Focused Attention for Action Recognition

Vladyslav Sydorov
vladyslav.sydorov@inria.fr
Karteek Alahari
karteek.alahari@inria.fr
Cordelia Schmid
Cordelia.Schmid@inria.fr

Univ. Grenoble Alpes, Inria, CNRS,
Grenoble INP, LJK, 38000 Grenoble,
France

Abstract

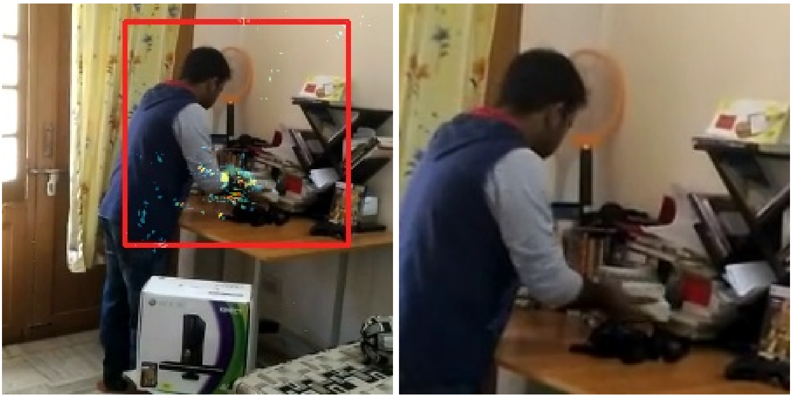
Current state-of-the art approaches to action recognition emphasize learning ConvNets on large amounts of training data, using 3D convolutions to process the temporal dimension. This approach is expensive in terms of memory usage and constitutes a major performance bottleneck of existing approaches. Further, video input data points typically include irrelevant information, along with useful features, which limits the level of detail that networks can process, regardless of the quality of the original video. Hence, models that can focus computational resources on relevant training signal are desirable. To address this problem, we rely on network-specific saliency outputs to drive an attention model that provides tighter crops around relevant video regions. We experimentally validate this approach and show how this strategy improves performance for the action recognition task.

1 Introduction

The arrival of diverse, large-scale datasets [1, 7, 9, 19] has paved the way for the success of large 3D CNN architectures [1, 34] for video-based action recognition. These architectures operate on batches of consecutive video frames and are characterized by extensive 3D convolutions over space and time.

The inputs to these computationally-heavy models have a large temporal dimension and, to constrain the already large number of parameters to reasonable limits, the spatial extent is relatively limited. This means the original video has to be downscaled, which causes some of the finer details to be lost. Consequently the loss of the details may negatively affect fine-grained recognition abilities of the network. Additionally, the annotations for videos in aforementioned datasets simply indicate that an action is happening at some point of time, without spatial localization. Under such conditions, one cannot fully exploit the receptive field of the network, as irrelevant information will inevitably be included along with useful video features.

We propose a general method that alleviates this problem of detail degradation and helps to further adapt the model to the video recognition task. First, a measure of saliency is defined to compare relative "importance" of video regions to solving the task. The measure of saliency is then used to extract focused representations in a weakly-supervised way, which are processed by an extended network to improve recognition performance.



(a) Frame at the input of the network (b) Input refined by the attention procedure

Figure 1: Example of attention on Charades action recognition dataset. a) Saliency scores (displayed as a heatmap) are localized around the object, a box maximizing the saliency measure within is selected. b) The network is provided with the relevant crop of the video, and can process it at a higher resolution.

Consider the example shown in Figure 1(a). Here the actions occurring are "Washing something with a towel" and "Tidying up a table" and the important regions of the image are the table, the towel and the person interacting with them. The relevant part of the video occupies only a small region of the input frame, while a large fraction of the frame is occupied by the background and is of lesser relevance to recognizing the action.

We utilize a saliency measure to source focused inputs from the original video data, which contain a more relevant portion of the video, as shown in Figure 1(b). The input dimensionality of the model remains fixed, but the tighter crop around the salient region allows the network to compute more expressive statistics of the relevant portion of the video and analyze it in finer detail.

In this paper we present a framework to leverage the focused attention, and show how providing the model with an additional focused view of the inputs allows it to better adapt to the wide variety of possible videos. Specifically, we obtain network-driven saliency estimations by back-propagating through a pretrained network. This allows us to define an attention procedure, through which we find the video region that encloses the most salient area. With the help of this attention procedure we extract these salient regions, and provide them to the network as an additional modality. This effective use of attention allows us to improve action recognition performance.

