Accurate Single-Stream Action Detection in Real-Time

Yu Liu, Fan Yang, Dominique Ginhac

Accurate Single-Stream Action Detection in Real-Time

Yu Liu, Fan Yang, Dominique Ginhac

Accurate Single-Stream Action Detection in Real-Time

Yu Liu
Laboratoire ImViA, Univ. Bourgogne Franche-Comté
Dijon, France
yu_liu@etu.u-bourgogne.fr

Fan Yang
Laboratoire ImViA, Univ. Bourgogne Franche-Comté
Dijon, France
fanyang@u-bourgogne.fr

Dominique Ginhac
Laboratoire ImViA, Univ. Bourgogne Franche-Comté
Dijon, France
dominique.ginhac@ubfc.fr

ABSTRACT

Analyzing videos of human actions involves understanding the spatial and temporal context of the scenes. State-of-the-art action detection approaches have demonstrated impressive results using Convolutional Neural Networks (CNNs) within a two-stream framework. However, most of them operate in a non-real-time, offline fashion, thus are not well-equipped in many emerging real-world scenarios such as autonomous driving and public surveillance. In addition, they are computationally demanding to be deployed on devices with limited power resources (e.g., embedded systems). To address the above challenges, we propose an efficient single-stream action detection framework by exploiting temporal coherence between successive video frames. This allows CNN appearance features to be cheaply propagated by motions rather than being extracted from every frame. Furthermore, we utilize an implicit motion representation to amplify appearance features. Our method based on motion-guided and motion-aware appearance features is evaluated on the UCF-101-24 dataset. Experiments indicate that the proposed method can achieve real-time action detection up to 32 fps with a comparable accuracy as the two-stream approach.

CCS CONCEPTS

• **Computing methodologies** → *Activity recognition and understanding*.

KEYWORDS

action detection, video analytics, convolutional neural network, embedded system

ACM Reference Format:
Yu Liu, Fan Yang, and Dominique Ginhac. 2019. Accurate Single-Stream Action Detection in Real-Time. In *Trento '19: International Conference on Distributed Smart Cameras, Sep. 09–11, 2019, Trento, Italy*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3349801.3349821

1 INTRODUCTION

Human action detection is a key element to video understanding. It has been an active area of research driven by a host of applications such as assistive robots, autonomous driving, and unmanned surveillance, etc. As it requires simultaneously localizing actors and classifying each of their action, action detection is a challenging problem where only a few of the current approaches permit real-time performance. Nevertheless, in many real-life scenarios not only robust recognition of actions is required, but also on-site and real-time operation. In addition, for ease of mobility and / or economical design, detection systems need to be integrated into embedded devices in applications such as mobile robotics and distributed surveillance. Factoring in limited power and computational resources further raises difficulty of the detection problem.

With the recently rising deep Convolutional Neural Network (CNN), vision-based tasks such as image classification and object detection have progressed significantly. Following the success of CNN, researchers also adopt CNN object detectors for the task of action detection. Methods relying on these detectors typically extract appearance features individually for each video frame to achieve frame-based action detection [21][17], or stack multiple features over a short period of time to predict action tubes [12][23]. Many efforts modeling actions' motion cues have also been made, such as exploiting a two-stream architecture to complement appearance and motion features.

When limited power and computational resources are considered, the above approaches are sub-optimal in two folds. Firstly, as consecutive video frames exhibit high content similarity, extracting features from all of them is redundant and costly. Moreover, the increased system complexity associated with adopting a two-stream architecture is not proportionally reflected in the detection accuracy. For instance, Singh et al.[24] demonstrate a detection improvement by nearly 15% when incorporating an accurate optical flow stream to aid the appearance stream, however at the cost of real-time performance and doubling CNN parameters. In contrast, we believe that exploiting the temporal continuity of video frames enables efficient action video processing. This concept in fact has been applied in a number of vision-based tasks but action detection. In addition, we hypothesize that a simple implicit motion representation can be used to facilitate detection in a more direct manner than the two-stream approach.

