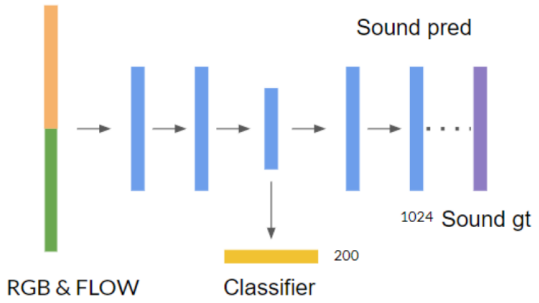


Figure 2. Sound generation with Encoder-Decoder.

do not have sound, but the sound plays an important role in videos. Therefore, we want to generate the sound representation for those videos without sound.

We use two basic structures to generate the sound: LSTM in Figure 1 and encoder-decoder in Figure 2. First, we use the feature representation extracted by TSN as structure input, then go through the structure and get the output feature. The groundtruth sound representation is extracted by AENet and VGGish. In training stage, we use videos with sound to be the training set, and in testing stage, we will generate the sound representation for those videos without sound. We have four kinds of settings: L2 loss+ w/ classifier, L2 loss+ w/o classifier, KL loss+ w/ classifier, KL loss+ w/o classifier. We want to know if label information and different kind of loss are important to the generation.

Table 2. AENet generation with LSTM.

AENet / LSTM	Top-1 acc.	Top-5 acc.
w/o generation (baseline)	4.41%	11.78%
L2 loss, w/ classifier	4.53%	11.69%
L2 loss, w/o classifier	5.19%	13.44%
KL Div., w/ classifier	4.47%	11.50%
KL Div., w/o classifier	4.45%	11.40%

Table 3. VGGish generation with LSTM.

VGG / LSTM	Top-1 acc.	Top-5 acc.
w/o generation (baseline)	1.57%	7.29%
L2 loss, w/ classifier	1.54%	6.91%
L2 loss, w/o classifier	1.95%	7.59%
KL Div., w/ classifier	1.59%	6.85%
KL Div., w/o classifier	1.59%	6.83%

Table 4. AENet generation with fully-connected.

AENet / FC	Top-1 acc.	Top-5 acc.
w/o generation (baseline)	4.41%	11.78%
L2 loss, w/ classifier	4.70%	11.70%
L2 loss, w/o classifier	4.70%	11.70%
KL Div., w/ classifier	4.52%	11.48%
KL Div., w/o classifier	4.55%	11.60%

Table 5. VGGish generation with fully-connected.

VGGish / FC	Top-1 acc.	Top-5 acc.
w/o generation (baseline)	1.57%	7.29%
L2 loss, w/ classifier	2.19%	7.86%
L2 loss, w/o classifier	2.11%	7.84%
KL Div., w/ classifier	1.71%	7.23%
KL Div., w/o classifier	1.72%	6.90%

Feature generation performance The generation performance is found in Tables 2,4,3,5. We can see that for AENet feature, L2 loss + w/o classifier performs the best, and for Vggish feature, L2 loss + w/ classifier performs the best. Therefore, we choose these two model to generate our sound representation.

2.3. Pose-centric features

Our preliminary evaluation, see also Section 4, shows that classes with large intra-class variations, i.e., more abstract classes, are hard to learn for baseline models. To attempt an improvement of these classes, we learn fine-grained, human pose-based features.

Method. We generate discriminative human pose features with the help of *Recurrent Pose Attention Networks* (RPAN) [2]. Given the convolutional feature maps $\mathbf{C}_t$ of each video frame, attention maps $\boldsymbol{\alpha}_t^J$ are learned for each joint J in a human pose. The learning process is supervised by the inclusion of an l2-regression term. As the Moments in Time dataset does not provide human pose annotations, we employ the human pose detector in [1] to retrieve groundtruth annotations.

For the purposes of this work, we simplify the formulation of $\tilde{\boldsymbol{\alpha}}_t = \left[\tilde{\alpha}_t^0, \cdots, \tilde{\alpha}_t^J\right]$ by dropping the partial parameter sharing used in [2]:

$$\tilde{\boldsymbol{\alpha}}_t = \mathbf{v} *_J \tanh\left(\mathbf{A}_h \cdot \mathbf{h}_{t-1} + \mathbf{A}_c *_D \mathbf{C}_t + b\right) \quad (1)$$

$$\alpha_t^J = softmax\left(\tilde{\alpha}_t^J\right) \quad (2)$$

where $*_J$ denotes a $(1 \times 1 \times J)$ convolution. The term $\mathbf{A}_h \cdot \mathbf{h}_{t-1}$ has dimension $D = 32$ and is therefore broadcasted over the spatial dimensions. Input $\mathbf{h}_{t-1}$ is the previous output of the recurrent network learned on body parts, see below.

Given the attentional maps α_t^J, we can construct human body parts P by summation. We follow the work in [2], and construct five body parts *torso, elbow, wrist, knee, ankle*. More formally, we construct F_t^P:

$$\mathbf{F}_t^P = \sum_{J \in P} \sum_k \boldsymbol{\alpha}_t^J \circ C_t \quad (3)$$

where $\circ$ denotes elemenwise multiplication (attention maps are broadcasted over the channel dimension). The result is a fixed-size descriptor for each body part. These five pose features are then max-pooled to form the input to an LSTM recurrent network, for details please refer to [2].

Performance. The method by itself achieved a top-1 accuracy of 21.0% on Moments in Time Mini dataset. This is largely due to the lack of human poses in many classes, which will result in $\mathbf{F}_t^P = \mathbf{0}$. In fact, using the pose detector in [1], we were not able to extract any pose for roughly 47% of all frames.

2.4. Attribute

We consider that some specific objects will appear in related videos, e.g., a knife often appears in the video of `cutting` and `slicing`, a mower often appears in the video of `mowing`, and a computer often appears in the video of `typing`. According to the above inference, we can take these specific objects as the attributes of related videos. In order to obtain the attributes of videos, we use ResNet101 [5] pre-trained on two publicly available multi-label datasets, NUS-WIDE [8] (81 concept labels) and MS-COCO [6] (80 object labels) to extract the feature. We extract features at one frame per second because we believe that the composition of objects will not change dramatically on a framewise basis.

Method We concatenate the extracted feature of three frames as $X \in R^{3\times 2048}$. Given the input X,

$$y = f(X, \theta), y \in R^{200} \quad (4)$$

where $y = [y^1, y^2, ..., y^{200}]^T$ are the predicted label confidences computed by two fully-connected layers.

Table 6. The accuracy of different feature extracted from ResNet101 pre-trained on NUS-WIDE dataset and COCO dataset.

Dataset	Top-1 accuracy	Top-5 accuracy
NUS-WIDE[8]	10.02%	27.15%
MS-COCO[6]	10.14%	27.57%

Table 7. Five classes perform the best.

Best classes	NUS-WIDE[8]	MS-COCO[6]
Top-1	Grilling: 60%	Grilling: 56%
Top-2	Mowing: 54%	Clinging: 54%
Top-3	Typing: 52%	Howling: 54%
Top-4	Welding: 52%	Hiking: 48%
Top-5	Clinging: 46%	Boiling: 46%

Performance According to the result show in Table 7, we can find out that the more similar composition of objects is, the higher accuracy we will get.

