# Dance with Flow: Two-in-One Stream Action Detection

Jiaojiao Zhao and Cees G. M. Snoek
University of Amsterdam

## Abstract

*The goal of this paper is to detect the spatio-temporal extent of an action. The two-stream detection network based on RGB and flow provides state-of-the-art accuracy at the expense of a large model-size and heavy computation. We propose to embed RGB and optical-flow into a single two-in-one stream network with new layers. A motion condition layer extracts motion information from flow images, which is leveraged by the motion modulation layer to generate transformation parameters for modulating the low-level RGB features. The method is easily embedded in existing appearance- or two-stream action detection networks, and trained end-to-end. Experiments demonstrate that leveraging the motion condition to modulate RGB features improves detection accuracy. With only half the computation and parameters of the state-of-the-art two-stream methods, our two-in-one stream still achieves impressive results on UCF101-24, UCFSports and J-HMDB.*

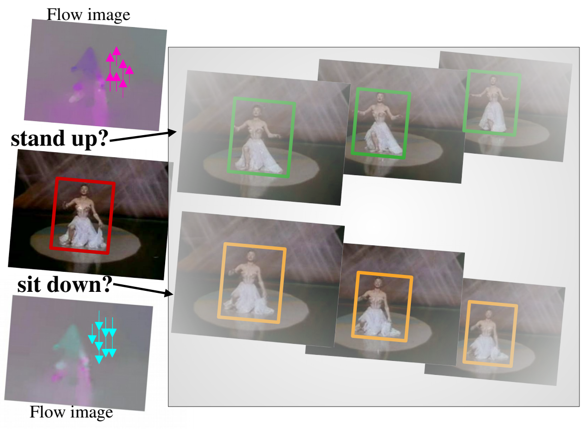


Figure 1: **Two-in-one stream.** We propose to embed RGB and optical-flow into a single stream for spatio-temporal action detection. Besides efficiency gains, it helps recognizing whether the dancer in the current frame is standing up or sitting down without considering the future. By utilizing information from flow images, the dancer is given a moving direction, up or down, better indicating the action.

## 1. Introduction

This paper strives for the spatio-temporal detection of human actions in video, which is a crucial ability for self-driving cars, autonomous care robots, and advanced video search engines. The leading approach for this challenging problem relies on fast detectors at the frame level [29, 37], which are then linked [1, 13, 37] or tracked [48] over time. Kalogeiton *et al*. [21] and Singh *et al*. [36] further showed it is advantageous to stack the features from subsequent frames before predicting action class scores and determining the enclosing tube. Most of the state-of-the-art action detectors exploit a two-stream architecture [35], one for RGB and one for optical-flow, which are individually trained before fusion. However, the double computation and parameter demand of two-stream methods does not lead to double accuracy compared to a single stream. We propose to embed RGB and optical-flow into a single stream for action detection.

We are inspired by progress on feature normalization, especially conditional normalization [8, 10, 18], which has been successfully employed to visual question answering [5], visual reasoning [30], image style transfer [17] and super-resolution [47]. Peretz *et al*. [30] propose a feature-wise linear modulation layer which enables a recurrent neural network over an input question to influence convolutional neural network computation over an image. It demonstrates that features are capable to be modulated via a simple feature-wise affine transformation based on conditioning information. However, as their modulation layer is agnostic to spatial location, it is unsuited for action detection. In [47], Wang *et al*. developed a spatial feature transform layer, which is conditioned on categorical semantic probability maps, to modulate a super-resolution network. Encouraged by these works, we propose a motion condition layer and a motion modulation layer to adjust an RGB-stream for spatio-temporal action detection.

We make the following contributions in this paper. We propose to embed RGB and optical-flow into a single stream for spatio-temporal action detection. It reduces the computational costs of conventional two-stream detection networks by half while maintaining its high accuracy. We introduce the two-in-one stream with motion condition layer and motion modulation layer, which learns video representations of appearance-stream features conditioned on optical-flow. As shown in Figure 1, the motion condition will guide the model to pay more attention on what moves, rather than the static background. The method is easily embedded in existing appearance- or two-stream action detection networks, and trained end-to-end, leading to new state-of-the-art on *UCF101-24*, *UCFSports* and *J-HMDB*.

## 2. Related Work

The spatio-temporal detection of human actions in video has a long tradition in computer vision, *e.g.* [3, 4, 22]. Early success came from detection based on exhaustive cuboid search, efficient feature representations, and SVM-based learning, [42, 43, 52]. This was later extended with more flexible sequences of bounding boxes [24,44,51], or spatio-temporal proposals [19, 45], together with engineered appearance and motion features, most notably the dense trajectories [28]. The past few years, architectures integrating detection and deep representation learning have been leading [7,11,16,25,41,49,50], typically combining appearance and flow streams [14, 15, 21, 37]. We follow this tradition.

The two-stream network was first introduced by Simonyan and Zisserman in [35]. Their convolutional architecture included a separate RGB-stream and a flow-stream, which were combined by late fusion, for SVM-based action classification. In [9], Feichtenhofer *et al.* investigated a number of ways to fuse the RGB and flow streams in order to best take advantage of their fused representation for action classification. While we concentrate on action detection in the paper, we are interested in RGB and flow as well, but rather than combining the two streams in a late fusion, we prefer a single stream.

Gkioxari and Malik [13] introduced a two-stream architecture with R-CNN detectors in action detection. They fused features from the last layer of an RGB- and a flow-stream, and then trained action specific SVM classifiers. A Viterbi algorithm [40] was adopted to link the detection boxes per frame into a tube. Weinzaepfel *et al.* [48] also used a two-stream R-CNN detector but replaced the linking by a tracking-by-detection method. Both methods are not end-to-end trainable and restricted to trimmed videos.