In this paper, we investigate efficient and accurate action detection with real-time performance, which is potentially well-suited for embedded systems. We make the following contributions:

- Integrate a feature propagation framework originally designed for video object detection [27]. Guiding appearance features by motion permits real-time action detection.
- Employ motion-aware features by amplifying appearance features with implicit motion information. This serves to replace the heavy two-stream architecture.

We evaluate the proposed framework on the UCF-101-24 dataset. Experimental results show that our motion-guided and motion-aware features enable real-time detection without compromising accuracy.

2 RELATED WORK

Recently, deep CNNs' remarkable results have made them the primary choice for computer vision tasks such as image classification and object detection. They also demonstrate leading performance in action recognition, an area of research closely related to action detection. Inspired by these advances, many efforts are made to address action detection by incorporating CNN image object detectors within an action recognition framework. In this section, areas of research relevant to our action detection problem are reviewed.

Image object detection. Modern CNN object detectors can be grouped into two families. The first one consists of a two-stage framework popularized by R-CNN [8]. This branch of method first extracts potential object regions from images, on which it then performs object classification and bounding box regression. State-of-the-art two-stage detectors (such as Faster R-CNN [19] and R-FCN [2]) incorporate the fast and simple Region Proposal Network (RPN) in the first stage, pushing towards near real-time performance. Alternative to the two-stage method, object detectors such as YOLO [18] and SSD [15] directly predict both the regressed bounding boxes and their associated classes in a single pass without the intermediate region proposal. In exchange for minor drops in accuracy, these single-stage methods can achieve real-time detection.

Video object detection. Following the introduction of the VID challenge [20], a number of researchers explores improving single-image detection by exploiting detection results from multiple video frames. For instance in Seq-NMS, Han et al. [9] associate high-confidence bounding boxes from consecutive frames by their spatial overlap, followed by rescoring of each box to boost weaker detection. Kang et al. [13] map detection results to adjacent frames by motion and apply tracking algorithms to enforce long-term temporal consistency. More recently, there have been studies on CNN architectures which simultaneously handle frame-based detection and tracking regression across frames [6].

The above approaches typically do not concern efficient processing. On the other hand, other works specifically target efficient detection by exploiting temporal redundancy within video frames. For example, Zhu et al. [27] propagate deep feature maps from key frames to their successive frames via flow fields. This accelerates video object detection as features can be obtained by feeding only a sparse set of key frames to the time-consuming deep feature extractor. In a similar spirit, Liu et al. [14] propagate frame-level information across frames using their proposed recurrent-convolutional architecture. Feature warping has also been applied in other video-based tasks other than object detection [7].

Action recognition is typically approached as a classification problem for trimmed videos. Among the many techniques used to solve action recognition, the two-stream network [22] introduced by Simonyan et al. demonstrates state-of-art performance. Under this framework, two CNNs, one for the spatial stream (e.g., RGB images) and the other one for the temporal stream (typically optical flows), run separately followed by a fusion step. A number of ways to fuse the appearance and motion features for a complementary action representation have been investigated [16][5]. Various forms of motion cues other than optical flows have also been looked into, such as feature difference [26] and motion history image [1].

Action detection further addresses simultaneous action localization and classification. A number of efforts have been made to extend from object detectors. The extensions mainly include adopting the two-stream framework to obtain frame-level detection from each stream, followed by a fusion technique to exploit information from both streams. For instance of a late fusion, Saha et al. [21] boost the overall detection confidence by how much the detection results from both streams agree with each other. Similarly, Peng et al.[17] apply all the region proposals from both streams to recall as many potential actions as possible. Both of the above works involve the use of two-stage object detectors and computationally expensive optical flow, which prohibit real-time deployment.