2 Related Work

Action Recognition: Video action recognition has evolved considerably since the introduction of deep networks. Starting from the highly successful two-stream networks [22], a selection of approaches emerged that focuses on bringing together spatial and temporal information. Some of them model feature evolution with RNNs [12, 38], others try to find a robust technique for sampling the video frames [13, 31, 32, 42].

The temporal aspect of video recognition is more directly captured by 3D convolutional networks [28, 29]. Recently, the I3D family of deep networks [1] has managed to model long-range temporal dependencies by employing extensive 3D convolutions. These networks have achieved superior performance compared to previous attempts.

Our approach is generally applicable to any CNN. We utilize it in conjunction with the I3D model in this paper.

Attention for video recognition: This problem has been pursued in various forms — reinforcement learning for sampling frames [37], guided pooling of temporal [5, 14] and spatial [17] features.

Self-attention approaches, initially described for language modeling [30], have proven especially useful. A drop-in extension to I3D via non-local pooling [34] performs self-attention within the spatiotemporal space of the CNN input. Video action transformers [6] modify this approach to consider only the relationships between human regions and rest of the input volume. In the same spirit, Sun *et al.* study pairwise interactions between 1×1 convolutional features [27]. Long-term feature banks [36] consider the temporal aspect by accumulating temporal features and applying attention pooling between accumulated and individual frame features. Self-attention has also been utilized to model long-term interactions between memory states of a 3D LSTM [35].

An important difference between the methods described above and our approach is the way inputs are treated. Contemporary approaches focus on discovering relationships between video features while staying within the bounds of an input RGB volume. In contrast, we follow the attention cues to the original videos, and provide more informative inputs. Essentially, we provide a different perspective on how attention can be utilized.

Processing videos in different modalities: Videos are commonly processed by CNNs in different modalities. Originally, optical flow has been used almost universally as an additional stream [22], then pose features have been considered for this purpose [2, 43]. Recently SlowFast has looked at RGB features via two branches at different temporal speeds [4].

We consider "focused" inputs as a similar yet distinct modality of the input data, as they provide additional information about finer details to the network.

Saliency/unsupervised methods: A lot of work has been done on understanding CNN behavior and visual explanation for decisions they make. Erhan *et al.* [3] seek to visualize filters by maximizing activations. Simonyan *et al.* [24] acquire image-specific saliency maps via a single back-propagation pass and use them to obtain segmentation masks. Guided back-propagation further improves the quality of saliency maps [25, 39]. Zhou *et al.* [41]generate class activation maps to identify discriminative image regions, while GradCAM [16] extends this technique to provide high-quality visualization with strong localization capabilities.

While we are not aiming to produce class maps, the unifying idea of utilizing the network itself to produce saliency is at the core of our approach. We leverage the gradients to identify the salient regions in the video in a network-specific way.

Saliency methods have also been used to guide recognition. Sudhakaran *et al.* [26] employ Imagenet-pretrained network to obtain CAM maps [41], which are then sed to rescale video features, in a form of soft spatial attention. In contrast, we use task-specific saliency and hard attention.

Hard Attention: In the work of Jaderberg *et al.* [10], spatial transformers act as a differentiable attention mechanism that aid image classification, their network learns to crop patches from the input image. Concurrently to our work, Katharopoulos *et al.* [11] utilize learnable attention to efficiently sample informative patches from high-resolution images, thus reducing computation time and memory footprint.

3 Focusing attention

3.1 Motivation

The state-of-the-art approach to the problem of action recognition in videos is to process the video with a complex CNN based model to answer the question of whether a certain action is present in it. We define a video as an RGB volume of dimension $T \times H \times W \times 3$, where T is the temporal dimension, H and W are height and width of the frames respectively. When solving an action recognition task, we assume that the video contains certain spatiotemporal regions, which can be leveraged to recognize the action occurring in it. In other word, this RGB video contains some relevant sub-regions and some portion of irrelevant background data.

Current state-of-the-art models assume fixed size inputs. These CNN networks are also heavy, with a big receptive field in the temporal dimension T, which implies that the spatial dimensions H and W are limited due to the memory constraints. For example, I3D as described in [1, 4, 36] has inputs of size 224×224 only.

Fitting the video into these dimensions involves scaling and cropping operations, during which some of the finer details are inevitably lost. A common tactic used both in video and image recognition [1, 23] is to crop the center region from the input. This accounts for the common bias of recorded media to contain the concept of interest in the center, but is in the essence a heuristic operation — there is no guarantee that important features are in fact contained in the center. Interestingly, a lot of recent video action datasets focus on close-up [7, 15] or egocentric [21] videos, both settings in which this problem is less likely to occur.