End-to-end two-stream detectors based on faster-RCNN were proposed in [29, 34]. In [29], Peng and Schmid performed region of interest pooling and score fusion to incorporate an RGB-stream and a flow-stream. In [16], Hou *et al.* extended 2D region of interest pooling to 3D tube-of-interest pooling with 3D convolutions, which directly generate tubelets for action detection. Singh *et al.* adopted a two-stream single-shot-multibox detector (SSD) [27] for realizing real time detection in [37]. Singh *et al.* [36] also introduced a transition matrix to generate a set of action proposals on pairs of frames. Kalogeiton *et al.* [21] proposed to exploit temporal continuity by taking as much as six frames as input for their single-shot multibox detector, leading to state-of-the-art results. In this paper, we take the single-shot multibox detector network as our backbone, using single [37] or multiple [21] frames as input, but rather than separating the streams for RGB and flow we introduce a single two-in-one stream.

Li *et al.* [26] proposed an action detector using an LSTM architecture with motion-based attention. Our two-in-one stream not only takes motion as attention, which helps to locate actions, but also uses motion to modulate RGB features which helps to better classify actions. Moreover, our method is easily embedded in existing appearance- or two-stream action detection and classification networks.

## 3. Two-in-One Network

We define the RGB-stream network $\mathcal{D}_\theta^{rgb}$ trained on single frame for spatio-temporal action detection as:

$$(L^{rgb}, S^{rgb}) = \mathcal{D}_\theta^{rgb}(I^{rgb}) \tag{1}$$

where $I^{rgb} \in \mathbb{R}^{H\times W\times 3}$ is a single RGB frame of height $H$ and width $W$ which is the input for the network $\mathcal{D}_\theta^{rgb}$. $L^{rgb} \in \mathbb{R}^{Q\times 4}$ and $S^{rgb} \in \mathbb{R}^{Q\times(P+1)}$ are $Q$ box locations and corresponding box classification scores for $P$ action classes and a background class. $\theta$ represents the parameters of the learned network. Similarly, we define a flow-stream network on single frame for spatio-temporal action detection as:

$$(L^{of}, S^{of}) = \mathcal{D}_\theta^{of}(I^{of}) \tag{2}$$

$I^{of} \in \mathbb{R}^{H\times W\times 2}$ is a single optical flow image with $x$ and $y$ components of the velocity respectively in two channels. The two-stream method includes training the two networks $\mathcal{D}_\theta^{rgb}$ and $\mathcal{D}_\theta^{of}$ independently, and fuses the results $(L^{rgb}, S^{rgb})$ and $(L^{of}, S^{of})$.

**Motion condition layer.** In our method, $I^{of}$ is regarded as a motion map with the same resolution as the corresponding RGB image $I^{rgb}$. We take $I^{of}$ as prior information $\Psi$ when applying an RGB-stream network $\mathcal{D}_\theta^{rgb}$ to estimate where and what actions may occur. Then we formulate our two-in-one network as a condition network:

$$\begin{aligned}(L^{\twoheadrightarrow}, S^{\twoheadrightarrow}) &= \mathcal{D}_\theta^{\twoheadrightarrow}(I^{rgb}|\Psi) \\ &= \mathcal{D}_\theta^{\twoheadrightarrow}(I^{rgb}|\mathcal{MC}(I^{of}))\end{aligned} \tag{3}$$

$$\Psi = \mathcal{MC}(I^{of}) = \mathcal{MC}((I^{of_x}, I^{of_y})) \tag{4}$$

$\twoheadrightarrow$ means two-in-one stream, $\mathcal{MC}(\bullet)$ is a mapping function to generate simple features from the flow images. So the two-in-one stream $\mathcal{D}_\theta^{\twoheadrightarrow}$ learns a model conditioned on motion information by a motion condition layer.

**Motion modulation layer.** We introduce a motion modulation ($\mathrm{M}^2$) layer to modify the features learned from RGB

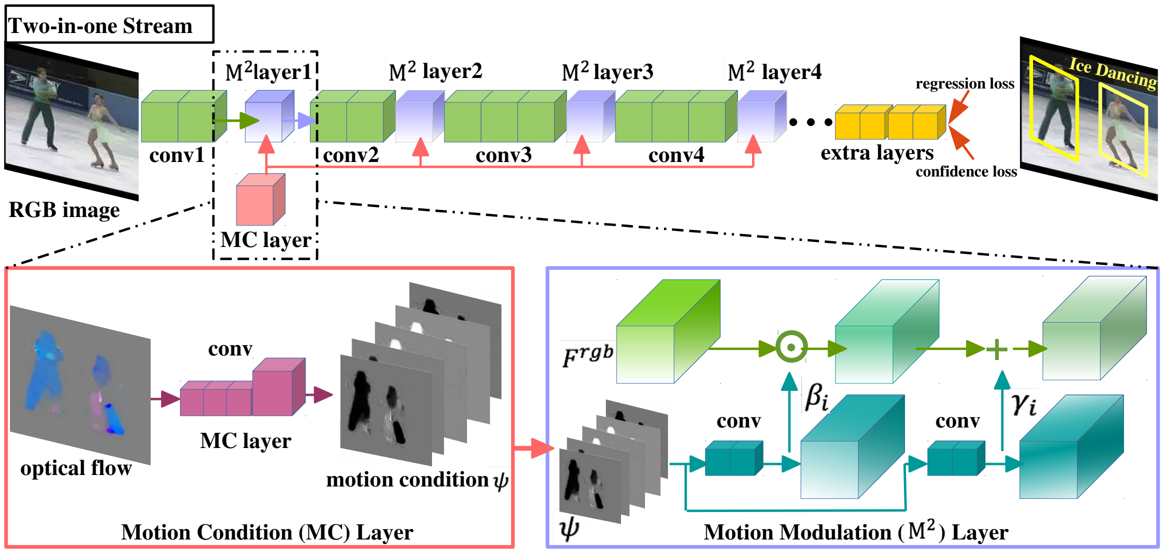


Figure 2: **Two-in-one network architecture**. The motion condition layer (pink cube) maps flow images to prior condition information. The condition inputs to the motion modulation layer (purple cube) to generate transformation parameters which are used to modulate RGB features ($F^{rgb}$). The network has half the computation and parameters of a two-stream equivalent, while obtaining better action detection accuracy.