2.5. Attribute consistency loss

Method *Attribute consistency loss* (ACL), introduced by [3], focuses on the domain adaptation under the setting of fine-grained recognition. ACL hopes the deep model will be more generalized to examples from the real world instead of overfitting on a given dataset.

In order to do so, ACL uses the concept of multi-task learning: predict classes and attributes at the same time by sharing the last features extracted from the deep model. Here attributes can be any properties we detected from examples. In our case, we use the scores of a model dealing with COCO object detection tasks (80 objects in total). In other words, our attributes represents the probability of occurrence of each object in a video. Besides predicting classes and attributes, the other part in ACL is to reduce the distribution distance (measured by symmetric KL divergence) between predicted attributes and mapped attribute, where mapped attribute is mapped from predicted classes.

In our case, we calculate all the objects' scores for each video in the training set. The results are then grouped by action class to aggregate the mean. Consequently, we have the function to map from action class to 80 object occurrence scores.

Table 8. Five highest and lowest intra-class object variation.

Class	Highest	Class	Lowest
Feeding	1.91618	Erupting	1.28444
Spreading	1.91230	Protesting	1.33439
Scratching	1.87774	Waxing	1.34714
Chewing	1.87588	Tattooing	1.36594
Biting	1.87104	Mowing	1.40809

Performance First, we tried a simple model (LSTM over DenseNet) to evaluate the score with/without ACL on the Moments dataset. The one with ACL took longer time to converge but got close accuracy compared to the one without ACL. We also computed the intra-class object variation, and the results are found in Table 8. However, from Table 9, we find that the F1 score of a class with lower intra-class attributes variation will be higher (negative correlation), showing that if the videos in a class have relatively consistent object occurrence, its easier for a model to perform prediction. Moreover, the model with ACL has lower correlation than the one without ACL. After we apply ACL on TSN, the performance drops a bit. Due to the lack of time, we abandon it and did not do deeper examination. If more attributes extractors from different views are applied, it might be beneficial for future fusion.

Table 9. Correlation between F1 score and intra-class object variation

Correlation	Top-1 acc.	Top-5 acc.
Base model	-0.6019	-0.5462
With ACL	-0.5883	-0.5933

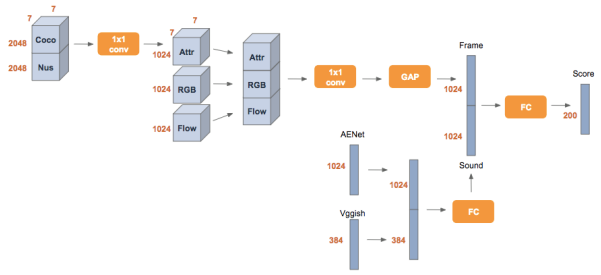


Figure 3. The structure of early fusion method. We fuse the feature maps of the modalities mentioned above at different stages, and then predict the final results.

3. Fusion

We evaluate two fusion schemes, Early Fusion and Late Fusion.

3.1. Early fusion

The early fusion structure is depicted in Figure 3. Let the feature map extracted from ResNet101 pre-trained on COCO dataset be denoted as $C \in R^{7\times7\times2048}$, the feature map extracted from ResNet101 pre-trained on NUS-WIDE dataset as $N \in R^{7\times7\times2048}$, the rgb feature map extracted from TSN as $R \in R^{7\times7\times1024}$, the optical flow feature map extracted from TSN as $F \in R^{7\times7\times1024}$, the feature extracted from AENet as $E \in R^{1024}$ and the feature map extracted from VGGish as $V \in R^{3\times128}$.

For the frame part, first, we concatenate C and N, then go through a 1×1 convolution layer to fuse these two

modalities and denote the fusion feature map of attribute as $A \in R^{7\times7\times1024}$. Second, we concatenate A, R and F then go through a 1×1 convolution layer to fuse these three modalities and denote the fusion feature map of frame as $M \in R^{7\times7\times1024}$. Third, let M go through a global average pooling layer and get $M \in R^{1024}$.

For the sound part, first, we do zero padding for those videos without sound. Second, we concatenate E and V, then go through a fully-connected layer to fuse these two modalities and denote the fusion feature map of sound as $S \in R^{1024}$.

Last, we concatenate M and S as our final feature $\in R^{2048}$ and go through a fully-connected layer to get the prediction.

Table 10. The accuracy of early fusion.

Method	Top-1 accuracy	Top-5 accuracy
Early fusion	22.19%	45.45%

Table 11. The accuracy of early fusion compares to the accuracy of late fusion.

	Increase	Decrease
Top-1	Ascending: +37%	Sailing: -72%
Top-2	Bending: +24%	Protesting: -57%
Top-3	Playing music: +18%	Surfing: -54%
Top-4	Biting: +15%	Hiking: -46%
Top-5	Baking: +14%	Diving: -42%

Performance According to the result shown in Table 11, we can find out that the method of late fusion is better than the method of early fusion on the video classification problem.

3.2. Late fusion

We take the (pre-softmax) prediction scores of every modalities and do the simple and (scalar) weighted average. The results are shown in Table 12.

Table 12. Late fusion of 7 modalities on the MIT Mini validation set.

Method	Top-1 accuracy	Top-5 accuracy
Average fusion	37.09	65.29
Weighted fusion	44.21	72.96

3.3. Ablative study of Late Fusion

In order to identify which modalities provides the largest impact, we perform an ablative study. Given the classifier scores for the seven modalities, we run two late fusion methods (summation and scalar weighting) and report the results in Tables 13 and 14. Note that we tried other parameterized fusion methods, but do not report results here, as those severely overfitted.

Table 13. Ablative study for (late) sum fusion of 7 modalities on the MIT Mini validation set.

Configuration	Top-1 accuracy (%)	Top-5 accuracy (%)
Full	37.09	65.29
w/o TSN (RGB)	31.45	58.11
w/o Flow	34.2	62.01
w/o Aenet	36.96	65.21
w/o Attribute	37.05	65.3
w/o VGGish	36.58	64.82
w/o RPAN	44.84	73.83
w/o TRN (RGB)	31.92	60.12

Table 14. Ablative study for (late) weighted fusion of 7 modalities on the MIT Mini validation set.

Configuration	Top-1 accuracy (%)	Top-5 accuracy (%)
Full	44.21	72.96
w/o TSN (RGB)	37.62	65.92
w/o Flow	39.81	69.22
w/o AENet	43.76	72.77
w/o Attribute	44.22	73.05
w/o RPAN	44.24	73.34
w/o TRN (RGB)	37.47	66.95

Clearly, RGB features remain the most important modality. Pose features did not perform well in the 7-modality fusion, however, it should be noted that RPAN did add a 6% improvement when only the first six modalities were considered, i.e., TRN was left out.

4. Analysis

We train a baseline model consisting of ResNet-50 with an added LSTM layer, and share the observations of our analysis. We begin by studying the confusion matrix and distinguishing different types of confusion:

Semantic similarity is an issue where classes have similar meaning. An example is `slicing`, which is misrecognized as `chopping` in 28% of validation set cases.

