Mid-level Fusion for End-to-End Temporal Activity Detection in Untrimmed Video

Md Atiqur Rahman
mrahm021@uottawa.ca
Robert Laganière
laganier@uottawa.ca

School of Electrical Engineering and
Computer Science
University of Ottawa
Ottawa, Canada

Abstract

In this paper, we address the problem of human activity detection in temporally untrimmed long video sequences, where the goal is to classify and temporally localize each activity instance in the input video. Inspired by the recent success of the single-stage object detection methods, we propose an end-to-end trainable framework capable of learning task-specific spatio-temporal features of a video sequence for direct classification and localization of the activities. We, further, systematically investigate *how* and *where* to fuse multi-stream feature representations of a video and propose a new fusion strategy for temporal activity detection. Together with the proposed fusion strategy, the novel architecture sets new state-of-the-art on the highly challenging THUMOS'14 benchmark – up from **44.2% to 53.9% mAP (an absolute 9.7% improvement)**.

1 Introduction

With the rapid proliferation of cheap and accessible cameras (e.g., smartphone, surveillance cameras etc.), the amount of video data amassed daily is enormous. No surprise, the most prevalent and interesting contents of these videos are humans. Therefore, human activity analysis plays a central role in automatic video understanding. Though impressive success has been achieved in video activity recognition [5, 15, 29, 31, 33, 36, 37], many real-world applications, such as surveillance video analysis, semantic video search etc. require analyzing videos that are long, and temporally untrimmed. This leads to the task of Temporal Activity Detection (TAD) which requires not only classifying each activity, but also determining the temporal bounds of the activities. TAD is a much more challenging problem than video activity recognition, for the activities of interest are buried in a long video sequence which is mostly dominated by temporally cluttered backgrounds and may contain zero, one or multiple activity instances of the same class or of different classes.

The state-of-the-art approaches to TAD have been directly inspired by the advances in object detection and are broadly categorized into two-stage and single-stage methods. The two-stage methods require two separate phases of proposal generation and classification, thus performing significantly slower compared to their single-stage counterparts which can combine both phases into a single step. However, irrespective of being two-stage or single-stage, most of the existing state-of-the-art methods for TAD (e.g., [4, 6, 17, 22, 25]) do not learn spatio-temporal feature representations of a video end-to-end, but rather extract deep features from short snippets either using 2D Convolutional Neural Networks (CNNs) (e.g., [29, 36]), or 3D CNNs (e.g., [5, 34]), followed by complex feature aggregation to provide the temporal modeling. Since these feature extractors are specifically optimized for image/video classification task, such off-the-shelf representations may not be optimal for localization of activities in diverse video domains. The handful of attempts towards providing an end-to-end TAD framework (e.g., [2, 28, 38]) are all based on the two-stage approach, thereby inheriting the drawbacks as mentioned above. Besides, their reliance on relatively shallow 3D CNNs (e.g., [34]) having limited temporal footprint essentially leads to unsatisfactory performance [5].

Moreover, recent state-of-the-art methods to TAD (e.g., [6, 17]) mostly rely on a two-stream architecture [29] to capture the appearance and motion information of a video which are usually integrated very late in the network at the *predictions* level. The limited body of research that suggested *mid-level* fusion (e.g., [10, 12, 20]) are all based on simple fusion strategies (e.g., *sum*, *max*, or *convolution*) making them inferior to temporal relationship modeling [19]. Besides, these methods mainly address the video activity recognition problem. Therefore, further research is warranted into exploring *how* best to fuse such multi-stream information while investigating *where* in the feature abstractions level such fusion would be appropriate in the context of temporal activity detection.

Realizing the above limitations, this paper attempts to fill in the gaps by proposing a single-stage end-to-end trainable framework for TAD that leverages multi-stream cues of a video based on sophisticated feature fusion strategies. Drawing inspiration from the leading single-stage object detector called SSD [24], we build a multi-scale temporal feature hierarchy atop a two-stream 3D CNN that learns task-specific appearance and motion features of a video in an end-to-end manner. With the two-stream feature representations, we systematically investigate different sophisticated fusion methods at different levels in the feature abstractions with a view to finding the optimal strategy (i.e., *how* and *where* to fuse). This leads us to propose a new fusion method based on efficient bilinear pooling [32] operation.

Contributions. Our contributions are three fold: (1) we propose a single-stage end-to-end approach to TAD; (2) we demonstrate effective ways to fuse multi-stream feature representations of a video and propose a new *mid-level* fusion strategy for TAD; (3) finally, we set new state-of-the-art on THUMOS'14 and MEXaction2 benchmarks.

2 Related Work

Temporal Activity Detection: The early two-stage approaches to TAD [3, 16, 27, 35, 41] could not afford end-to-end training either on the proposal or the classification phase. Inspired by the leading two-stage object detector called Faster-RCNN [26], some recent works (e.g., [7, 13, 14, 38]) offered end-to-end training by directly applying Faster-RCNN architecture to TAD. Most notably, R-C3D [38] closely followed this architecture by building upon C3D [34] to learn spatio-temporal feature representations end-to-end, while CMS-RC3D [2] extended R-C3D by incorporating a temporal feature pyramid. However, the performance of these methods is limited by the short temporal footprint of C3D, which has been addressed by a recent work called TAL-net [6] that leveraged longer temporal contexts by capitalizing on a 3D ConvNet called I3D [5] as the feature extractor network.