images. An M$^2$ layer is able to influence the appearance network by incorporating motion and weighting the action area. We first learn a pair of affine transformation parameters $(\beta, \gamma)$ from the prior flow condition $\Psi$ by a function $\mathcal{F} : \Psi \longmapsto (\beta, \gamma)$. Concretely, the two-in-one network is further expressed as:

$$\begin{aligned} (\beta, \gamma) &= \mathcal{F}(\Psi), \\ (L^{\twoheadrightarrow}, S^{\twoheadrightarrow}) &= \mathcal{D}_{\theta}^{\twoheadrightarrow}(I^{rgb}|\beta, \gamma) \end{aligned} \tag{5}$$

In order to modulate the appearance network, we apply a transform function $\mathcal{M}^2(\bullet)$ with the learned transformation parameters $(\beta, \gamma)$ to the RGB features $F^{rgb}$.

$$\mathcal{M}^2(F^{rgb}) = \beta \odot F^{rgb} + \gamma \tag{6}$$

$\odot$ is an element-wise multiplication operation. The RGB feature maps $F^{rgb}$ has the same dimensions with parameters $\beta$ and $\gamma$. The flow information represented by $(\beta, \gamma)$ influences the appearance network by both feature-wise and spatial-wise manipulations. The complete network with the motion condition layer and the motion modulation layer is shown in Figure 2.

**Network architecture.** Due to sparsity of flow images, we adopt simple convolutional layers to extract low-level motion condition information. $1 \times 1$ convolutional layers attempt to keep the spatial pixel-wise motion vectors. The motion condition then inputs to a motion modulation (M$^2$) layer in which it is separately mapped to a pair of transformation parameters $\beta$ and $\gamma$. Two groups of $1 \times 1$ convolutional layers are independently adopted for generating each of the parameters $\beta$ and $\gamma$. The low-level RGB features from the appearance network are adjusted by $\beta$ and $\gamma$. The motion modulation layer is capable to be added to any bottom layer of the appearance network, including conv1, conv2, conv3 and conv4. All of them share the motion condition layer. The whole network is end-to-end trainable.

**Feature visualization.** In order to intuitively understand the method, we show the generated feature maps from the appearance network before and after modulation by motion condition in Figure 3. We randomly select some feature maps from the motion condition layer in the first row. The features are low-level and sparse, which are taken as prior conditions. From the second row to the last row, we show the corresponding scale ($\beta$) and shift ($\gamma$) maps generated from conditions, RGB features without modulation and features modulated by $\beta$ and $\gamma$. It is interesting to see the difference between the features without and with modulation in Figure 3. For example the modified features of the actor areas in feature maps 0 and 43, after modulation, especially for the female ice skater, which is blended into the background on the regular RGB stream. On the 28-th feature map, a feature response is even hard to see on both actors before modulation. Feature maps 10 and 127 show the change in $x$-direction features and $y$-direction features. The flow condition pushes the model to focus on moving actors.

**Training loss.** In order to demonstrate the generalization and flexibility of the proposed method, we embed the motion condition layer and the motion modulation layer in a

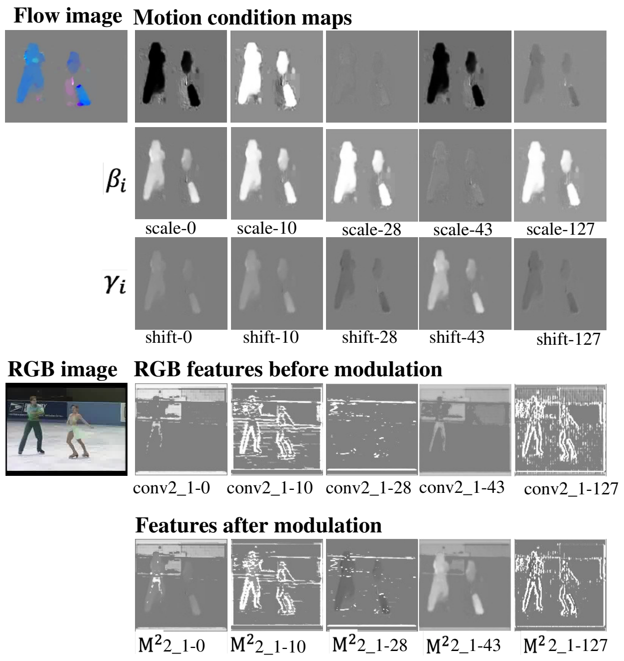


Figure 3: **Feature maps.** Visualization of the motion condition maps, scale maps, shift maps, RGB features without modulation and features with modulation. The modulated features focus more on moving actors.

single-frame appearance stream and a multi-frame appearance stream. The basic loss function is derived from the one for object detection [27, 31]. Defining ${x^p}_{ij} = \{1, 0\}$ as an indicator for matching the $i$-th default box to the $j$-th ground truth box of action category $p$. The overall loss function contains the localization ($loc$) loss and the confidence ($conf$) loss:

$$\mathcal{L}(x, c, l, g) = \frac{1}{N}(\mathcal{L}_{loc}(x, l, g) + \mathcal{L}_{conf}(x, c)) \tag{7}$$

with $N$ representing the number of matched default boxes. $c$ represents multiple classes confidences. $l$ and $g$ are the predicted box and the ground truth box.