Visual similarity Certain actions cannot be discriminated by visual features alone, but require other modalities. An example for this case is `howling` being falsely classified as `barking` by the RGB baseline in 24% of examples.

Subset of class Numerous actions form a subset of or intersect with another action class, which necessitates multi-label classification. The classes `pedaling` and `bicycling` exemplify this, where the latter is misclassified as the former in 16% of cases.

Time reversed classes show similar visual content, but are reversed from each other. One classic instance here is `closing`, which is misclassified in 20% of cases as `opening`.

4.1. F1-score ranking

In the following, we rank classes by their (baseline) F1-score and note our observations. While we cannot list all classes, we list a selection actions in Table 15. Note that for space reasons the table does not show all best- or worst-performing actions.

Table 15. F1-score for selected actions in baseline ResNet-50 + LSTM model.

Class name	F1-Score
Erupting	0.612
Rafting	0.549
Bulldozing	0.454
...	...
Spreading	0.040
Catching	0.026
Opening	0.020
Pulling	0.000

We observe that performance correlates with intra-class variation. Classes such as `erupting` are typically subject to smoke, lava, etc. and therefore easy to recognize. This is unlike the actions with low F1-score in Table 15: Actions like `pulling` are more abstract and can be associated with one of a diverse set of objects; these actions hence have a large intra-class variability.

We propose that more fine-grained features are necessary to improve the failure cases with high intra-class variance. In particular, instead of relying on background information, fine-grained information about pose needs to be retrieved and processed.

5. Conclusions

In this paper, we evaluated many modalities in videos on the Moments in Time dataset, which has a significant intra-class variation among the categories. This work discussed the essential elements of videos from different aspects, and demonstrated experiments on different modalities. Our experiments also indicate that late fusion with many modalities performs better than early fusion.

References

[1] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[2] W. Du, Y. Wang, and Y. Qiao. Rpan: An end-to-end recurrent pose-attention network for action recognition in videos. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3745–3754, Oct 2017.

[3] T. Gebru, J. Hoffman, and L. Fei-Fei. Fine-grained recognition in the wild: A multi-task domain adaptation approach. *CoRR*, abs/1709.02476, 2017.

[4] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 776–780. IEEE, 2017.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016.

[6] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick. Microsoft coco: Common objects in context. *CoRR*, abs/1405.0312, 2014.

[7] M. Monfort, B. Zhou, S. A. Bargal, A. Andonian, T. Yan, K. Ramakrishnan, L. M. Brown, Q. Fan, D. Gutfreund, C. Vondrick, and A. Oliva. Moments in time dataset: one million videos for event understanding. *CoRR*, abs/1801.03150, 2018.

[8] T. seng Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. Nus-wide: A real-world web image database from national university of singapore. In *In CIVR*, 2009.

[9] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[10] N. Takahashi, M. Gygli, and L. V. Gool. Aenet: Learning deep audio features for video analysis. *CoRR*, abs/1701.00599, 2017.

[11] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Val Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[12] B. Zhou, A. Andonian, and A. Torralba. Temporal relational reasoning in videos. *CoRR*, abs/1711.08496, 2017.

SYSU iSEE submission to Moments in Time Challenge 2018

Shuosen Guan
School of Data and Computer Science
Sun Yat-Sen University
GuangZhou, China
gshuosen@gmail.com

Haoxin Li
School of Electronics and Information Technology
Sun Yat-Sen University
GuangZhou, China
LiHaoxin05@gmail.com

Abstract

This report introduces our submission to the Moments in Time Challenge 2018. In this task, we integrate static information, short-term temporal information, long-term temporal information and acoustic information to recognize the actions or events in the videos. Our method finally obtains top-1 accuracy of 27.9% in full-track validation set and 33.6% in mini-track validation set.

1. Introduction

Moments in Time dataset includes a collection of one million labeled 3 second videos, which aims to to help AI systems recognize and understand actions and events in videos.

In this report, we focus on learning different time scale representations for video classification and incorporating other sources of information such as audio signal to provide complementary information. In the following sections we will present our approach and show the results.

2. Approach

In order to understand the videos from multiple temporal scale, we combine static information, short-term temporal information and long-term temporal information via a simple late fusion. In addition, we utilize acoustic signal features since it provide complementary information. We ensemble these models to get the final predictions of the videos. Next we describe each component in detail.

2.1. Static Information

For static information, we exploit frame-based features to recognize actions or events. We deploy Inception-Resnet-V2[6] architecture with temporal segment networks[8] framework. During training, each video is divided into 3 segments and one frame is sampled from each segment. The frame-wise prediction is fused by average pooling. During testing, 20 frames equidistant in time are sampled and the predctions are averaged to genarate video-level prediction.

To improve performance, we fintune the model from ImageNet pretrained and Kinetics-400 pretrained ones. The model finetuned from Kinetics-400 pretrained model achieves higher accuracy. Besides, considering training on hard samples, we try to use focal loss[4] in this classification task and find that it just accelerated convergence but did't increase the performance.

Our performance comparison on validation set is showed in Table 1.

Models	Full-track Top-1	Mini-track Top-1
IR-scratch	0.1946	-
IR-ImageNet	0.2419	-
IR-Kinetics-400	**0.2524**	0.3026
IR-Kinetics-FL	0.2513	**0.3124**

Table 1. Performance comparison of different models for static information.(IR here denotes InceptionResnetV2.)

2.2. Short-term Temporal Information

To encode spatial and short-term temporal information, we apply Pseudo-3D Residual Networks[5] in our approach. We use 199 layers variant as our base framework and mix different P3D Blocks as described in [5]. In the training stage, one 16-frame clip is randomly sampled from each video as the input while during testing we sample 4 clips uniformly from each video and fuse the output of the final layer.

We first pretrain our model from Kinetics-400 dataset and then full-train the model on the Moments in Time dataset. For Mini-track, to accelerate the training process and capture longer term motion information, we experiment with different sampling strategies on the input: sampling clips from consecutive frames and down-sampling clips with different sampling intervals. Accuracy comparison on validation set is described in Table 2.

Models	Full-track Top-1	Mini-track Top-1
P3D-Kinetics	0.2091	**0.2634**
P3D-Kinetics-s2	-	0.2612
P3D-Kinetics-s4	-	0.2614

Table 2. Performance of different models using Pseudo-3D Residual Networks with different sampling interval. s2, s4 denote the sampling interval of 2 frames and 4 frames respectively.

2.3. Long-term Temporal Information

To capture long-term temporal information, we intend to model the temporal evolutions of features. We first extract frame-level features using our Kinetics pretrained Inception-Resnet-V2 model from 10 frames uniformly sampled from each video, and then apply a temporal convolution (denoted as TemporalConv or TC below for simplicity) and a parametric pooling along time dimension, which follows a MOE model like [1] to classification.

Inspired by ARTNet proposed in [7], we further employ a multiplicative interactions (denoted as MultiplyInter or MI below for simplicity) to model relations across features as a supplement to the TemporalConv features.