However, the impressive speed and accuracy of the single-stage object detectors such as SSD [24] led to the development of the recent TAD approaches that generate a multi-scale temporal feature pyramid based on a set of anchors of predefined (e.g., SSAD [22]) or learnable scales (e.g., GTAN [25]). Decouple-SSAD [17] improves upon SSAD by having two separate branches for classification and localization and achieves state-of-the-art performance on THUMOS'14 benchmark. However, none of these approaches could afford end-to-end feature learning as they are based on feature extraction and aggregation from multiple different 2D/3D CNNs. Our proposed approach is also based on SSD, but performs activity detection on top of task-specific spatio-temporal features that are learned end-to-end.

Multi-stream Feature Fusion: The two-stream network [29] is the pioneering work to propose *late* fusion by integrating the individual predictions of the appearance and motion streams for video activity recognition. Most state-of-the art approaches to video activity recognition (e.g., [5, 19, 33, 36, 37]) and temporal activity detection (e.g., [6, 17]) mainly follow this work. There have been only a handful of works that demonstrated optimal results using *mid-level* fusion. For example, Feichtenhofer *et al*. proposed *mid-level* fusion via concatenation of the two streams followed by convolution [10], or element-wise multiplication followed by residual connections [12]. EPIC-Fusion [20] fused audio, appearance, and motion information via concatenation followed by fully-connected layer for ego-centric action recognition. However, none of these works explored any sophisticated fusion strategies. In this work, we explore more sophisticated *mid-level* fusion strategies for TAD, while investigating *where* in the feature abstractions level such fusion would be appropriate.

3 Proposed Approach

Our primary goal is to design a single-stage end-to-end trainable TAD framework. To this end, we build on a two-stream architecture [29] – a spatial stream to model the appearance information from RGB frames, and a temporal stream to capture the motion contexts from pre-computed optical flow. This leads to the second goal which is to find the optimal strategy to fuse such multi-stream information in the context of TAD. We first explain our baseline architecture that uses *late* fusion in Sec. 3.1. We then explore different sophisticated fusion strategies and explain our proposed fusion method in Sec. 3.2 and 3.3.

3.1 Baseline Architecture

Analogous to SSD [24], we start with an activity recognition 3D CNN and transform it into an activity detection framework. Fig. 1 shows the overall architecture of the baseline TAD framework that consists of two separate but similar branches, one for each stream.

3D CNN Feature Extractor: The input to our network is a pair of video sequences $(\mathcal{I}_{\text{rgb}}, \mathcal{I}_{\text{flow}}) \in R^{T \times H \times W \times C}$ each consisting of T RGB and FLOW frames respectively. Each frame has height, width and channel dimensions of H, W, and C (3 for RGB, 2 for flow). $(\mathcal{I}_{\text{rgb}}, \mathcal{I}_{\text{flow}})$ are first passed through a two-stream 3D CNN to learn rich spatio-temporal feature representations end-to-end. The architecture of the 3D CNN is adopted from the state-of-the-art video activity recognition network S-3DG [37]. We extract feature maps $(\mathcal{F}_{\text{rgb}}, \mathcal{F}_{\text{flow}}) \in R^{\frac{T}{8} \times \frac{H}{32} \times \frac{W}{32} \times 1024}$ from the *Mixed_5c* block of S-3DG, pass them through *average pooling* to collapse the spatial dimensions, finally, apply 1D convolutions (kernel size 3, strides 1) to further extend the temporal receptive field to produce temporal-only two-stream feature maps $(\mathcal{F}^{'}_{\text{rgb}}, \mathcal{F}^{'}_{\text{flow}}) \in R^{\frac{T}{8} \times 1024}$ that serve as the base feature maps to perform activity detection. The feature extractor 3D CNN being *fully-convolutional*, the length of the video sequence T can be arbitrarily long.

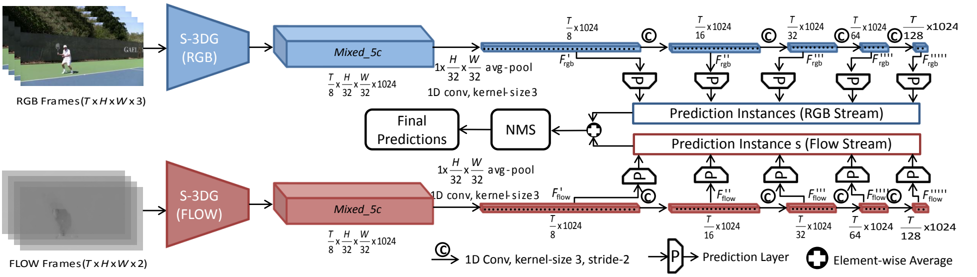

Figure 1: Adopting a two-stream architecture, we perform activity detection on clips of RGB and FLOW frames of length T which are first passed through a feature extractor 3D CNN, followed by a temporal feature hierarchy to generate multi-scale temporal feature maps. A set of 1D temporal convolutional filters are learned to generate activity predictions. Predictions from both streams are fused via *element-wise* averaging in the baseline approach. Final predictions are generated via Non-Maximum Suppression (NMS) of the fused predictions.