The confidence loss applies the softmax loss as below:

$$\begin{aligned} \mathcal{L}_{conf}(x, c) &= -\sum_{i \in Pos}^{N} x_{ij}^{p} log(\hat{c}_i^p) - \sum_{i \in Neg} log(\hat{c}_i^0) \\ \hat{c}_i^p &= \frac{exp(c_i^p)}{\sum_p exp(c_i^p)} \end{aligned} \tag{8}$$

The localization loss applies a smooth L1 loss [12] between the predicted box and the ground truth box. The network regresses to offsets for the center $(cx, cy)$ of the default box($d$) and for its width ($w$) and height ($h$).

$$\begin{aligned} \mathcal{L}_{loc}(x, l, g) &= \sum_{i \in Pos}^{N} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^{k} smooth_{\mathrm{L1}}(l_i^m - \hat{g}_j^m) \\ \hat{g}_j^{cx} &= (g_j^{cx} - d_i^{cx})/d_i^w \qquad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h \\ \hat{g}_j^w &= log(\frac{g_j^w}{d_i^w}) \qquad \hat{g}_j^h = log(\frac{g_j^h}{d_i^h}) \end{aligned} \tag{9}$$

For the multi-frame appearance stream, we follow Kalogeiton *et al.* [21] to train the network.

**Two-in-one two-stream.** Our method emphasizes to utilize RGB and optical flow information in one stream. Furthermore, it is possible to follow the standard practice of two-stream action detection. We train a two-in-one detector conditioned on flow images, and a separate flow detector which only takes as input the flow images. For a single-frame two-in-one two-stream, we use average fusion method to merge the results from each stream following [37]. And for multi-frame two-stream, the late fusion [9] is a better choice [21].

**Linking.** Once the frame-level detections or tubelet detections are achieved, we link them to build action tubes. We adopt the linking method described in [37] for frame-level detections and the method in [21] for tubelet detections.

Code is available at https://github.com/jiaozizhao/Two-in-One-ActionDetection.

## 4. Experiments

### 4.1. Datasets, Metrics & Implementation

**Datasets.** We perform experiments on three action detection datasets. *UCF101-24* [39] is a subset of *UCF101*. It contains 24 sport classes in 3207 untrimmed videos. Each video contains a single action category. Multiple action instances with the same class, but different spatial and temporal boundaries may occur. We use the revised annotations for *UCF101-24* from [37]. *UCF-Sports* [32] contains 10 sport classes in 150 trimmed videos. We follow [24] to divide the training and test splits. *J-HMDB* [20] contains 21 action categories in 928 trimmed videos. We report the average results on three splits.

**Metrics.** Following [34, 38, 48], we utilize video mean Average Precision ($mAP$) to evaluate action detection accuracy. We calculate an average of per-frame Intersection-over-Union (IoU) across time between tubes. A detection is correct if it's IoU with the ground truth tube is greater than a threshold and its action label is correctly assigned. We compute the average precision for each class and report the average over all classes.

**Implementation.** We adopt a real-time single shot multibox detector (SSD) network [27] as the backbone. We

| | Action Detection | | | Action Classification | | |
|---|---|---|---|---|---|---|
| Method | *mAP* | *Efficiency* | | *Top1 Accuracy* | *Efficiency* | |
| | % | sec/frame | # param. (M) | % | sec/frame | # param. (M) |
| flow-stream | 11.60 | 0.04 | 26.82 | 81.65 | 1.10 | 58.35 |
| RGB-stream | 18.49 | 0.04 | 26.82 | 84.99 | 1.10 | 58.35 |
| two-stream | 19.79 | 0.09 | 53.64 | 91.14 | 2.10 | 116.70 |
| two-in-one stream | 20.15 | 0.04 | 26.93 | 86.94 | 1.15 | 58.48 |
| two-in-one two stream | **22.02** | 0.09 | 53.75 | **92.00** | 2.13 | 116.83 |

Table 1: **Two-in-one *vs.* baselines** for action detection on *UCF101-24* and action classification on *UCF101*. Two-in-one with motion modulation works well for both action detection and action classification.

insert the developed motion layers into two state-of-the-art appearance SSD networks, one based on single frame [37] and the other based on multiple frames [21]. We use VGG-16 pre-trained weights on ImageNet as model initialization. The input size is 300x300 for both of them. We follow [21] to use 6 continuous frames as input to the multi-frame SSD. The initial learning rate is set to 0.001 for the single-frame network and 0.0001 for the multi-frame network on all the three datasets and changed by applying step decay strategy. We trained a flow-stream, an RGB-stream and our two-in-one stream for 13.2, 13.2 and 15.5 hours, respectively.

Alternatively, we considered to use appearance information to modulate flow stream. However, it does not work well. It appears difficult to modulate features from flow images which are sparse, using RGB images which are more dense.

## 4.2. Ablation Study

All the ablation studies are performed on *UCF101-24*. We only report $mAP$ at the most challenging high $IoU$ thresholds 0.5:0.95 (with step 0.05). Initially, in order to maintain the spatial pixel-wise motion vectors, we apply 1x1 convolution kernels to all layers in the motion condition layer and the motion modulation layer. We use layer parameter stride to control the size of $\beta$ and $\gamma$. Then the motion modulation layer is applied to conv1 of SSD. Flow images are generated using the method in [2], which we refer to as BroxFlow.

**Two-in-one *vs.* baselines.** We compare the two-in-one stream to its corresponding RGB-stream, flow-stream and two-stream in Table 1. Runtime and # param. are also reported for comparing the efficiency. Our single two-in-one stream exceeds a single RGB-stream by 1.5%. Notably, two-in-one even outperforms the corresponding two-stream with only half the computation cost and # param..

We also consider action classification, on *UCF101*. We follow [46], with ResNet152 as backbone. The *Top 1* accuracy and efficiency shown in Table 1 illustrate our strategy also works for action classification and generalizes beyond SSD with VGG16. For training, our two-in-one stream converges at the 100-th epoch, but the RGB- and flow-stream converge at 200-th and 300-th epoch, respectively. Our motion modulation strategy works better for the detection task, which needs localization representations that are translation-variant, compared to the classification task which favors translation invariance.