Moreover, Temporal Relation Network[9] models the temporal dependencies between multiple frames at multiple time scales. Here we use the pretrain model[1] provided by the author to model multi-scale temporal informatioin for classification.

Results of different methods on validation set are illustrated in Table 3.

Methods	Full-track Top-1	Mini-track Top-1
TemporalConv	0.2626	0.3251
MultiplyInter	**0.2638**	**0.3268**
TRN	0.2120	-

Table 3. Performance of different methods for long-term temporal information.

2.4. Acoustic Information

We also utilize acoustic features as complementary information in our approach. We first compute log mel spectrograms from the audio of each video and use a pre-trained VGGish model[3] to extract 128-D semantically meaningful, high-level embedding features[2], and then take the features as input and use a 4 layers full-connected network for classification. We finally obtain 0.045 top-1 accuracy on validation set.

[1] `http://relation.csail.mit.edu/models/TRN_moments_RGB_InceptionV3_TRNmultiscale_segment8_best.pth.tar`

3. Ensemble Results

Finally, we ensemble the models mentioned above to get the prediction. Results on Full-track and Mini-track are showed in Table 4 and Table 5 respectively. It should be noted that in both two tracks, we use the consecutive-sampling strategy mentioned above in P3D models for final combinations.

Models combinations	Top-1	Top-5
IR+P3D	0.2638	0.5187
IR+P3D+TRN	0.2676	0.5262
IR+P3D+TC+TRN	0.2746	0.5345
IR+P3D+TC+MI+TRN	0.2786	0.5368
IR+P3D+TC+TRN+audio	0.2756	0.5291
IR+P3D+TC+MI+TRN+audio	**0.2796**	**0.5397**

Table 4. Top-1 and Top-5 accuracy of different models combinations on Full-track.

Models combinations	Top-1	Top-5
IR+P3D	0.3246	0.6082
IR+P3D+TC	0.3337	0.6196
IR+P3D+MI	0.3347	**0.6237**
IR+P3D+TC+MI	**0.3358**	0.6219

Table 5. Top-1 and Top-5 accuracy of different models combinations on Mini-track.

References

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[3] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson. Cnn architectures for large-scale audio classification. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017.

[4] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999 – 3007, Venice, Italy, Oct 2017. IEEE.

[5] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5534–5542. IEEE, 2017.

[6] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on

learning. In *AAAI Conference on Artificial Intelligence*, pages 4278 – 4284, San Francisco, California USA, 2017. AAAI.

[7] L. Wang, W. Li, W. Li, and L. Van Gool. Appearance-and-relation networks for video classification. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1430 – 1439, Salt Lake City, Utah, June 2018. IEEE.

[8] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36, Amsterdam, Netherlands, 2016. Springer, Springer, Cham.

[9] B. Zhou, A. Andonian, and A. Torralba. Temporal relational reasoning in videos. *arXiv preprint arXiv:1711.08496*, 2017.

Trimmed Event Recognition：submission to ActivityNet Challenge 2018

Lei Zhou, Jiaze Wang, Xiaojiang Peng, Yali Wang, Yu Qiao

Shenzhen Institutes of Advanced Technology, CAS, China

Abstract

This notebook paper describes our system for the trimmed event recognition (Moments in Time) task in the ActivityNet challenge 2018. We investigate multiple state-of-art approaches for the event recognition in short, trimmed videos. With these approaches, we derive an ensemble of deep models.

1. Introduction

Event recognition has remained a challenging task in the computer vision community. The research about event recognition also are very important in other tasks like video understanding. And Moments in Time dataset is a large-scale human-annotated collection of one million short videos corresponding to dynamic events unfolding within three seconds [1].

The rest of this paper is organized as follows. Section 2 presents our approach in detail, finally Section 3 concludes this work.

2.Experments

2.1 Data Augmentation

We fix the size of input image or optical flow fields as 256 × 340, and the width and height of cropped region are randomly selected from {256,224,192,168}. Finally, these cropped regions will be resized to 224×224 for network training. What's more, we use additional three-quarters validation set as a part of training set and use the other one-quarter validation set (Seen as Spilt1 Val) to evaluate the performance of our models.

2.1 CNN models

The main model we use is STRNet, i.e. Spatiotemporal Recalibration Networks. STRNets are 3D networks which aim to solve the temporal disturbance problem in vanilla 3D networks and factorized spatiotemporal networks. Due to features from different pipelines can capture different information. We use four other models to capture additional information. According to the ensemble results, they can significantly improve the performance on the on the Spilt1 validation set. These four models are Resnet101[2], TSN [3], TRN [4] and I3D [5].

2.2 Experiments results

Table 1 shows the Top-1 and Top-5 accuracy of the baseline models on the Spilt1 validation set. The best single model is the STR18_tr, with a Top-1 accuracy of 29.76% and a Top-5 accuracy of 56.71%.

Index	Model	Test Set	Top1	TOP5	Ave
A	STR18_tr	Spilt1 Val	**29.76%**	**56.71%**	**43.23%**
B	R101	Spilt1 Val	27.49%	52.41%	39.95%
C	I3D	Spilt1 Val	24.40%	49.37%	36.88%
D	TRN	Spilt1 Val	24.59%	48.90%	36.74%
E	STR18_tr_of	Spilt1 Val	17.98%	39.50%	28.74%
F	TSN	Spilt1 Val	24.67%	49.52%	37.10%
G	STR34	Spilt1 Val	28.64%	55.99%	42.31%

TABLE1: Classification Accuracy: We show Top-1 and Top-5 accuracy of the baseline models on the Spilt1 validation set.

As is shown is TABLE 2, the Ensemble model (average) gets the Top-1 accuracy as 32.08% and Top-5 accuracy as 59.23%.

Index	Test Set	Top1	TOP5	Ave
A	Spilt1 Val	29.76%	56.71%	43.23%
A+B	Spilt1 Val	31.59%	58.28%	44.93%
A+B+C	Spilt1 Val	31.23%	58.40%	44.81%
A+B+D	Spilt1 Val	31.91%	58.76%	45.33%
A+B+D+E	Spilt1 Val	31.82%	59.10%	45.46%
A+B+D+E+F	Spilt1 Val	31.82%	59.09%	45.46%
A+B+D+E+G	Spilt1 Val	**32.08%**	**59.23%**	**45.65%**

TABLE2: Ensemble Results: We show Top-1 and Top-5 accuracy of the ensemble models on the Spilt1 validation set.

3.Conclusion

This paper describes our team's solution to task of trimmed event recognition. Features from different pipelines can capture different information. We propose several 3D spatial-temporal models for event recognition. We also investigate the performance of several 2D CNNs like TSN, TRN and Resnet101.

References

[1] Monfort M, Zhou B, Bargal S A, et al. Moments in Time Dataset: one million videos for event understanding[J]. arXiv preprint arXiv:1801.03150, 2018.

[2] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778

[3] Wang L, Xiong Y, Wang Z, et al. Temporal segment networks: Towards good practices for deep action recognition[C]//European Conference on Computer Vision. Springer, Cham, 2016: 20-36.

[4] Zhou B, Andonian A, Torralba A. Temporal Relational Reasoning in Videos[J]. arXiv preprint arXiv:1711.08496, 2017.