Two-stream Multi-scale Feature Hierarchy: Following [22], we build a two-stream feature hierarchy on top of the 3D CNN feature extractor. Starting with $(\mathcal{F}^{'}_{\text{rgb}}, \mathcal{F}^{'}_{\text{flow}}) \in R^{\frac{T}{8}\times 1024}$, we cascade four 1D convolutional layers with kernel size of 3 and strides of 2 to produce four two-stream feature maps with decreasing temporal resolution, namely, $(\mathcal{F}^{''}_{\text{rgb}}, \mathcal{F}^{''}_{\text{flow}}) \in R^{\frac{T}{16}\times 1024}; \cdots; (\mathcal{F}^{'''''}_{\text{rgb}}, \mathcal{F}^{'''''}_{\text{flow}}) \in R^{\frac{T}{128}\times 1024}$. These feature maps together form a two-stream multi-scale feature hierarchy $\mathcal{F}^{MS}_{\text{rgb}} = \{\mathcal{F}^{'}_{\text{rgb}}, \cdots, \mathcal{F}^{'''''}_{\text{rgb}}\}$, $\mathcal{F}^{MS}_{\text{flow}} = \{\mathcal{F}^{'}_{\text{flow}}, \cdots, \mathcal{F}^{'''''}_{\text{flow}}\}$.

Prediction Layers: In the baseline approach, predictions are made on each feature map belonging to the two-stream feature hierarchy $(\mathcal{F}^{MS}_{\text{rgb}}, \mathcal{F}^{MS}_{\text{flow}})$. Each prediction layer consists of 1D convolutional filters (kernel size 3, strides 1). Following [22], we associate K *default temporal segments* with each temporal location in a feature map, each having the same default center but different scale ratios $s_k \in \{s_1, s_2, \cdots, s_K\}$. For example, each temporal location $i^{'} \in [0, \frac{T}{8})$ in $\mathcal{F}^{'} \in R^{\frac{T}{8}\times 1024}$ has a base temporal scale of $\frac{1}{T/8}$. Therefore, the *default temporal segment* with scale ratio s_k at $i^{'}$ has the default center and default width as $d_c = \frac{i^{'}+0.5}{T/8}$ and $d_w = s_k \cdot \frac{1}{T/8}$, respectively. Each prediction layer, therefore, employs $K(C+3)$ filters at each temporal location to generate the following predictions: i) class scores $\{p_r\}_{r=1}^{C}$ over C activity classes including the *background* class; ii) the center and width offsets Δ_c and Δ_w relative to d_c and d_w, respectively; and iii) an overlap score p_{ov} indicating the overlap of the *default activity segment* with the closest ground-truth segment, which is passed through *sigmoid* function to produce a confidence value in the range $[0, 1]$. Δ_c and Δ_w are used to compute the actual center ϕ_c and actual width ϕ_w which are in turn used to compute the activity start time ϕ_{start} and end time ϕ_{end} as follows –

$$\phi_c = d_c + \alpha_1 d_w \Delta_c \quad \text{and} \quad \phi_w = d_w \exp(\alpha_2 \Delta_w) \tag{1}$$

$$\phi_{start} = \phi_c - \frac{\phi_w}{2} \quad \text{and} \quad \phi_{end} = \phi_c + \frac{\phi_w}{2} \tag{2}$$

where α_1 and α_2 are hyper-parameters used to control the effect of Δ_c and Δ_w, respectively.

***Late* Fusion:** The baseline approach generates the final predictions by averaging the two streams' individual predictions followed by Non-Maximum Suppression (NMS).

3.2 *'How'* to Fuse the Two Streams

As aptly reasoned in [11], to discriminate between activities having similar motion or appearance pattern (e.g., 'brushing teeth' vs. 'brushing hair'), it is necessary for the appearance and motion streams to interact earlier in the network. We, therefore, propose *mid-level* fusion. However, the existing works employ simple straightforward methods (e.g., *sum*, *max*, or *convolution*) for *mid-level* fusion that fail to capture the full correspondence between the different modalities, thereby leading to poor performance as demonstrated by our experiments. To this end, we are inspired by the latest advancement in *Visual Question Answering (VQA)* that makes efficient use of *bilinear pooling* [32] based methods to allow for high-level interactions between the different modalities. Below we discuss the traditional *mid-level* fusion methods followed by the efficient bilinear pooling based methods. Afterwards, we propose a new fusion method. For the purpose of the following discussions, we assume that $\mathcal{F}_{\text{rgb}} \in R^{T\times C}$, $\mathcal{F}_{\text{flow}} \in R^{T\times C^{'}}$ represent two input feature maps that need to be fused, where T and $C, C^{'}$ represent the temporal and channel dimensions respectively.