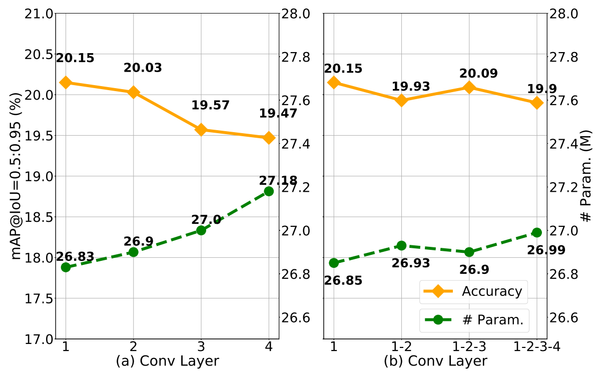


Figure 4: **Where to add the modulation layer?** Accuracy on *UCF101-24* and # param. with varying: (a) single modulated layer, and (b) multiple modulated layers. A single modulation layer at conv1 gets the best result.

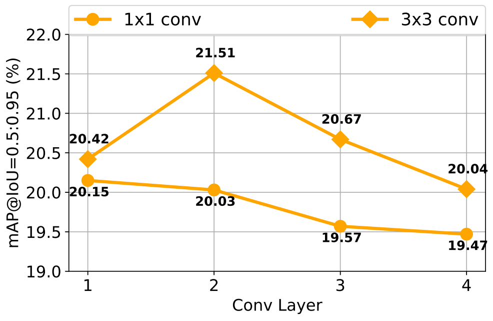


Figure 5: **How to design the condition layer?** Comparing accuracy on *UCF101-24* when applying 1x1 conv or 3x3 conv to the last layer of the motion condition layer. The 3x3 conv performs better.

**Where to add the modulation layer?** The motion condition layer is leveraged to generate low-level motion features as flow images are more sparse. We add the motion

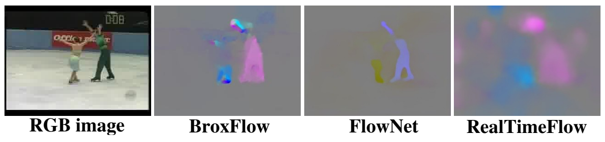


Figure 6: **What flow?** Examples of flow images generated by different flow methods.

| | BroxFlow | FlowNet | RealTimeFlow |
|---|---|---|---|
| flow-stream | 11.60 | 7.13 | 3.58 |
| RGB-stream | 18.49 | 18.49 | 18.49 |
| two-stream | 19.79 | 19.75 | 18.53 |
| two-in-one stream | **21.51** | 19.97 | 19.16 |

Table 2: **What flow?** No matter what flow images are applied on *UCF101-24*, our two-in-one stream outperforms the corresponding flow-, RGB- and two-stream. We obtain the best result with BroxFlow.

modulation layer to the bottom convolutional layers with low-level RGB features. We conduct two experiments on which layer to add the modulation. We compare the accuracy and # param. after applying a modulation layer to conv1, conv2, conv3 and conv4 in Figure 4 (a). Accuracy decreases and # param. increases slightly for deeper layers. Next we add the modulation layers to multiple convolutional layers simultaneously in Figure 4 (b). Applying multiple modulation layers does not change the results much. Thus, we prefer to use a single modulation layer. Note that accuracy drops for deeper layers as we use 1x1 convolution kernels to process flow images, leading to smaller receptive field for deeper layers.

**How to design the condition layer?** To further improve the method, we consider whether the 1x1 convolution kernel for the motion condition layer is the best choice. Besides keeping the spatial pixel-wise motion, it may need to consider some context of motion to better fit the RGB features. We adopt the 3x3 convolution kernels to the last layer of the condition network. Figure 5 demonstrates that considering motion context boosts the accuracy for all layers. As a bigger receptive field is used, the conv2 model achieves the best results, about 1.5% improvement compared to 1x1 convolution kernels. The run time hardly increases for deeper layers, and is still 0.04 sec per frame. The # param. are 26.85, 26.92, 27.01 and 27.19 M respectively of conv1, conv2, conv3 and conv4. Considering the trade-off between the results and parameters, we believe conv2 provides the best accuracy/efficiency trade-off.

**What flow?** As we leverage flow information as prior conditions, we wonder how the model is influenced by flow images. Here we adopt flow images generated by three different methods (seen in Figure 6) and evaluate how

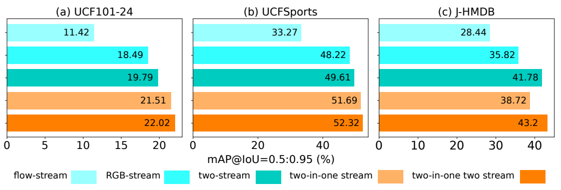
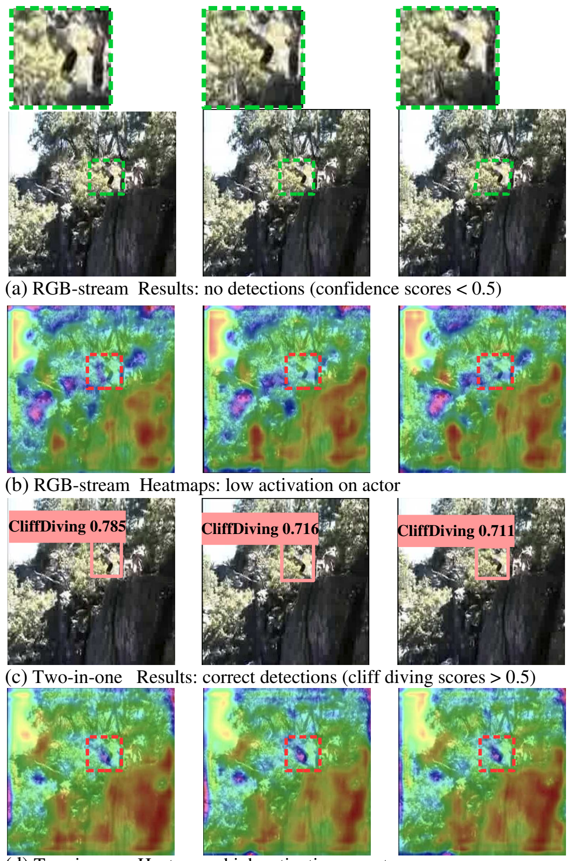