[5] Carreira J, Zisserman A. Quo vadis, action recognition? a new model and the kinetics dataset[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017: 4724-4733.

Moments in Time: submission to ActivityNet Challenge 2018

Shu-Dong Yang, Zhi-Wei Ren, De-Ming Cong, Di Wen, Kong Ye,
Jun-Yan He, Xiao Wu

Southwest Jiaotong University
{shudong.yang, vincentren}@my.swjtu.edu.cn,
sc16d2w@leeds.ac.uk
{congdeming1995, tiankong2586,junyanhe1989,wuxiaohk}@gmail.com

Abstract

This paper describes our method for the Moments in Time Recognition Challenge Full track to ActivityNet Challenge 2018. In this task, we propose a method for action recognization by using Non-local Neural Networks,the Deformable Convolutional Networks and Temporal Relational Reasoning in Videos. We further demonstrate that a ConvNet trained audio information can help with the recognization of the Moments in Time dataset. We only use RGB frames and audio features to training.

1. Introduction

In recent years, Video-based human action recognition has become an intensive area of research in the fields of computer vision and pattern recognition [5, 6, 7]. There are two significant information in this fields: appearance and motion. For appearance, since hand-craft features such as sift are unable to capture global information, numerous recent methods have utilized Convolutional Neural Networks (CNNs) whose superiority over hand-crafted ones in this field has been shown. [2] As for motion, it is frequently represented by optical flow or other motion-based descriptors. Simonyan et al. [4] 's two-stream CNN network which employed optical flow into temporal network to extract motion information. However, the pre-calculation of optical flow is significantly complicated which illustrated the exceeding difficulty of applying it into real-time recognition.We propose an approach for human action recognition which fused various CNNs (Audio-TSN, TRN [7], DCN[1] , Non-local Network [6]) to extract different features from different videos with an end-to-end training process. Considering the application of real-time action recognition, optical flow has not been implemented in our approach. In this submission to the challenge, we aim to evaluate the proposed model on the Moments in Time dataset

2. Moments in Time

Moments in Time Dataset is a large-scale human-annotated collection of one million short videos which has the same length of 3 second. There are 339 different classes in total. And there are existing some action partly or even fully depend on the audio information. Moments in Time dataset is also joint as a task in the ActivityNet Challenge 2018. There are two different tracks. The first track is the full track, which is a classification task on the entire Moments in Time dataset. It contains 339 classes, 802,264 training videos, 33,900 validation videos, and 67,800 testing videos. The second track is the mini track, which is a classification take for students on a sub set of Moments in time dataset. It contains 200 classes, 100,000 training videos, 10,000 validation videos, and 20,000 testing videos.

3. Method

Our approach use 3 kind of neural network for extract apperance features from RGB image: Non-local Network [6], Temporal Relation Network (TRN) [7], Deformable ConvNets (DCN). We found that audio information also play a important role in video analyzing, so we also use audio feature to imporve our recognition accuracy.

We only use RGB frames adn audio features as our training data. Therefore we lost some temporal information from still image, so we choose 3D-based convolution neural networks (CNN) to exploit temporal information from continuous frames. Because of Non-local Network is the most powerful architecture in 3D-CNN, so we choose it for temporal feature learning. Relevance also exists in contiguous video frames. For instance, in the action of drinking, taking

the glass should be anterior to getting the mouse close to it. The relevancy of action makes it irreversible. So we use TRN for extract temporal relationship between continuous frames. And we think in still image, there is also an relation between objects such as bottle and person. In our approach, we do not use object detection methods such as Faster R-CNN [3] or YOLO directly. We use the deformable ConvNets for spatial relationship learning because it success in oject detection area. Audio is also an important feature in our approach, we use mel spectrogram feature as our training data, then use Temporal Segment Network (TSN) to training.

3.1. Training

1. For Non-local Network, we use both i3d and c2d as Non-local Network's backbone, the network is trained using SGD for 400k iterations. The base learning rate is 0.01 and the stepsize is 150k and 300k.

2. When traning TRN network, we use the released pre-trained RGB model for full track. For mini track, we use InceptionV3 as network's backbone, and we choose 8-segments and multiscale strategy for training.

3. For DCN network, we use ResNet101 as network's backbone, and replace the res5-c bottleneck to DCN ConvNets.

4. For audio stream, we first extract wav file from video and use audioset to get mel spectrogram feature, then use TSN network to training.

4. Conclusion

Although we do not get the best performance in competition, but it shows that relationship in continuous frames plays an importanct role in video analyzing. And we found audio also can help effectively.

References

[1] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. *CoRR, abs/1703.06211*, 1(2):3, 2017.

[2] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, et al. Cnn architectures for large-scale audio classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 131–135. IEEE, 2017.

[3] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[4] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[5] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Val Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[6] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. *CVPR*, 2018.

[7] B. Zhou, A. Andonian, and A. Torralba. Temporal relational reasoning in videos. *arXiv:1711.08496*, 2017.

Team SSS Submission to the Moments in Time Challenge 2018

Yao Zhou, **Pingchuan Ma** and **Yu Lu**
SenseTime Research
{zhouyao,mapingchuan,luyu}@sensetime.com

Abstract

This draft presents our methods and results on the Moments in Time challenge 2018. To tackle the task of recognizing the events or activities in trimmed videos, we tried various models with different input modalities. In summary, we not only explored both 2D and 3D models with the different backbones, but also fed both RGB frames and optical flows into these models. Finally, we ensemble these models to achieve the better recognition performance.

1. Our Approach and Result

In the Moments in Time challenge 2018, the participants should design methods to recognize the events in the three-seconds videos. This challenge uses the Moments in Time dataset as the benchmark. This dataset is challenging for the reason that (1) the events occurred in videos are abstract, (2) the events are not only performed by human, but also animals or objects, (3) the events are visual and/or audible actions. This dataset contains 802,264 videos for training, 33900 for validation and 67800 for testing. At this time, each video only belongs to one of 339 classes.

We present the experiment result at Table 1.

Model	Modality	K	Backbone	Top-1 Acc.	Top-5 Acc.
C2D TSN	RGB	5	Inception-v3	27.2%	51.5%
C2D TSN	RGB	5	ResNet-101	29.5%	55.8%
C2D TRE	RGB	5	ResNet-101	30.8%	56.6%
C2D R101	Flow	5	ResNet-101	16.4%	37.5%
C2D TRE	Flow	5	ResNet-101	16.8%	38.0%
I3D Inv1	RGB	16	Inception-v1	26.2%	50.3%
I3D R50	RGB	16	ResNet-50	26.6%	50.5%
NL-I3D R50	RGB	16	ResNet-50	28.1%	53.7%
I3D Inv1	Flow	16	Inception-v1	10.1%	27.9%

Table 1. The experiment results on the validation set of Moments in Time full dataset. The first column indicates the model names. For 2D models, K present the num of segments for training. For 3D models, K presents the num of frames for training. The first group presents the results performed by the C2D models with different backbones when using the RGB modality. The second group presents the optical flow results. The last two groups are I3D models with different backbones and modalities.