Sum, Max, and Convolution Fusion: These methods can be formulated as follows:

$$\mathcal{F}_{\text{sum}} = \mathcal{F}_{\text{rgb}} \oplus \mathcal{F}_{\text{flow}} \tag{3}$$

$$\mathcal{F}_{\text{max}} = \max(\mathcal{F}_{\text{rgb}}, \mathcal{F}_{\text{flow}}) \tag{4}$$

$$\mathcal{F}_{\text{conv}} = (\mathcal{F}_{\text{rgb}} \,\|\, \mathcal{F}_{\text{flow}}) * f + b \tag{5}$$

where $\mathcal{F}_{\text{sum}}, \mathcal{F}_{\text{max}} \in R^{T\times C}$; $\mathcal{F}_{\text{conv}} \in R^{T\times C^{\text{conv}}}$; $f \in R^{1\times(C+C^{'})\times C^{\text{conv}}}$ is a filter bank of C^{conv} filters; $b \in R^{C^{\text{conv}}}$ are biases. Here, $\oplus$, $\|$, and $*$ represent element-wise addition, channel-wise concatenation and cahnnel-wise convolutions respectively. We set $C^{\text{conv}} = C = C^{'}$.

Multi-modal Low-rank Bilinear Pooling (MLB) [21]: MLB tries to reduce the computational complexity of the bilinear pooling operation by factoring a three-dimensional weight tensor of bilinear pooling into three two-dimensional weight tensors. MLB first non-linearly projects the input feature maps into a common embedding space where they are fused via element-wise multiplication, which is then followed by a linear projection as follows:

$$\mathcal{F}_{\text{MLB}} = (\sigma(\mathcal{F}_{\text{rgb}}U) \odot \sigma(\mathcal{F}_{\text{flow}}V))P \tag{6}$$

where $\mathcal{F}_{\text{MLB}} \in R^{T\times C^{\text{MLB}}}$; $U \in R^{C\times D}$, $V \in R^{C^{'}\times D}$, $P \in R^{D\times C^{\text{MLB}}}$ are the weight tensors, $D = min(C, C^{'})$ is the dimensionality of the common embedding space. Here, $\odot$ and σ represent element-wise multiplication and non-linear activation respectively. We set $C^{\text{MLB}} = C = C^{'}$.

Multi-modal Factorized Bilinear Pooling (MFB) [39]: MFB provides an improvement over MLB. As shown in Fig. 2(a), MFB first projects the input feature maps into a higher dimensional space, then fuses them via element-wise multiplication followed by Dropout and SumPooling as follows:

$$\mathcal{F}_{\text{MFB}} = \text{SumPooling}(\text{Dropout}(\mathcal{F}_{\text{rgb}}U \odot \mathcal{F}_{\text{flow}}V), J) \tag{7}$$

where $\mathcal{F}_{\text{MFB}} \in R^{T\times C^{\text{MFB}}}$; $U \in R^{C\times D}, V \in R^{C^{'}\times D}$ are the weight tensors, $D = J \times C^{\text{MFB}}$ is the dimensionality of the embedding space. Here, J is the window-size for sum-pooling and $\odot$ represents element-wise multiplication. We set $C^{\text{MFB}} = C = C^{'}$. As explained in [39], the output of the MFB fusion are passed through power normalization ($z \leftarrow sign(z)|\, z\,|^{0.5}$) followed by l_2-normalization ($z \leftarrow z/\|z\|$) to avoid unsatisfactory local minima during training.

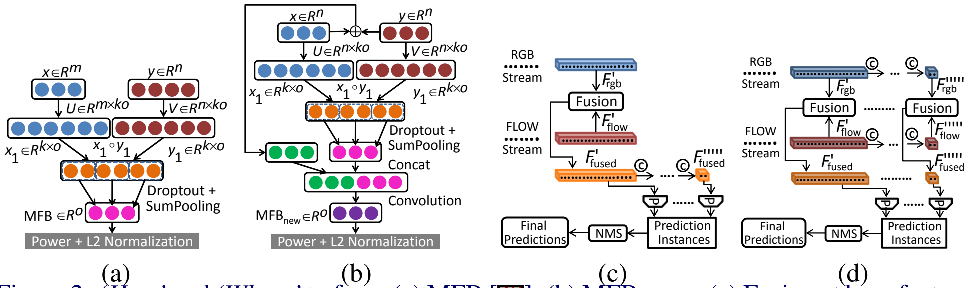

Figure 2: *'How'* and *'Where'* to fuse: (a) MFB [39]; (b) MFB_new; (c) Fusion at base feature maps $(\mathcal{F}^{'}_{\text{rgb}}, \mathcal{F}^{'}_{\text{flow}})$; (d) Fusion at feature hierarchies $(\mathcal{F}^{MS}_{\text{rgb}}, \mathcal{F}^{MS}_{\text{flow}})$. Legends from Fig. 1.

Proposed Fusion Method: We build on the MFB fusion. However, unlike MFB, we non-linearly project the input feature maps into the higher dimensional space. We, further, boost the fused feature map by having it convolved with the element-wise summation of the original feature maps as shown in Fig. 2(b). This essentially serves as a residual connection as it allows the original input feature maps to directly interact with the transformed fused feature map, thereby providing better representation capacity than MFB. We denote the proposed fusion approach as MFB_new which is formulated as follows:

$$z = \text{SumPooling}(\text{Dropout}(\sigma(\mathcal{F}_{\text{rgb}}U) \odot \sigma(\mathcal{F}_{\text{flow}}V)), J) \tag{8}$$

$$\mathcal{F}_{\text{MFB\_new}} = ((\mathcal{F}_{\text{rgb}} \oplus \mathcal{F}_{\text{flow}}) \,\|\, z) * f + b \tag{9}$$

where $\mathcal{F}_{\text{MFB\_new}} \in R^{T \times C^{\text{new}}}$; and U, V, and J represent the same as in MFB, whereas, $f \in R^{1\times(C+C^{\text{new}})\times C^{\text{new}}}$ is a filter bank of C^{new} filters and $b \in R^{C^{\text{new}}}$ are biases. For the proposed fusion, we set $C^{\text{new}} = C = C^{'}$. Similar to MFB, we pass the output of MFB_new through the normalization steps as explained above.