We state that an attention operation will be helpful for the purpose of extracting useful data from the video inputs and fully exploiting the receptive field of the network. Moreover, we think that an attention operation should be tied to the network itself, instead of depending on externally sourced cues like human gaze data or additional object detectors.

3.2 Saliency based attention

To be able to discern relevant regions versus background we need to define an "importance" or "saliency" metric. "Saliency" has multiple connotations and in general terms can be thought of as a measure of where would a person or a model "look" to make a decision regarding the input.

Let us consider an input space $\mathbb{R}_+^{T \times H \times W \times 3}$ of RGB videos. An action recognition saliency over video V may be defined as a function $S(V) : \mathbb{R}_+^{T \times H \times W \times 3} \to \mathbb{R}_+^{T \times H \times W}$, which assigns high values to regions which are important for making a prediction and low values to others. With this spatiotemporal saliency metric we can measure the relative importance of each region. A saliency function should necessarily be dependent on the network parameters. The choice of saliency function will be discussed in section 4.1.

Processing a video with a CNN involves preprocessing steps to make it compatible with the CNN architecture. The video is transformed with a heuristic spatial transform:

$$V_{\text{in}} = F(V, \theta) : \mathbb{R}^{T \times H \times W \times 3} \to \mathbb{R}^{T \times H_{\text{in}} \times W_{\text{in}} \times 3}, \tag{1}$$

which commonly consists of individual crop and resize transforms, and where θ defines the parameters of these transforms. The parameters of these transforms do not take the contents into account and to address this we propose to employ a saliency based attention transform

A to obtain an input, focused on "important" regions:

$$V^s_{\text{in}} = A(V, S(V)), \tag{2}$$

which produces a video volume $V^s_{\text{in}} \in \mathbb{R}^{T \times H_{\text{in}} \times W_{\text{in}} \times 3}$, similar to the V_{in} obtained with heuristic operation (1), but prioritizes maximization of saliency score within.

We have described a general approach so far, which boils down to the usage of network-dependent attention for the purpose of providing a CNN with focused inputs. To define the approach precisely, we must choose a saliency function S and an attention transform A. In the next section, we present these details.

4 Attention for Action Recognition

4.1 Saliency function

We aim to efficiently locate regions important for action recognition in a class-independent way. Accordingly, we obtain saliency by taking a derivative with respect to the inputs. The technique consists in essence of a single back-propagation pass through the network, which allows for on-the-fly extraction of focused inputs during training. It was originally employed in a class-specific way, to visualize a network's notion of object class [24].

The choice of the function being back-propagated is important — we experiment with using the loss function. The intuition for utilizing the loss function as a guidance is the following: loss minimization is the way of training networks to extract useful signal from the video, hence regions that affect the loss function the most contribute to the network predictions the most.

We train our model with binary cross-entropy loss per-class:

$$L_{\text{CE}}(y_n, \hat{y}_n) = y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n), \tag{3}$$

where $y_n \in \{0, 1\}$ are ground truth labels and $\hat{y}_n \in [0, 1]$ are class predictions. It is then possible to obtain saliency during training by back-propagating L_{CE}. The downside of minimizing the training loss is that it requires knowledge of the ground truth labels y_n, which are not available for the evaluation step. To account for this limitation, we choose to obtain saliency predictions by back-propagating the negative entropy over class predictions:

$$L_{\text{E}}(\hat{y}_n) = \hat{y}_n \log \hat{y}_n. \tag{4}$$

In trained networks, low entropy of class outputs can be thought of as a mode, where the network is certain of the prediction being made. In accordance with the definition of saliency as a decision making mechanism, it would make sense to use L_{E} to guide the saliency. We find that after pretraining, saliency maps obtained with both L_{E} and L_{CE} are similar, thus we can use L_{E} as a proxy for L_{CE}.

After propagating the cross-entropy/entropy back towards the video, we obtain the Jacobian $J \in \mathbb{R}^{T \times H_{\text{in}} \times W_{\text{in}} \times 3}$ of the same output dimensionality as the RGB input. To get a positive per-pixel saliency magnitude we apply the L2-norm at each pixel location as follows:

$$s_{tij} = \|(J_{tij1}, J_{tij2}, J_{tij3})\|, \tag{5}$$

where J_{tijc} are indexed values of a Jacobian, thus obtaining $S(V_{\text{in}}) = s$, $s \in \mathbb{R}_+^{T \times H_{\text{in}} \times W_{\text{in}}}$.

4.2 Attention function

We obtain our saliency measure directly from the network, i.e., we compute it with respect to and of the same dimensionality as the network inputs V_{in}. We assume that saliency is an inherent property of the video itself and is preserved by the operation F. This assumption allows us to reason about the quantity $S(V)$ by observing $S(V_{\text{in}})$.