Trimmed Event Recognition Submission to ActivityNet Challenge 2018

Jiaqing Lin, Akikazu Takeuchi
STAIR Lab, Chiba Institute of Technology, Japan
{lin, takeuchi}@stair.center

1. Overview

This paper describes STAIR Lab submission to ActivityNet 2018 Challenge for guest task C: Trimmed Event Recognition (Moments in Time) [1]. Our approach is to utilize three networks, Audio Net, Spatial-temporal Net, and DenseNet to make individual predictions, then use MLP to fuses the results to make an overall prediction. The flow chart of our approach is shown in figure 1.

2. Implementation

2.1 Audio network

Our audio dataset training is different from other methods. Usually, auditory raw waveforms are used as input and are fed into a model like SoundNet [2]. In our case, firstly, we converted auditory raw waveforms to spectrogram images, then fed them to 2D ResNet101 [3] to train a classifier. The top-1 accuracy of this model is 13.04%, which is higher than top-1 accuracy 7.60% presented in [1].

2.2 Spatial-temporal network

We used 3D ResNet101 [4] to extract spatial-temporal visual features from a video. To train a classifier, a temporal position in an input video is randomly selected, and 16 frames are extracted around the selected temporal position. The frames are spatially cropped by multi-scale random four corner and center cropping, and horizontally flipped with 50% probability. Other parameters are same as the paper [4].

2.3 2D RGB network

Single frame in a video is still informative even in the action recognition. So we used DenseNet [5] for extracting image features from a randomly selected frame in a video. Number of layers was 201.

2.4 Fusion

We utilized the three models above to predict the test set. Log Softmax function is applied to the last layer of each model, and results are concatenated to generate two vectors, one including audio prediction, the other without audio prediction. Then, MLP is trained. Top-1 and top-5 accuracy of out method for the validation set are shown in table 1.

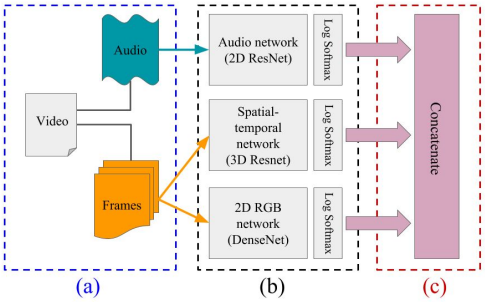


Fig. 1. (a) Extract audio and frames from a video as each network input. (b) Three networks are 2D ResNet, 3D ResNet, and DenseNet. (c) Fuse concatenated three results by using MLP.

Table 1

Model	Modality	Top-1(%)	Top-5(%)
2D ResNet101	Auditory	13.04	28.03
3D Resnet101	Spatial+Temporal	24.85	50.37
DenseNet	Spatial	24.5	48.4
Fusion (MLP)	A+S+T	29.97	57.26

References

[1] Monfort, Mathew, et al. "Moments in Time Dataset: one million videos for event understanding." *arXiv preprint arXiv:1801.03150* (2018).

[2] Aytar, Yusuf, Carl Vondrick, and Antonio Torralba. "Soundnet: Learning sound representations from unlabeled video." In *Advances in Neural Information Processing Systems*, pp. 892-900. 2016.

[3] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[4] Hara, Kensho, Hirokatsu Kataoka, and Yutaka Satoh. "Learning spatio-temporal features with 3D residual networks for action recognition." *Proceedings of the ICCV Workshop on Action, Gesture, and Emotion Recognition*. Vol. 2. No. 3. 2017.

[5] Huang, Gao, et al. "Densely connected convolutional networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. Vol. 1. No. 2. 2017.

Submission to Moments in Time Challenge 2018

Yunkai Li[1], Ziyao Xu[1], Qian Wu[2,†], Yu Cao[3,†], Shiwei Zhang[4,†], Lin Song[5,†], Jianwen Jiang[6,†], Chuang Gan[6*], Gang Yu[1*], Chi Zhang[1*]
[1]Megvii Inc. (Face++), {liyunkai, xuziyao, yugang, zhangchi}@megvii.com
[2]Zhejiang University, wq1601@zju.edu.cn
[3]Beihang University, cqcy1208@buaa.edu.cn
[4]Huazhong University of Science and Technology, swzhang@hust.edu.cn
[5]Xian Jiaotong University, stevengrove@xtu.xjtu.edu.cn
[6]Tsinghua University, jjw17@mails.tsinghua.edu.cn, ganchuang1990@gmail.com

***Abstract*—This paper introduces our solution for the full track of the Moments in Time 2018 video event recognition challenge. Our system is built on spatial networks and 3D convolutional neural networks to extract spatial and temporal features from the videos. We also take advantage of multi-modality cues, including optical flow and audio information to further improve the performances. Our final submission is an ensemble of 5 models: three based on RGB frames as well as one optical flow model and one audio model, achieving top1 38.1%, top5 65.3% on the validation set.**

I. Introduction

Video recognition is one of the most fundamental research topics in the computer vision. With development of computation and release of large video classification dataset such as Kinetics [8] and Moments in Time [13], it has therefore been an urgent need to develop more efficient automatic video understanding and analysis algorithms.

Currently, there are three kinds of successful frameworks that dominate the video recognition (1) two-stream CNNs [15], [17], (2) 3D CNNs [16] and its variant [14], [18], and (3) 2D CNNs with temporal models on top such as LSTM [3], [9], temporal convolution [1] and attention modeling [10], [11]. The winner of Kinetics challenges last year [1] proposed a novel solution by first extracting the multi-modality features from the learned networks and then fed them into the off-shelf multi-modality temporal models to conduct video classification. However, these approaches are not applicable to large-scale video datasets, such as Moments in Time [13], since they rely on extracting features from all videos beforehand, which is extremely time-consuming and expensive.

To address these challenge, we mainly adopt end-to-end training architectures with three modalities, namely appearance, motion and acoustic information. We compared the performance of different models and finally chose Inflated 3D and Non-local module for appearance modality and 2D CNN model for the motion and acoustic modalities.

The remaining sections are organized as follows. Section II presents some details of our method. In section III, we compare different approaches, followed by the conclusion of this report in section IV.

† Work down while interning at Megvii

II. The proposed Method

In this section, we will introduce the applied multiple models and modality, including observation and obtained score for each model.

A. Appearance clues

We have experimented different methods including 2D CNNs, Temporal Segment Networks and inflated 3D neural network to extract the video feature. We extract RGB frames from the videos at 25 fps as original resolution and applied random crop as augmentation.

Spatial Network. We used Xception network [2] pre-trained on the Kinetics dataset [8], as well as SENet and SEResNeXt [6] initialized on ImageNet.

In training, one single RGB image is randomly selected from the video as the input. In validation, we followed the testing method in TSN [17] with 25 segmentation and average fusion.

Temporal Segment Networks. We also explored the Temporal Segment Networks with 5 segments. Surprisingly, it is not as effectively as in other activity recognition tasks. The performance of TSN is even lower than single image performance described above. We speculate that it is due to the large intra-classes variances and short video duration of the dataset.

Inflated 3D Network. We combine the Inflated ResNet50 network with non-local modules as the base model.