Figure 7: **Generalization ability.** Accuracy comparison on: (a) *UCF101-24*, (b) *UCFSports*, (c) *J-HMDB*, with different methods. Two-in-one stream even outperforms two-stream on *UCF101-24* and *UCFSports*. Two-in-one stream fused with a flow-stream obtains the best accuracy on all three datasets.

our strategies work. We use BroxFlow [2] (accurate flow method), Flownet [6] (deep network method) and a real-time but less accurate optical flow method [23] (RealTimeFlow). From Table 2, it is concluded that no matter which kind of flow images are applied, our two-in-one stream outperforms RGB-streams and corresponding two-streams. We also note that the more accurate the flow images, the more improvement the two-in-one stream obtains. Even when using the somewhat noisy RealTimeFlow images, the two-in-one stream still improves the RGB-stream. However, a two-stream based on RealTimeFlow obtains almost the same accuracy as the RGB-stream, which illustrates that two-stream depends on the the quality of flow images. Our two-in-one stream is more robust to the quality of flow images. Moreover, we report the flow computation in seconds/frame for the three kinds of flow methods: BroxFlow (0.098), FlowNet (0.183) and RealTimeFlow (0.014). RealTimeFlow only needs 0.014 seconds to generate one flow image, at the expense of a slightly lower *mAP*.

**Generalization ability.** To stress the generalization ability of our proposal, we compare the results on three different datasets. Following the conclusions of our ablation so far, we use the BroxFlow image for generating condition and apply a 3x3 kernel to the last layer of the motion condition layer. The motion modulation layer is only leveraged for the conv2 layer of the appearance stream. We report results in Figure 7.

Obviously, the proposed two-in-one stream performs better than other one-stream networks. It is noteworthy that our two-in-one stream even outperforms traditional two-stream networks on *UCF101-24* and *UCFSport* by 2% with only half the parameters of a two-stream network. On *J-HMDB*, two-in-one is 3% higher than RGB-stream but 3% lower than two-stream. We look into *J-HMDB* and find that most videos in the dataset have neighbouring repeated frames. For fair comparison, we just download the BroxFlow images used in [21,37]. However, the provided BroxFlow image between the two repeated RGB frames is not 0, as it should be, but similar to the last flow frame. The issue affects our two-in-one stream due to the fact that we need



(a) RGB-stream Results: no detections (confidence scores < 0.5)

(b) RGB-stream Heatmaps: low activation on actor

(c) Two-in-one Results: correct detections (cliff diving scores > 0.5)

(d) Two-in-one Heatmaps: high activation on actor

Figure 8: **Visualization of detection and heatmaps** on conv4 layers from RGB-stream network in (a) (b) and two-in-one stream network in (c) (d). We add the green dashed boxes to indicate the action. The two-in-one stream has higher activation on actions, resulting in correct detection.

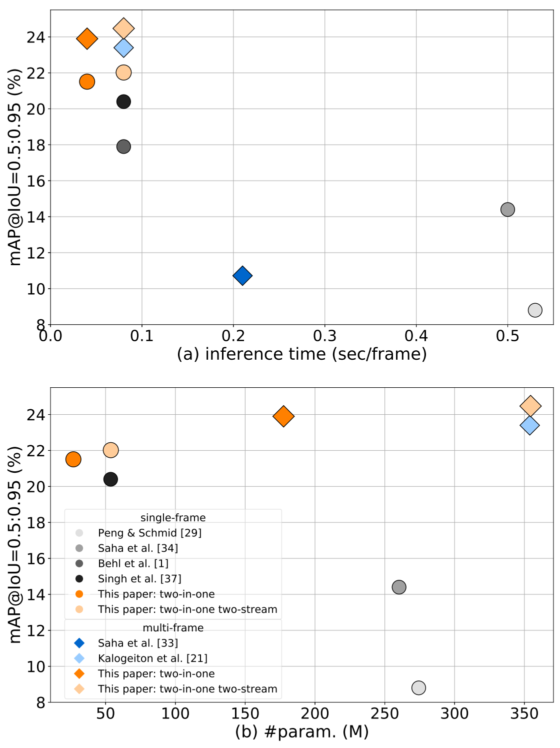


Figure 9: **Efficiency comparison to the state-of-the-art.** Accuracy *vs.* (a) inference time (second per frame) and (b) # param. (M) on *UCF101-24*. Our two-in-one stream best balances accuracy and efficiency.

correct flow image as the condition of the corresponding RGB frame. We expect that two-in-one will present better results on *J-HMDB* after correcting the flow images. As expected, adding a separate flow-stream to our two-in-one stream gives the best accuracy on all datasets.

## 4.3. Qualitative Analysis

The motion condition layer and the motion modulation layer are beneficial to generate better video representations for spatio-temporal action detection. But how do the layers make a difference to the appearance network? To understand this behavior, we visualize in Figure 8, the detection results of an RGB-stream network and a two-in-one network. Also, we visualize the gradient-weighted class activation heatmaps [53] for better understanding how the motion conditions influence the behavior of the appearance network. We choose a challenging case of cliff diving here. The image resolution is low and the actor is quite tiny. The cluttered background obviously increases the difficulty to detect actions. We manually overlay green dashed boxes to indicate the locations of the actor and zoom in to highlight where the action is happening. The second row shows that the RGB-stream fails to detect any actions. From the corresponding heatmaps, it is apparent that the appearance network pays more attention to the background than to the actions. There are only weak responses on the action positions. We manually overlay red dashed boxes to highlight the position of the actor on the heatmaps. From the heatmaps for the two-in-one network in the last row, we clearly see it is capable to balance the activation on actions and background. The responses on action positions are strengthened. As expected, the two-in-one stream performs better than the RGB-stream. It outputs correct detections for cliff diving on all the frames (forth row ).