3.3 *'Where'* to Fuse the Two Streams:

We consider *mid-level* fusion at different feature abstraction levels with a view to finding the optimal level to fuse such multi-stream information in the context of TAD. In particular, we explore the base feature maps $(\mathcal{F}^{'}_{\text{rgb}}, \mathcal{F}^{'}_{\text{flow}})$, as well as the multi-scale feature hierarchies $(\mathcal{F}^{MS}_{\text{rgb}}, \mathcal{F}^{MS}_{\text{flow}})$ as the potential candidates for fusion.

Fusion at Base Feature Map $\mathcal{F}^{'}$: Fig. 2(c) shows how to integrate the two-streams at the base feature map level $(\mathcal{F}^{'}_{\text{rgb}}, \mathcal{F}^{'}_{\text{flow}})$ to produce a fused base feature map $\mathcal{F}^{'}_{\text{fused}}$ which is then used to generate the multi-scale feature hierarchy, the input to the prediction layers.

Fusion at Multi-scale Feature Hierarchy $\mathcal{F}^{MS}$: Fig. 2(d) shows how to integrate the two-streams at the multi-scale feature hierarchy level $(\mathcal{F}^{MS}_{\text{rgb}}, \mathcal{F}^{MS}_{\text{flow}})$ to produce a fused multi-scale feature hierarchy $\mathcal{F}^{MS}_{\text{fused}}$. Predictions are performed on each feature map in $\mathcal{F}^{MS}_{\text{fused}}$.

3.4 Training and Inference:

Ground-truth Matching and Hard-negative Mining: During training, each of the K predictions at a temporal location is labeled as *positive* if its tIoU overlap with any of the ground-truth segments is greater than 0.5, otherwise *negative*. Following [22], we also adopt *hard negative mining* to keep the ratio of positive to negative samples as 1:1.

Loss Function: We use a multi-task loss $\mathcal{L}$ including a classification loss $\mathcal{L}_{\text{cls}}$ (Softmax loss), a localization loss $\mathcal{L}_{\text{loc}}$ (Smooth-L1 loss), and tIoU overlap loss $\mathcal{L}_{\text{ov}}$ (Smooth-L1 loss):

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \beta \mathcal{L}_{\text{loc}} + \gamma \mathcal{L}_{\text{ov}} \tag{10}$$

where β and γ are hyper-parameters used to trade-off among the three different losses.

Inference: A prediction instance is made up as $\psi = \{\phi_{start}, \phi_{end}, R, p_f\}$, where, $R = \text{argmax}(\{p_r\}_{r=1}^{C})$ denotes the final class prediction, and $p_f = \max(\{p_r\}_{r=1}^{C}) \cdot p_{ov}$ is the final confidence score which is used to conduct *NMS* to remove any redundant predictions.

4 Experiments

4.1 Experimental Setup

Datasets: We conducted experiments on two activity detection benchmarks – THUMOS'14 [18] and MEXaction2 [1]. THUMOS'14 contains over 22 hours of video from different sports activities and is very challenging, for each video is more than 3 minutes long and has 15 activity instances on average. The validation set and test set contain 200 and 213 temporally untrimmed videos, respectively with annotations for 20 different activity classes. Following the standard practice, we perform training on the validation set (with a 90-10 split for hyperparameter tuning) and report results on the test set.

The MEXaction2 [1] dataset, on the other hand, consists of approximately 77 hours of untrimmed videos from The Institut national de l'audiovisuel (abbreviated as INA), plus some trimmed video clips from YouTube and UCF101 [30]. There are only two activity categories – "HorseRiding" and "BullChargeCape".

Data Preparation Following [22], we perform activity detection on clips of length T by densely sampling frames from each untrimmed video. During training, neighboring clips are generated with a 75% overlap to handle activity instances located near the boundary and to increase the amount of training data. During inference, the overlap is limited to 50% for faster processing. We set T=512 as approximately 99% activity instances in both datasets are shorter than 512 frames.

Implementation Details: We initialize RGB and FLOW branches of the 3D CNN feature extractor with the pre-trained weights of D3D [31] and S-3DG [37] respectively. Optical flow is computed using TV-L1 algorithm [40]. We set $K = 5$ with scale ratios {0.5, 0.75, 1.0, 1.5, 2.0}, sum-pooling window-size $J = 2$, dropout ratio = 0.2 (for MFB and MFB_new), α_1 and α_2 to 0.1, β and γ to 10, and tIoU threshold for *NMS* to 0.2 via cross-validation.