Targeting real-world scenarios, Singh et al. [24] propose an online and real-time action detection framework by combining the two-stream networks, SSD object detector, fast optical flow estimator and an online linking algorithm. To further leverage video temporal information, methods based on detection at the clip-level (rather than frame-level) are proposed. For example, Kalogeiton et al. [12] predict classified and regressed action tubes on stacked features extracted from a short sequence of input frames. Although these methods have achieved promising results, temporal continuity of videos is not explicitly exploited as detection or feature extraction is still performed on each frame independently. They also employ a two-stream architecture which increases the complexity of their detection pipeline.

3 METHOD DESCRIPTION

In this section, we present our single-stream action detection framework with motion-guided and motion-aware features. Figure 1 provides a summary of the method. We follow the deep feature flow framework [27] which treats a sparse set of RGB images as key frames, and the rest as non-key frames. When a key frame is inputted, its CNN feature is extracted and directly used for detecting actions. For a non-key frame, a flow field between this frame and its preceding key frame is first estimated and fine-tuned. Using the transformed flow field, we propagate (or spatially warp) the appearance feature of the preceding key frame. Finally the warped appearance feature is amplified with motion information and used to predict regressed boxes and their associated action classes. The entire network is end-to-end trainable. Each building block is explained in detail in the following sections.

3.1 Motion-guided appearance feature

In videos, image content varies slowly over consecutive frames. This is more so reflected in CNN deep feature maps which encode high-level semantics. Hence, applying the bottom-up feature extraction for every video frame is redundant and costly. To exploit the inter-frame redundancy for efficient video processing, we adopt the work of deep feature flow, guiding feature maps from key frames to their subsequent frames by relative motions.

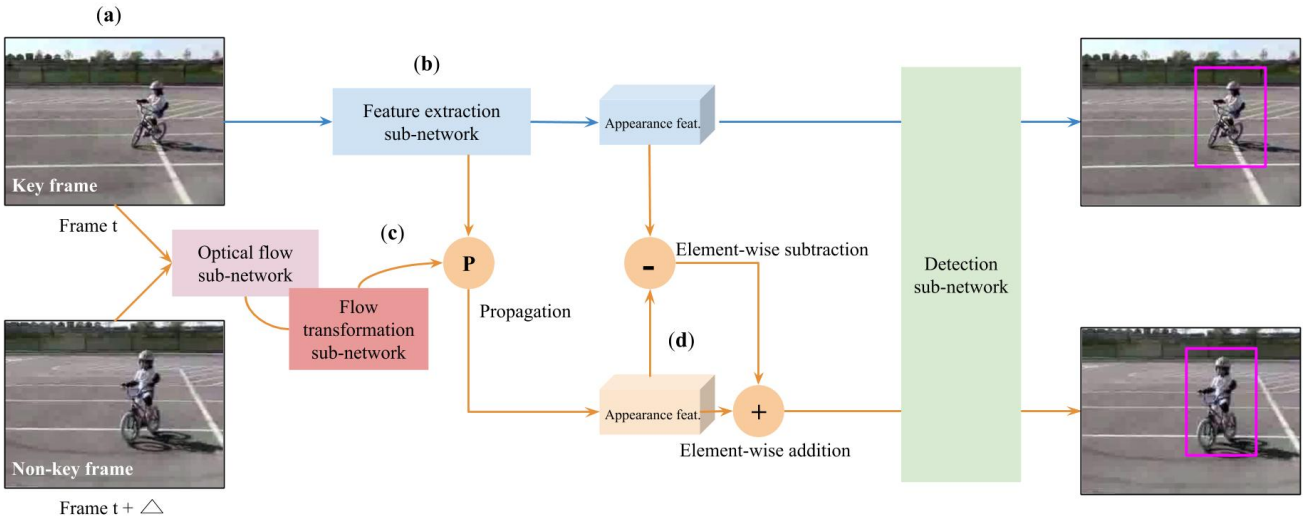

Figure 1: Illustration of our motion-guided action detection framework with motion-aware appearance features. (a) Two types of input frames. (b) For a key frame, appearance feature is obtained from the CNN feature extraction sub-network. (c) For a non-key frame, the flow sub-network estimates a flow field between the non-key frame and its preceding key frame. The resulted flow is first transformed and then used to propagate appearance feature. (d) An implicit motion cue (difference in features) amplifies the appearance feature. Blue and orange arrows are carried out at a key and non-key frame respectively.