We apply spatial and temporal convolution separately in the ResNet block [5], which improves accuracy while reducing the calculations. We pre-trained the model in Kinetics, and fine-tuned the network as the base model of TSN. In validation inference, we crop the input lager than 224, with spatially average the predictions after the Softmax layer as described in [4].

Word2Vec Network. Considering the large intra-class variance of the dataset, We tried to transfer labels into vectors and minimize the distance between feature and vectors instead of classification loss.

B. Motion clues

We used an OpenCV implementation of TV-L1 [19] algorithm for computing dense optical flow and converted 2-

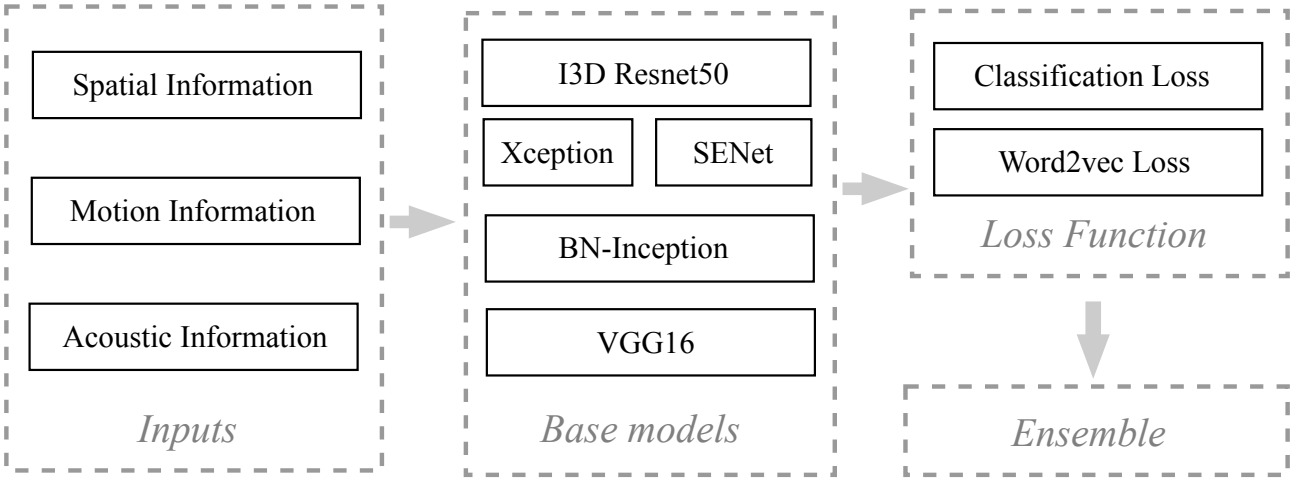


Fig. 1. The designed framework in our method. we apply 2D and 3D convolutional neural network to extract spatial and temporal feature from the video while take advantage of multi-modality cues, including optical flow and audio information. Our final submission is an ensemble of these models.

TABLE I
RESULTS ON VALIDATION SET.

Model	Modality	Top-1 Accuracy(%)	Top-1 Accuracy(%)
ResNet50	RGB	28.3	53.2
TSN	RGB	27.4	53.2
Xception	RGB	31.8	59.2
SENet152	RGB	33.7	61.3
SEResNeXt	RGB	33.0	60.2
Word2Vec ResNet50	RGB	29.9	56.2
I3d ResNet50	RGB	34.2	61.4
BN-Inception	Flow	19.1	41.2
VGG16	Audio	9.1	21.3
Ensemble		**38.1**	**65.2**

channel optical flow vectors (u, v) into its magnitude and direction and stored them as RGB images. We used these images in the BN-Inception [7] network and takes a stack of 5 consecutive optical flow fields as input. We employed SGDR [12] strategy in optical flow training, since we found that restart the learning rate is helpful to promote the accuracy. We obtained a validation accuracy of 19.09% (top-1,) 41.17% (top-5)

C. Acoustic clues

In compliance with the common practice to processing audio features, a convolutional network based audio classification system is used. With each video divided into 10 frames, its frequency domain information is extracted through Fourier Transformation, histogram integration and logarithm transformation. The vocal information of each video is shaped as 10x96x64 to a VGG classification net to generate a label probability distribution prediction.

The character that vocal information is hard to do augmentation makes it likely to overfit the training set. So generally less complex net leads to a better evaluation results.

D. Training

In this section, we present some details of our method during training stage. We train the our network end-to-end with 0.01 initial learning rate and reducing it by a factor of 10 at every 15 epoches. For each RGB and acoustic model, we train about 30 epoches and 60 epoches for flow model. We train our model on the 8 Titan GPUs for single image and TSN experiment, while 3D models are trained in distribution mode.

III. EXPERIMENT RESULTS

In this section, we present some experiments in our method in the Table II-A. In this table, we show the results with different 2D/3D mdoels. From the results, we can find that i3d resnet50 with model non local can achieve the best results. While the single image method accuracy is actually not much lower than i3d network, TSN performs not as efficient as other datasets. We found that spatial and temporal information are mutually complementary for final feature fusion. Meanwhile, motion and acoustic information are essential though the scores are low, showing the importance of different modality clues.

Finally, we ensemble all the models on the score after softmax function to obtain 38.1% top-1, and 65.2% top-5 accuracy on the validation set.

IV. Conclusion

In Moments in Time Challenge 2018, we design a new spatio-temporal action recognition framework. We make advantage of both 2D spatial network and 3D network, as well as multi modalities. By this means, we can better extract the feature of the video in more patterns. In the future, we will explore the fundamental difference between Moments in Time dataset with other datasets and find better general presentation under large variance in intra-class distribution.

References

[1] Y. Bian, C. Gan, X. Liu, F. Li, X. Long, Y. Li, H. Qi, J. Zhou, S. Wen, and Y. Lin. Revisiting the effectiveness of off-the-shelf temporal modeling approaches for large-scale video classification. *arXiv preprint arXiv:1708.03805*, 2017.

[2] F. Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv preprint arXiv:1610.02357*, 2016.

[3] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. pages 2625–2634, 2015.

[4] B. Ghanem, J. C. Niebles, C. Snoek, F. C. Heilbron, H. Alwassel, R. Khrisna, V. Escorcia, K. Hata, and S. Buch. Activitynet challenge 2017 summary. *arXiv preprint arXiv:1710.08011*, 2017.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 2017.

[7] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[8] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[9] F. Li, C. Gan, X. Liu, Y. Bian, X. Long, Y. Li, Z. Li, J. Zhou, and S. Wen. Temporal modeling approaches for large-scale youtube-8m video understanding. *arXiv preprint arXiv:1707.04555*, 2017.

[10] X. Long, C. Gan, G. de Melo, X. Liu, Y. Li, F. Li, and S. Wen. Multimodal keyless attention fusion for video classification. AAAI, 2018.

[11] X. Long, C. Gan, G. de Melo, J. Wu, X. Liu, and S. Wen. Attention clusters: Purely attention based local feature integration for video classification. *CVPR*, 2018.