Evaluation Metrics: Apart from TAD, we also evaluate our approach on another relevant task called Temporal Activity Proposal (TAP). We use the standard performance measure called mean Average Precision (mAP) for TAD, while for TAP, we use the standard metric called AR-AN. AR-AN is computed as Average Recall (AR) at different average number of proposals per video (AN). AR is defined to be the average of the recall values computed at tIoU's from 0.5 to 1.0 with a step size of 0.05.

4.2 Results

Evaluation on *'How'* and *'Where'* to Fuse: In order to evaluate the different *mid-level* fusion strategies as discussed in Sec. 3.2 and 3.3, we trained models on THUMOS'14 by extracting two-stream features $(\mathcal{F}'_{rgb}, \mathcal{F}'_{flow})$ from the fixed 3D CNN, followed by fusion of

How to Fuse	***Where*** to Fuse		
	Mid-level		*Late*
	$\mathcal{F}'$	$\mathcal{F}^{MS}$	
Averaging			46.42
Sum	47.08	48.48	
Max	46.09	47.53	
Convolution	46.86	47.45	
MLB	46.65	47.77	
MFB	38.29	49.85	
MFB_new	37.47	**50.88**	

Table 1: mAP(%) for *Mid-level* and *Late* fusion strategies on THUMOS'14.

Model	*Fixed*	*5c*	*5b_5c*
Baseline	46.42	47.48	48.97
MFB	49.85	51.39	52.60
MFB_new	**50.88**	**52.08**	**53.95**
RGB	-	-	47.84
FLOW	-	-	48.56

Table 2: mAP(%) for End-to-End Training on THUMOS'14. Top: Fused models. Bottom: Single-stream models.

Model	
[17]	44.20
[17]+MFB	47.25
[17]+**MFB_new**	**48.27**

Table 3: mAP(%) Comparisons of Decouple-SSAD [17] with its variants on THUMOS'14.

the two streams using the network architectures shown in Fig. 2(c)-(d). As Tab. 1 shows, the *mid-level* fusion strategies perform better than *late* fusion. This can be explained by the fact that *mid-level* fusion utilizes the correlation between features from different modalities, thus making them more discriminative in the common feature space than their individual feature space [8]. Furthermore, the proposed fusion methods outperform the conventional methods (e.g., *sum, max, convolution*), while fusion at the $\mathcal{F}^{MS}$ level produces better results than $\mathcal{F}'$. However, performance gain is more strongly pronounced for MFB and MFB_new with the latter outperforming the other methods, thereby attesting to its superior representation capacity. Subsequent discussions refer to MFB and MFB_new models fused at the $\mathcal{F}^{MS}$ level.

Evaluation on End-to-End Feature Learning: To demonstrate the effect of end-to-end feature learning, we consider three different training configurations on THUMOS'14: i) *Fixed* does not train the feature extractor 3D CNN just like the models shown in Tab. 1; ii) *5c* trains only the last convolutional layer of the 3D CNN (i.e., *'Mixed_5c'*); and, iii) *5b_5c* trains the last two convolutional layers (i.e.; *'Mixed_5b', 'Mixed_5c'*). The top part of Tab. 2 shows the performance of the baseline model that uses *late* fusion, as well as the models based on MFB and MFB_new fusion under these training configurations. As evidenced from the table, end-to-end feature learning improves performance by a noticeable margin for all 3 models, thus validating our design choice for end-to-end learning, something the existing state-of-the-art single-stage approaches (e.g., [17, 22, 25]) could not afford. It is noteworthy to mention that we did not find any performance improvement by training more layers which could be attributed to the fact that the amount of training data for THUMOS'14 is not sufficient to train all layers of a 3D CNN as also reported in [34]. All subsequent comparisons are based on the models trained using *5b_5c* configuration.

Ablation Study: To validate our design choice for using both RGB and FLOW streams, we train models on THUMOS'14 based on the individual streams and compare the results with the fused models. As shown at the bottom part of Tab. 2, the FLOW-only model performs better than the RGB-only, whereas, the fused models outperform the single-stream models. These results are in agreement with common observation in video activity recognition (e.g., [5, 11, 29, 36]), thus validate our design choice for a fused two-stream network.

We, further, investigate the effects of the proposed fusion strategies across other methods

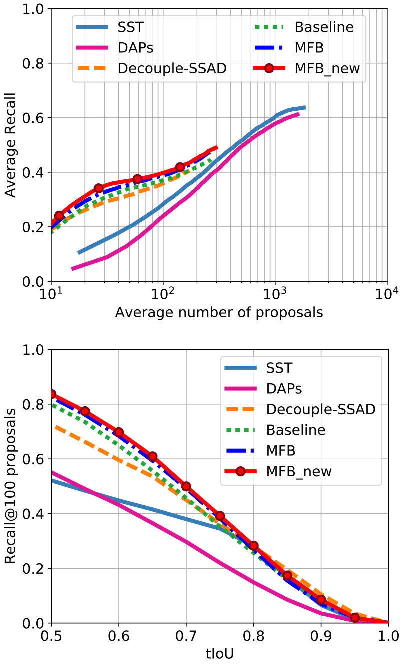

Figure 3: AR-AN curve (top) and Recall@100 vs. tIoU curve (bottom) on THUMOS'14.