Figure 2 illustrates how feature propagation functions in our framework. During inference the heavy and expensive feature extraction sub-network is only applied on a sparse set of key frames. The feature maps of successive non-key frames are then obtained by propagating the features from their preceding key frames via two-channel flow fields, followed by the detection sub-network. By applying feature propagation on a dense set of non-key frames, computation is greatly reduced as propagation is a fast and inexpensive operation compared to CNN deep feature extraction. Specifically, propagation is carried out by a spatial warping function based on bilinear sampling [11] for all locations and channels in the feature maps, where a flow field is resized to have the same spatial resolution as a feature map. More details about the warping operation can be referred from [27].

A flow field is estimated between a non-key frame and its preceding key frame. Previous works for solving action detection typically compute flow fields independently from their detection model (e.g., using third-party optical flow algorithms). This inherently incurs high consumption of time or requires preparation of flow results in advance, prohibiting real-time and online operation. In contrast, our framework integrates a fast flow estimation sub-network with low complexity. This gets rid of the need for having an external flow estimator. In additional, it allows joint training with all other sub-networks specific to the task of action detection.

Finally, because the fast flow sub-network might be less accurate, we apply a 4-layer convolutional sub-network to fine-tune its flow field. The layer design follows the work of [7], which is loosely inspired by residual blocks. The input to this transformation sub-network is the raw flow field estimated by the flow sub-network; the outputted flow after transformation is used to propagate appearance feature. All sub-networks (feature extraction, optical flow, flow transformation and action detection) are jointly trained to minimize loss incurred by detection results; no groundtruth flow is required.

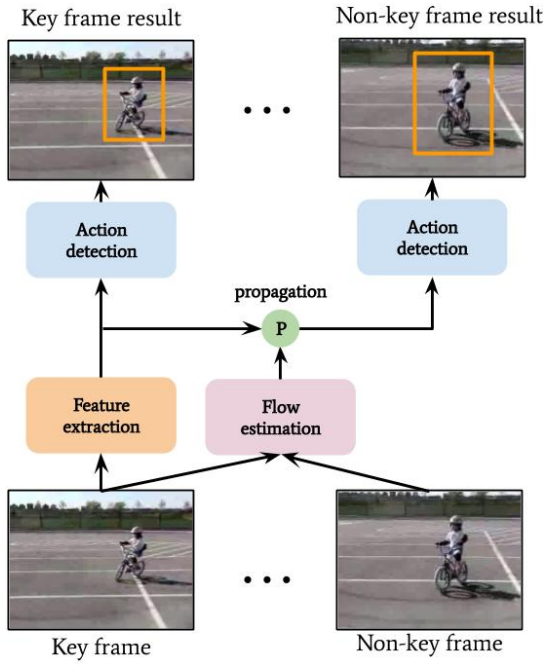

Figure 2: Illustration of the motion-guided action detection based on [27].

3.2 Motion-aware appearance feature

Our proposed method embeds motion information directly into the appearance feature for efficiency gains. We hypothesize that such two-in-one approach can have a comparable improvement for action detection without requiring a separate motion network which doubles the parameters and computation.

We choose to model the motion between two frames implicitly by the difference between their CNN features. The intuition behind is that negative and positive values in the feature difference encode locations where the body parts disappear and appear respectively, thus implicitly capturing specific motion patterns. Such a simple technique has been utilized in [26]. In our framework, the element-wise difference between the feature map of a key frame and its successive non-key frame is first computed. We then use this motion representation to amplify the appearance feature of the non-key frame via element-wise addition. Finally, the amplified appearance with motion awareness is fed to the detection sub-network. Currently we only amplify appearance features for non-key frames.