[12] I. Loshchilov and F. Hutter. Sgdr: stochastic gradient descent with restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[13] M. Monfort, B. Zhou, S. Adel Bargal, A. Andonian, T. Yan, K. Ramakrishnan, L. Brown, Q. Fan, D. Gutfruend, C. Vondrick, and A. Oliva. Moments in Time Dataset: one million videos for event understanding. *ArXiv e-prints arXiv:1801.03150*, 2018.

[14] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 2017.

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

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

[17] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool. Temporal segment networks: Towards good practices for deep action recognition. *ECCV*, 22(1):20–36, 2016.

[18] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. *arXiv preprint arXiv:1711.07971*, 2017.

[19] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint Pattern Recognition Symposium*, pages 214–223. Springer, 2007.

UNSW Video Classification System for Moments in Time Challenge 2018

Zhihui Li
School of Computer Science and Engineering
University of New South Wales
zhihuilics@gmail.com

Lina Yao
School of Computer Science and Engineering
University of New South Wales
lina.yao@unsw.edu.au

Abstract

This paper presents our system for the video understanding task of the Moments in Time Challenge 2018. Because of limited computational resources, we only used three features in the system, including 2 visual features and 1 audio feature. After we have the prediction scores of these three features, we combine them using late fusion and obtain the final result. Specifically, we observe average fusion can get promising results in our experiments.

1. Introduction

Although researchers have devoted much research attention to the visual understanding problem, it is still a challenging problem. The ubiquitous video record devices have created videos far surpassing what the users can watch. Hence, it becomes increasingly urgent to develop efficient algorithms for automatic video analysis.

Researcher have made much progress to introduce large-scale datasets for training reliable deep learning models, for example, ImageNet [4], and Youtube8M [3] . Recently, researchers from MIT have introduced the Moments in Time Dataset, a collection of one million short videos with a label each, corresponding to actions and events unfolding within 3 seconds.

2. The Proposed System

In this section, we describe the proposed system for Moments in Time Challenge 2018.

2.1. Feature Extraction

For the limitations of computing resources, we only employ three features in our system, including 2 visual features and 1 audio feature.

Visual Features: We first pretrain an Inflated 3D ConvNet (I3D) [2] model on ImageNet and Kinects datasets. Then we apply the pretrained model to the Moments dataset, and extract the last pooling layer as the representation for each video.

In addition, we use the TRN-Multiscale [5] following the baselines reported in [3], since it achieves the best single model performance. We do not pre-train or fine-tune on this model. In other words, we only do inference for this model.

Audio Feature: We employ raw waveforms as the input modality and adopt the network architecture from SoundNet [1]. The only difference is that we changed the last layer to predict the categories from the Moments dataset. We fine-tune the model downloaded from the official website.

2.2. Inference

For the TRN-Multiscale and audio features, the inference is conducted end-to-end. For the I3D feature, we feed the last pooling layer into a 200-mixture Mixture of Experts (MoE) layer for classification.

2.3. Fusion

When the prediction scores for the three models are ready, we fuse them using average fusion for its simplicity and efficiency.

3. Results

We report the results in Table 1. From the experimental results we can see that the I3D model gets the best single model performance on the validation set. Also, we observe that with only three models, we get similar performance as the baseline reported in the baseline paper.

4. Conclusion

In this paper, we have presented our system for the Moments challenge. Although the computing resource is very limited (with only one Titan Xp), we finally achieve promising results on the validation set.

Table 1. The performance evaluation on the validation set.

Feature Name	Mode	Top-1	Top-5
I3D	Visual	29.53	56.28
TRN-Multiscale	Visual	28.27	53.87
SoundNet	Audio	7.60	17.96
Average Fusion	–	30.25	57.84

Acknowledgement

Dr. Yao is partially supported by the Australian Research Council DECRA project.

References

[1] Y. Aytar, C. Vondrick, and A. Torralba. Soundnet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 892–900, 2016. 1

[2] J. Carreira and A. Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4724–4733, 2017. 1

[3] M. Monfort, B. Zhou, S. A. Bargal, A. Andonian, T. Yan, K. Ramakrishnan, L. M. Brown, Q. Fan, D. Gutfreund, C. Vondrick, and A. Oliva. Moments in time dataset: one million videos for event understanding. *CoRR*, abs/1801.03150, 2018. 1

[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1

[5] B. Zhou, A. Andonian, and A. Torralba. Temporal relational reasoning in videos. *CoRR*, abs/1711.08496, 2017. 1

Moments in Time Summary

Elliot Holtham[1], Moumita Roy Tora[1], Keegan Lensink[1], David Begert[1], Lili Meng[1], Megan Holtham[1], Eldad Haber[1], Lior Horesh[2], Raya Horesh[2]

[1] – Xtract AI

[2] – IBM Research

Before tackling the 339 class challenge, the full dataset was split into 20 classes to create a smaller test problem upon which different solution methods could be examined more quickly. For the 20 class subset, a variety of actions were chosen that would require different data streams for effective classification (for example barking & clapping for audio), bowling and rafting for RGB images, and ascending for motion. After looking through several of the training videos, it was apparent that the content and action of the video could abruptly change throughout several of the videos. Each 3 second video was split into 3 x 1s segments with hope that at least one of the second segments would capture the main action of the video.

For the mp4 videos which contained audio, .wav files were extracted from the video. Initially the .wav files were converted into spectrograms which were then trained using ResNet 101 network. Because of lack of time on the final 339 class problem, a pre-trained VGG on Audioset was used for the audio files. The features from each second for each video was extracted and then the three consecutive feature vectors were passed into an LSTM for the classification and the creation of the first data stream. The features for the videos with no audio files were zero padded such that the dimensions matched the other streams.

Static images were extracted from the videos at 5 fps using ffmpeg. The images were used to fine-tune a pre-trained ResNet 101 model from ImageNet. The trained ResNet 101 model was used in two ways. Firstly, the features from each frame were extracted and one random frame feature from each second used to train a LSTM to create a separate stream. Secondly, as in the MIT/IBM paper, the logits from 6 equidistant frames were averaged to produce a separate data stream.

Motion from the videos was extracted in three ways. Firstly, a pre-trained (ImageNet then Kinetics) I3D model was fine-tuned on the Moments data at 15 fps. Unfortunately our team was running out of time so didn't manage to fully fine-tune this model to the level that was certainly possible. Secondly, temporal slices from the videos were extracted from the 3D volume and used to fine-tine a pre-trained ResNet 101 model. For the 20 class subproblem, we had originally worked with our Leap-Frog network architectures (https://arxiv.org/abs/1705.03341) and had gotten better results than the ResNet 101 network, but didn't have time to train from scratch on the full 339 class problem. Thirdly, a pre-trained TRN model was also used that was provided by the "MIT-IBM Watson AI Lab and IBM Research" group. This provided model was run on the validation and test datasets.

The audio, temporal and spatial stream features were combined together with an LSTM before all of the logits from each stream was ensembled together in a weighted average of the logits

where the weights were based on the accuracy of that data stream on the validation dataset. All of the computations were run internally on desktop computers running GTX 1080Ti video cards and 500 GB M2 SSDs. The above workflow gave 34.00% top 1 and 61.75% top 5 on the validation set.