Stage	Method	mAP @tIoU (%)						
		0.1	0.2	0.3	0.4	0.5	0.6	0.7
Two-Stage	SCNN [27]	47.7	43.5	36.3	28.7	19.0	10.3	5.3
	DAPs [9]	-	-	-	-	13.9	-	-
	SST [3]	-	-	41.2	31.5	20.0	10.9	4.7
	TCN [7]	-	-	-	33.3	25.6	15.9	9.0
	R-C3D [38]	54.5	51.5	44.8	35.6	28.9	19.1	9.3
	SSN [41]	66.0	59.4	51.9	41.0	29.8	19.6	10.7
	CBR [14]	60.1	56.7	50.1	41.3	31.0	19.1	9.9
	BSN [23]	-	-	53.5	45.0	36.9	28.4	20.0
	TAL-net [6]	59.8	57.1	53.2	48.5	42.8	33.8	**20.8**
Single-Stage	SSAD [22]	50.1	47.8	43.0	35.0	24.6	15.4	7.7
	SS-TAD [4]	-	-	45.7	-	29.2	-	9.6
	G-TAN [25]	69.1	63.7	57.8	47.2	38.8	-	-
	DSSAD [17]	-	-	60.2	54.1	44.2	32.3	19.1
	Baseline	69.5	68.4	66.1	61.1	49.0	32.9	16.7
	MFB	70.7	69.6	67.1	61.9	52.6	35.5	18.0
	MFB_new	**73.0**	**71.9**	**69.2**	**65.0**	**53.9**	**38.1**	19.8

Table 4: State-of-the-Art mAP(%) Comparisons on THUMOS'14

SCNN [27]	SSAD [22]	**MFB_new**
7.4	11.0	**16.4**

Table 5: State-of-the-Art mAP(%) Comparisons on MEXaction2 at tIoU=0.5.

for TAD. To this end, we choose Decouple-SSAD [17], the existing state-of-the-art on THUMOS'14, which is also based on a multi-scale feature hierarchy. Using the author-provided code, we train Decouple-SSAD by replacing its *late* fusion strategy with *mid-level* fusion based on MFB and MFB_new. Table 3 compares the performance of the original Decouple-SSAD with its variants. As evidenced from the table, the proposed fusion strategies are able to provide performance boost across other methods, thus proving themselves to be generic enough.

State-of-the-Art Comparisons – Temporal Activity Proposal: We compare our proposed models with two state-of-the-art activity proposal generation methods on THUMOS'14 called DAPs [9] and SST [3], along with the current state-of-the-art TAD method on THUMOS'14 called Decouple-SSAD [17]. As Fig. 3 (top) depicts, at low AN values, our proposed models (Baseline, MFB, and MFB_new) outperform the other methods, with MFB_new delivering the best performance. This clearly suggests that our top predictions are much more likely to contain activity segments. However, just like Decouple-SSAD [17], our AR saturates more quickly than the two proposal generation methods, which is mainly due to the low average number of predictions generated per video by our method.

To further zoom into the boundary quality of the predicted activity segments, we plot AR values for the top 100 predictions against higher tIoU thresholds as shown in Fig. 3 (bottom). MFB_new outperforms all other methods for most of the thresholds.

State-of-the-Art Comparisons – Temporal Activity Detection: As Tab. 4 shows, our proposed models (Baseline, MFB, and MFB_new) consistently and significantly outperform

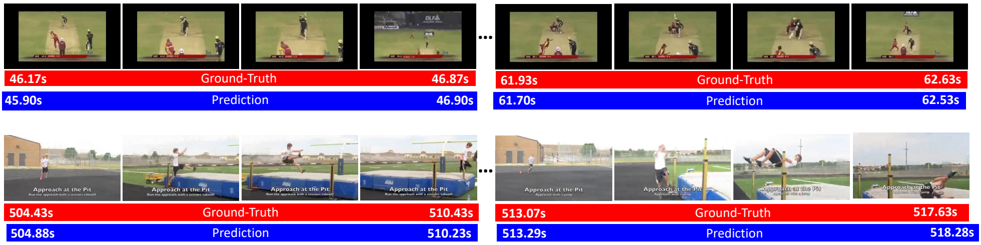

Figure 4: Visualization of the top predicted activity instances on two test videos from THUMOS'14. For each video, the first row shows some representative frames from two consecutive ground-truth activity segments, while the second and third rows represent the ground-truth (in red) and the predictions (in blue) of our model MFB_new, respectively.

Method	Baseball Pitch	Basketball Dunk	Billiards	Clean and Jerk	Cliff Diving	Cricket Bowling	Cricket Shot	Diving	Frisbee Catch	Golf Swing	Hammer Throw	High Jump	Javelin Throw	Long Jump	Pole Vault	Shotput	Soccer Penalty	Tennis Swing	Throw Discus	Volleyball Spiking
[17]	33.2	28.3	7.6	**48.1**	56.9	14.0	5.5	58.8	11.9	42.1	**75.2**	76.6	**83.8**	**94.6**	84.0	40.8	15.1	9.3	71.4	16.8
MFB_new	**43.0**	**54.6**	**8.0**	44.6	**75.9**	**43.7**	**28.7**	**76.7**	**34.9**	**57.8**	73.0	**79.9**	76.1	82.5	**84.6**	**41.6**	**32.9**	**24.4**	**80.5**	**35.5**

Table 6: Class-wise AP(%) comparisons between Decouple-SSAD [17] and MFB_new at tIoU=0.5 on THUMOS'14. MFB_new outperforms Decouple-SSAD in 16 out of 20 categories.

the other methods across different tIoU thresholds for the temporal activity detection task on THUMOS'14. Of particular interest is tIoU=0.5, MFB_new outperforms the current state-of-the-art Decouple-SSAD [17] by an absolute **9.7% mAP (53.9% vs. 44.2%)**. Class-wise AP comparisons between MFB_new and Decouple-SSAD [17] is shown in Tab. 6.