## 4.4. Comparison to the State-of-the-art

**Accuracy.** For fair comparisons, we just use the original images as in all the state-of-the-arts, without camera motion removal. We compare the $mAP$ at variable $IoU$ thresholds in Table 3. Considering the most challenging high $IoU$ thresholds 0.5:0.95, we observe that for the single-frame setting, our two-in-one stream achieves even better results than existing two-stream methods on *UCF101-24* and *UCF-*

| | *UCF101-24* | | | | *UCFSports* | | | *J-HMDB* | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0.20 | 0.50 | 0.75 | 0.50:0.95 | 0.50 | 0.75 | 0.50:0.95 | 0.50 | 0.75 | 0.50:0.95 |
| **Single-frame** | | | | | | | | | | |
| Peng & Schmid [29] | 71.80 | 35.90 | 1.60 | 8.80 | **94.80** | 47.30 | 51.00 | 70.60 | *48.20* | *42.20* |
| Saha *et al.* [34] | 66.70 | 35.90 | 7.90 | 14.40 | – | – | – | *71.50* | 43.30 | 40.00 |
| Behl *et al.* [1] | 71.53 | 40.07 | 13.91 | 17.90 | – | – | – | – | – | – |
| Singh *et al.* [37] | 73.50 | 46.30 | 15.00 | 20.40 | – | – | – | **72.00** | 44.50 | 41.60 |
| *This paper: Two-in-one* | *75.13* | *47.47* | *17.21* | *21.51* | 87.46 | *57.81* | *51.69* | 60.99 | 47.23 | 38.72 |
| *This paper: Two-in-one two stream* | **77.49** | **49.54** | **17.62** | **22.02** | *87.81* | **62.67** | **52.32** | 70.00 | **52.00** | **43.20** |
| **Multi-frame** | | | | | | | | | | |
| Saha *et al.* [33] | 63.06 | 33.06 | 0.52 | 10.72 | – | – | – | 57.31 | – | – |
| Kalogeiton *et al.* [21] | 76.50 | 49.20 | 19.70 | 23.40 | 92.70 | 78.40 | 58.80 | *73.70* | *52.10* | *44.80* |
| Singh *et al.* [36] | **79.00** | **50.90** | 20.10 | *23.90* | – | – | – | – | – | – |
| *This paper: Two-in-one* | 75.48 | 48.31 | *22.12* | *23.90* | *92.74* | *83.64* | *59.60* | 57.96 | 42.78 | 34.56 |
| *This paper: Two-in-one two stream* | *78.48* | *50.30* | **22.18** | **24.47** | **96.52** | **90.41** | **63.59** | **74.74** | **53.28** | **45.01** |

Table 3: **Accuracy comparison to the state-of-the-art.** Bold means top accuracy and italic means second top accuracy. For the high overlap setting of $mAP@IoU$=0.5:0.95, our two-in-one stream works well in both a single-frame and multiple-frame network for all three datasets. When we add an additional flow-stream to obtain a two-in-one two stream we further improve accuracy.

*Sports*. For instance, two-in-one stream outperforms Singh *et al.* [37] with the same SSD detector by more than 1% and Peng and Schmid [29] with a Faster-RCNN detector by an absolute 12% on *UCF101-24*. As analyzed previously, two-in-one stream performs modest on *J-HMDB* because of the data issue of the provided BroxFlow images. When we combine two-in-one into a regular two-stream network by fusing with a flow-stream, it produces good results on all three datasets. Compared to two-in-one stream, it gets about 5% improvement on *J-HMDB*. Moreover, when feeding our two-in-one network variants with multiple frames, as suggested by Kalogeiton *et al.* [21], our two-in-one stream outperforms the two-stream [21] a little on *UCF101-24* and *UCFSports* with only half computation and the number of parameters. Our two-in-one stream fused with a flow stream further boosts the results, outperforming the very recent work of Singh *et al.* [36].

**Efficiency.** Besides good detection accuracy, our method has the advantage of a reduced inference time and less # param.. Here we compare our methods from the efficiency aspect to the state-of-the-art on *UCF101-24*. We test our models on one NVIDIA GTX 1080 GPU. The trade-off between accuracy and inference time, as well as parameters are visualized in Figure 9. Among the single-frame methods, our two-in-one stream has the fastest run time with 0.04s per frame, two times faster than [1] and [37] and much faster than [34] and [29] (about 0.5s per frame). Moreover, the # param. of our two-in-one stream is smallest, about 26.93 M. While our two-in-one accuracy is even better than the two-stream methods by [1, 29, 34, 37]. Combining our two-in-one stream with a standard flow-stream gains an accuracy improvement at the expense of more computation and parameters. Our two-in-one alternative even outperforms [21] a little in accuracy with only half the parameters. The two-in-one two stream further improves the result with almost similar inference time, but slightly more parameters. We conclude that two-in-one stream networks provide a good accuracy/efficiency trade-off.

## 5. Conclusion

We propose an effective and efficient two-in-one stream network for spatio-temporal action detection. It takes flow images as prior motion condition when training an RGB-stream network. The network's motion condition layer and motion modulation layer address two issues in action detection: frame-level RGB images lack motion information and (static) background-context may dominant the learned representation. Our two-in-one stream achieves state-of-the-art accuracy at high $IoU$ thresholds, using only half of the parameters and computation of two-stream alternatives. Besides motion, we believe that other information such as depth-maps or infrared images may help locate the actors, and can be exploited as additional prior conditions for training two-in-one streams.