4 EXPERIMENTS

4.1 Experimental Setup

Datasets. We evaluate the proposed framework on the UCF-101-24 [25] dataset. It is a subset of UCF-101 which is composed of realistic action videos across 101 action classes from YouTube. The UCF-101-24 consists of 24 classes in 3207 videos with frame-level localization annotations. We follow the work of Singh et al.[24], using 2290 of these videos for training and the remaining ones for testing. For the moment, we trim the testing set and evaluate detection results only on video frames with action labels.

Evaluation metrices. The frame-level mean average precision (mAP) and frame-per-second (fps) are used to evaluate detection performance. The IoU threshold which determines whether a predicted box matches the groundtruth is fixed at 0.5 throughout our experiments.

Implementation details. We employ ResNet-101 [10], modified R-FCN [2] and reduced FlowNet [3] models for feature extraction, action detection and flow estimation respectively. The flow transformation sub-network consists of four convolution layers. The input to this sub-network is a 11-channel tensor obtained from the concatenation of the original flow field (2), RGB image of the non-key frame (3) and key frame (3), and their channel-wise difference (3). In the first three layers, 3x3 kernels are used and interleaved with ReLU non-linearity. The number of output channels is 16, 32, and 2 respectively, The output of the third layer is then concatenated with the original 2-channel flow, followed by a 1x1 convolution to output the transformed two-channel flow.

The entire system consisting of the above sub-networks is trained end-to-end. During training, from each mini-batch a pair of nearby video frames (I_r and I_i) is randomly sampled, one being the reference frame (similar to a key frame used during inference). We set I_r and I_i to be 9 frames apart at maximum. The deep feature map f_r is first obtained from the reference frame, while FlowNet runs on both frames to estimate the flow field. The estimated flow, after going through the transformation sub-network, is used to propagate f_r to f_i. The element-wise difference of f_i and f_r is used to augment f_i, which will be the final feature map fed to the detection sub-network. The incurred localization and classification losses are then back-propagated to update all components of all sub-networks. Here, we use ResNet with ImageNet pre-training. FlowNet is pre-trained on the Flying Chair dataset [3].

	mAP	Speed (fps)
Baseline [27]	65.92	33
+trans. flow	67.20	33
+trans. flow +motion amp.	68.84	32

Table 1: Performances of our variant architectures. The reported run time is the sum of pre-processing, network inference and post-processing time.

In both training and inference, input images are resized to 600x800 and 300x400 for the feature extraction and flow sub-network respectively. Training is conducted by stochastic gradient descent (SGD) for 96K iterations, where in the first 64K and the last 32K iterations learning rates are set to 5×10^{-4} and 5×10^{-5} respectively. We train our model using a single NVIDIA GeForce GTX 1080 Ti GPU for approximately 12 hours. During inference, every 10^{th} frame is sampled as a key frame for simplicity, whose deep feature is propagated to the successive 9 frames.

4.2 Results

We conduct experiments with different variations of architectures. Table 1 reports the performance of three experimented architectures. We take the original deep feature flow implementation as the baseline method which consists of only feature propagation.

To inspect the effect of flow transformation, we compare the resulted flow fields against those produced by the baseline method. Figure 3 shows some visualizations of the pre- and post-transformed flows along with their associated RGB images. Even without ground-truth flows to supervise training the flow transformation layers, we observe that using losses incurred by solely detection enables transformed flow to learn capturing motions more smoothly and correctly. This has a direct impact on the overall detection accuracy, as accurate flows facilitate more precise feature propagation for non-key frames.