Results on MEXaction2: Table 5 reports results on MEXaction2. We compare MFB_new with SCNN [27] and SSAD [22], as these are the state-of-the-art methods on this dataset. We outperform both of these methods pushing the current state-of-the-art from **11%** to **16.4% mAP**.

Qualitative Results: Figure 4 shows some qualitative results of our proposed model on THUMOS'14. As we can see, MFB_new can accurately classify and localize the activity segments and is able to handle moderate variations in activity duration (e.g.; short and long activity instances).

Inference Speed: Being end-to-end and single-stage in nature, our proposed approach runs at a moderate speed. With pre-computed optical flow frames, we find that MFB_new operates at 758 frames per second on Nvidia Titan Xp GPU. It is noteworthy to mention that in our current implementation, the post-processing step (i.e.; NMS) does not run on GPU, thus leaving further room for improvement.

5 Conclusion

In conclusion, we have presented an end-to-end trainable activity detection framework. We have also demonstrated effective ways to fuse multi-stream feature representations of a video and proposed a new fusion method for temporal activity detection. Our proposed approach achieves state-of-the-art results on THUMOS'14 and MEXaction2 benchmarks.

References

[1] Mexaction2. http://mexculture.cnam.fr/Datasets/mex+action+dataset.html. Accessed: 2020-04-01.

[2] Yancheng Bai, Huijuan Xu, Kate Saenko, and Bernard Ghanem. Contextual multi-scale region convolutional 3d network for activity detection. *ArXiv*, abs/1801.09184, 2018.

[3] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles. Sst: Single-stream temporal action proposals. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[4] Shyamal Buch, Victor Escorcia, Bernard Ghanem, Li Fei-Fei, and Juan Carlos Niebles. End-to-end, single-stream temporal action detection in untrimmed videos. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2017.

[5] 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.

[6] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A. Ross, Jun Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[7] Xiyang Dai, Bharat Singh, Guyue Zhang, Larry Davis, and Yan Qiu Chen. Temporal context network for activity localization in videos. 2017. doi: 10.1109/ICCV.2017.610.

[8] Srijan Das, Monique Thonnat, Kaustubh Sakhalkar, Michal F Koperski, Francois Bremond, and Gianpiero Francesca. A New Hybrid Architecture for Human Activity Recognition from RGB-D videos. In *MMM 2019 - 25th International Conference on MultiMedia Modeling*, 2019. URL https://hal.inria.fr/hal-01896061.

[9] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Daps: Deep action proposals for action understanding. In *ECCV*, 2016.

[10] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[11] C. Feichtenhofer, A. Pinz, and R. P. Wildes. Spatiotemporal multiplier networks for video action recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[12] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In *Advances in Neural Information Processing Systems 29*. 2016.

[13] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[14] Jiyang Gao, Zhenheng Yang, and Ramakant Nevatia. Cascaded boundary regression for temporal action detection. *ArXiv*, abs/1705.01180, 2017.

[15] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[16] Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] Yupan Huang, Qi Dai, and Yutong Lu. Decoupling localization and classification in single shot temporal action detection. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 2019.

[18] Haroon Idrees, Amir Roshan Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The THUMOS challenge on action recognition for videos "in the wild". *CoRR*, abs/1604.06182, 2016.

[19] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[20] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[21] Jin-Hwa Kim, Kyoung Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard product for low-rank bilinear pooling. In *5th International Conference on Learning Representations, ICLR*, 2017.

[22] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. In *Proceedings of the 25th ACM International Conference on Multimedia*, 2017.

[23] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *The European Conference on Computer Vision (ECCV)*, 2018.

[24] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.

[25] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, 2015.

[27] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 2016.

[28] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. 2017. doi: 10.1109/CVPR.2017.155.

[29] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, 2014.

[30] Khurram Soomro, Amir Roshan Zamir, Mubarak Shah, Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, 2012.

[31] Jonathan C. Stroud, David A. Ross, Chen Sun, Jia Deng, and Rahul Sukthankar. D3d: Distilled 3d networks for video action recognition. *ArXiv*, abs/1812.08249, 2018.

[32] Joshua B. Tenenbaum and William T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 2000.

[33] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[34] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.

[35] L. Wang, Y. Qiao, X. Tang, and L. Van Gool. Actionness estimation using hybrid fully convolutional networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[36] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Val Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[37] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *The European Conference on Computer Vision (ECCV)*, 2018.

[38] H. Xu, A. Das, and K. Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[39] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[40] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l1 optical flow. In *Proceedings of the 29th DAGM Conference on Pattern Recognition*, 2007.

[41] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.