**Acknowledgments** Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number D17PC00343. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

## References

[1] Harkirat S Behl, Michael Sapienza, Gurkirt Singh, Suman Saha, Fabio Cuzzolin, and Philip HS Torr. Incremental tube construction for human action detection. In *BMVC*, 2018. 1, 8

[2] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004. 5, 6

[3] Liangliang Cao, Zicheng Liu, and Thomas S Huang. Cross-dataset action detection. In *CVPR*, 2010. 2

[4] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006. 2

[5] Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In *NIPS*, 2017. 1

[6] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 6

[7] Kevin Duarte, Yogesh Rawat, and Mubarak Shah. Videocapsulenet: A simplified network for action detection. In *NeurIPS*, 2018. 2

[8] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In *ICLR*, 2017. 1

[9] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016. 2, 4

[10] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network. In *BMVC*, 2017. 1

[11] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. A better baseline for ava. *arXiv preprint arXiv:1807.10066*, 2018. 2

[12] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. 4

[13] Georgia Gkioxari and Jitendra Malik. Finding action tubes. In *CVPR*, 2015. 1, 2

[14] Chunhui Gu, Chen Sun, Sudheendra Vijayanarasimhan, Caroline Pantofaru, David A Ross, George Toderici, Yeqing Li, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018. 2

[15] Jiawei He, Zhiwei Deng, Mostafa S Ibrahim, and Greg Mori. Generic tubelet proposals for action localization. In *WACV*, 2018. 2

[16] Rui Hou, Chen Chen, and Mubarak Shah. Tube convolutional neural network (t-cnn) for action detection in videos. In *ICCV*, 2017. 2

[17] Xun Huang and Serge J Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 1

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 1

[19] Mihir Jain, Jan C. van Gemert, Hervé Jégou, Patrick Bouthemy, and Cees G. M. Snoek. Tubelets: Unsupervised action proposals from spatiotemporal super-voxels. *IJCV*, 124(3):287–311, 2017. 2

[20] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *ICCV*, 2013. 4

[21] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *ICCV*, 2017. 1, 2, 4, 5, 6, 8

[22] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008. 2

[23] Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. Fast optical flow using dense inverse search. In *ECCV*, 2016. 6

[24] Tian Lan, Yang Wang, and Greg Mori. Discriminative figure-centric models for joint action localization and recognition. In *ICCV*, 2011. 2, 4

[25] Dong Li, Zhaofan Qiu, Qi Dai, Ting Yao, and Tao Mei. Recurrent tubelet proposal and recognition networks for action detection. In *ECCV*, 2018. 2

[26] Zhenyang Li, Kirill Gavrilyuk, Efstratios Gavves, Mihir Jain, and Cees G. M. Snoek. Videolstm convolves, attends and flows for action recognition. *CVIU*, 166:41–50, 2018. 2

[27] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 2, 4

[28] Dan Oneata, Jakob Verbeek, and Cordelia Schmid. Efficient action localization with approximately normalized fisher vectors. In *CVPR*, 2014. 2

[29] Xiaojiang Peng and Cordelia Schmid. Multi-region two-stream r-cnn for action detection. In *ECCV*, 2016. 1, 2, 8

[30] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018. 1

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 4

[32] Mikel D Rodriguez, Javed Ahmed, and Mubarak Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008. 4

[33] Suman Saha, Gurkirt Singh, and Fabio Cuzzolin. Amtnet: Action-micro-tube regression by end-to-end trainable deep architecture. In *ICCV*, 2017. 8

[34] Suman Saha, Gurkirt Singh, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Deep learning for detecting multiple space-time action tubes in videos. In *BMVC*, 2016. 2, 4, 8

[35] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 1, 2

[36] Gurkirt Singh, Suman Saha, and Fabio Cuzzolin. Tramnet-transition matrix network for efficient action tube proposals. In *ACCV*, 2018. 1, 2, 8

[37] Gurkirt Singh, Suman Saha, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *ICCV*, 2017. 1, 2, 4, 5, 6, 8

[38] Khurram Soomro, Haroon Idrees, and Mubarak Shah. Predicting the where and what of actors and actions through online action localization. In *CVPR*, 2016. 4

[39] 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. 4

[40] Rastislav Šrámek, Broňa Brejová, and Tomáš Vinař. On-line viterbi algorithm and its relationship to random walks. *arXiv preprint arXiv:0704.0062*, 2007. 2

[41] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *ECCV*, 2018. 2

[42] Yicong Tian, Rahul Sukthankar, and Mubarak Shah. Spatiotemporal deformable part models for action detection. In *CVPR*, 2013. 2

[43] Du Tran and Junsong Yuan. Max-margin structured output regression for spatio-temporal action localization. In *NIPS*, 2012. 2

[44] Du Tran, Junsong Yuan, and David Forsyth. Video event detection: From subvolume localization to spatio-temporal path search. *TPAMI*, 2014. 2

[45] Jan C Van Gemert, Mihir Jain, Ella Gati, and Cees G. M. Snoek. Apt: Action localization proposals from dense trajectories. In *BMVC*, 2015. 2

[46] Limin Wang, Yuanjun Xiong, Zhe Wang, and Yu Qiao. Towards good practices for very deep two-stream convnets. *arXiv preprint arXiv:1507.02159*, 2015. 5

[47] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *CVPR*, 2018. 1

[48] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *ICCV*, 2015. 1, 2, 4

[49] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018. 2

[50] Zhenheng Yang, Jiyang Gao, and Ram Nevatia. Spatio-temporal action detection with cascade proposal and location anticipation. In *BMVC*, 2017. 2

[51] Gang Yu and Junsong Yuan. Fast action proposals for human action detection and search. In *CVPR*, 2015. 2

[52] Junsong Yuan, Zicheng Liu, and Ying Wu. Discriminative video pattern search for efficient action detection. *TPAMI*, 33(9):1728–1743, 2011. 2

[53] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 7