Next, we analyze the effect of the motion-aware appearance features. From table 1, it is shown that incorporating the implicit motion representation on top of our flow transformation improves the detection result by 2.7 mAP from the baseline. The mAP for each action class is reported in figure 4. It can be observed that the implicit motion significantly boosts certain action classes such as volleyball spike and golf swing, etc. that perform poorly by the baseline method. Some of these examples are depicted in figure 5 where our method is able to capture action when the baseline method fails. The gain can be explained by our feature amplification, which makes appearance features be aware of specific motion patterns.

Finally, in table 2 we compare our detection results against state-of-the-art methods [12][24] with real-time performance, as documented by El-Nouby et al. [4] and Singh et al. [24]. Our best performing model outperforms the other top performers in terms of run time while retaining comparable accuracy. It should be noted, as pointed

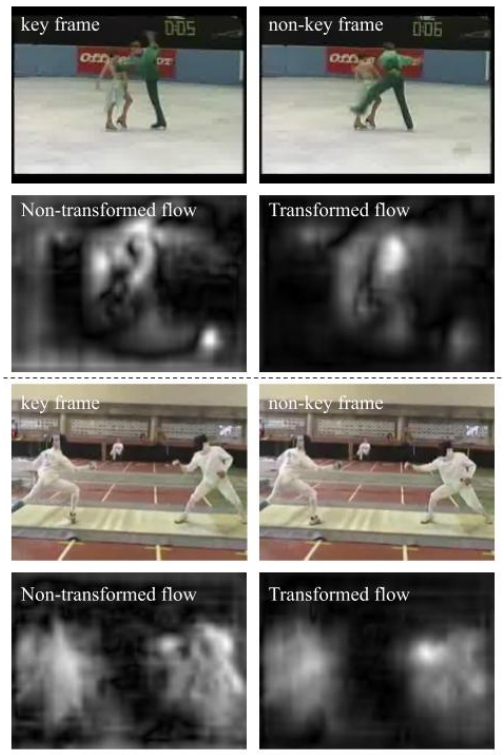

Figure 3: Effect of flow transformation. Action examples of Ice Dancing and Fencing indicate that transformed flows better capture motions (leg swing in Ice Dancing and arm movement of the right person in Fencing).

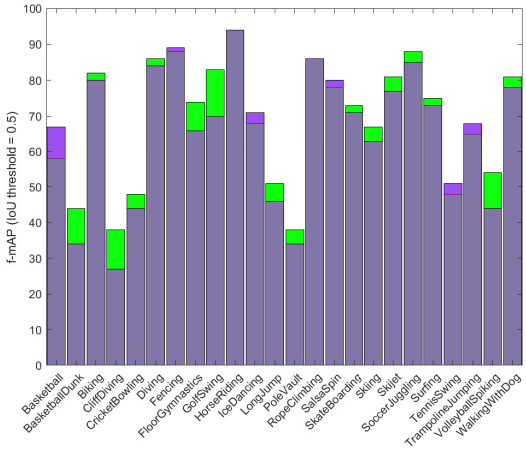

Figure 4: Detection results comparison (mAP) on individual action classes. Green bins indicate the amount our method improves upon the baseline; vice versa for the purple bins.

	mAP	Speed (fps)
Singh et al.[24] RGB	64.96	40
Singh et al.[24] RGB+Fast flow	65.66	28
Singh et al.[24] RGB+Acc. flow	68.31	7
Kalogeiton et al.[12]	69.50	25-30
Ours (+flow trans.+motion amp.)	68.84	32

Table 2: Comparisons of our best architecture with state-of-the-arts. Singh et al. apply two types of flow estimator; the fast one enables real-time performance while the accurate one provides higher accuracy.

out by El-Nouby et al., that other methods test on untrimmed videos as their frameworks handle both spatial and temporal detection. In other words, they could suffer from the disadvantage of having a greater chance of false positives on unlabeled action frames.

5 CONCLUSION

In this paper we propose an integrated action detection framework with real-time performance. By exploiting temporal coherence between video frames, we guide appearance features efficiently by motion which significantly improves detection run time. Additionally, we adopt a simple yet effective motion cue in the appearance stream, implicitly enabling motion awareness. This not only boosts detection accuracy, but also saves computation from having to utilize a two-stream network. We demonstrate that our proposed method is faster (up to 33 fps) than all previous works to our best knowledge while being able to achieve comparable accuracy with other best performers. In future work, we will include the functionality of temporal localization to form a complete spatio-temporal action detection pipeline. We will also perform adaptive key frame selection as opposed to the fixed sampling scheme currently being used. Lastly, more sophisticated motion representations will be considered. We will also perform thorough evaluation on untrimmed videos and more public datasets for a fair comparison.

6 ACKNOWLEDGMENTS

This work was supported by the H2020 Innovative Training Network (ITN) project ACHIEVE (H2020-MSCA-ITN-2017: agreement no. 765866).

REFERENCES

[1] Kai Chen, Jiaqi Wang, Shuo Yang, Xingcheng Zhang, Yuanjun Xiong, Chen Change Loy, and Dahua Lin. 2018. Optimizing video object detection via a scale-time lattice. In *IEEE Conference on Computer Vision and Pattern Recognition.*

[2] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. 2016. R-FCN: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems.*

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

[4] Alaaeldin El-Nouby and Graham W Taylor. 2018. Real-time end-to-end action detection with two-stream networks. *arXiv preprint arXiv:1802.08362.*

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

[6] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. 2017. Detect to track and track to detect. In *IEEE International Conference on Computer Vision.*

Figure 5: Action detection results by our proposed method. The first column corresponds to detection results on the key frames. The other four columns correspond to results of the following frames obtained from the propagated features. The presence of an orange dot in each image indicates when the baseline method fails to detect action.

[7] Raghudeep Gadde, Varun Jampani, and Peter V Gehler. 2017. Semantic video cnns through representation warping. In *IEEE International Conference on Computer Vision*.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[9] Wei Han, Pooya Khorrami, Tom Le Paine, Prajit Ramachandran, Mohammad Babaeizadeh, Honghui Shi, Jianan Li, Shuicheng Yan, and Thomas S Huang. 2016. Seq-NMS for video object detection. *arXiv preprint arXiv:1602.08465*.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition*.

[11] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. 2015. Spatial transformer networks. In *Advances in neural information processing systems*.

[12] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. 2017. Action tubelet detector for spatio-temporal action localization. In *IEEE International Conference on Computer Vision*.

[13] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, et al. 2018. T-CNN: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology*.

[14] Mason Liu and Menglong Zhu. 2018. Mobile video object detection with temporally-aware feature maps. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. SSD: Single shot multibox detector. In *European Conference on Computer Vision*.

[16] Eunbyung Park, Xufeng Han, Tamara L Berg, and Alexander C Berg. 2016. Combining multiple sources of knowledge in deep cnns for action recognition. In *IEEE Winter Conference on Applications of Computer Vision*.

[17] Xiaojiang Peng and Cordelia Schmid. 2016. Multi-region two-stream R-CNN for action detection. In *European Conference on Computer Vision*.

[18] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*.

[20] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. In *International Journal of Computer Vision*.

[21] Suman Saha, Gurkirt Singh, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. 2016. Deep learning for detecting multiple space-time action tubes in videos. In *British Machine Vision Conference*.

[22] Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*.

[23] Gurkirt Singh, Suman Saha, and Fabio Cuzzolin. 2018. Predicting Action Tubes. In *Proceedings of the European Conference on Computer Vision*.

[24] Gurkirt Singh, Suman Saha, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. 2017. Online real-time multiple spatiotemporal action localisation and prediction. In *IEEE International Conference on Computer Vision*.

[25] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *CRCV-TR-12-01*.

[26] Shuyang Sun, Zhanghui Kuang, Lu Sheng, Wanli Ouyang, and Wei Zhang. 2018. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[27] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. 2017. Deep feature flow for video recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.