We search a volume of dimension $T \times B_h \times B_w$ that maximizes the sum of saliency scores inside. Concretely we first reduce the temporal dimensions via max-pooling $s_{ij} = \max_{t=1\ldots T}(s_{tij})$. Next, we select a fixed size bounding box $B = (B_t, B_l, B_h, B_w)$, by finding B_t, B_l as follows:

$$B_t, B_l = \text{argmax}_{B_t, B_l} \sum_{i=B_t}^{B_t+B_h} \sum_{i=B_l}^{B_l+B_w} s_{ij}, \tag{6}$$

where B_t, B_l are the coordinates of the top left corner and B_h, B_w are the height and the width of the box respectively. The coordinates of the box are fixed across all time steps T.

We utilize the parameters θ of the original transformation F to obtain box coordinates $B^o = (B^o_t, B^o_l, B^o_h, B^o_w)$ with respect to the original video V and consider that B^o would also capture the maximum saliency in $S(V)$, if we had access to it directly. We crop the volume enclosed by B^o from the original video and scale it to $H_{\text{in}} \times W_{\text{in}}$. During the training phase, additional data augmentation functions can be applied (e.g., random crops, left-right flip, photometric adjustments). When accessing original video we reapply these augmentations. This procedure allows for efficient extraction of focused regions during training and evaluation stages.

4.3 Processing V_{in} and V^s_{in} inputs

The attention operation as described above provides a different, if similar, facet of the input video to the model. Even assuming that the saliency measure we obtain is perfect, there is always a possibility that some context information is lost. Additionally, if the video crop V_{in} is already a perfect representation then there is little need to apply a transformation. Another important issue is that the model has been trained by only applying the heuristic F and if we were to change the spatial transformation without further adjustments, the newly introduced domain shift might hurt the performance.

In lieu of these considerations, we think it is reasonable to keep both the original transformed data and the focused inputs and process them together. This can be thought of as another take on the approach of processing the video stream in different input modalities, which is a reasonable concept, when applied to video understanding [4, 40].

When training the full model, we first train the network with the usual transform F until convergence, we call the result a *Base* model. We explore two ways of processing the inputs together: temporal concatenation and late fusion, which allows us to use the same architecture for both modalities.

Temporal concatenation: we extend the input by concatenating the original input V_{in} and the focused region V^s_{in} along the temporal axis. This step is possible because the modalities of the inputs are similar. We finetune the *Base* model on temporally extended inputs.

Late fusion: we process the two modalities V_{in} and V^s_{in} via separate networks, both initialized from *Base*, and aggregate the output scores via mean-pooling. When finetuning, we keep the branch that processes the original input V_{in} fixed.

In other words, once the network training procedure reaches the performance limit, determined by the provided data, we allow further performance increase by accounting for important regions via extended inputs.

5 Experiments

We utilize a standard 3D CNN architecture [33], where the base model is ResNet50 [8], inflated into a network with 3D convolutions in an I3D [1] fashion.

5.1 Data and setup

For our experiments we use the Charades dataset [19], containing 9 848 videos across 157 action classes. In each video, a person can perform one or more actions. We evaluate the video classification task, which involves recognizing all the actions in blea video, without temporal localization. We train and evaluate on the publicly available subset of Charades following standard protocol [33, 36]. RGB frames are extracted at the rate of 8FPS.

5.2 Network Details

We use a model finetuned from Kinetics to test our method. We call it *Charades-Base*.

Kinetics-Base: As the base model we utilize ResNet50-I3D [18], which is publicly available along with the source code and pretrained weights. This model has been trained on Kinetics from scratch for 300 epochs. It achieves 64.01% (83.70%) top-1 (top-5) accuracy on the Kinetics validation set. The base network accepts inputs $T_{\text{in}} \times H_{\text{in}} \times W_{\text{in}} \times 3$, representing T_{in} RGB frames, where $H_{\text{in}} = W_{\text{in}} = 224$. At the last layer of the network, the logit outputs are of dimension $T_{\text{out}} \times N_{\text{class}}$. For $T_{\text{in}} = 64$, the corresponding $T_{\text{out}} = 7$.

Charades-Base: We adopt the Kinetics base model for Charades and introduce a different loss function. Specifically, a cross-entropy loss (L_{CE}) over network predictions $\hat{Y}$ is used, to accommodate for the multilabel nature of the Charades dataset. In the last layer of the network we accordingly replace the softmax operation with a sigmoid. The Charades dataset has temporal annotations Y_{frame} for every video frame along with per-video labels Y_{video} — we incorporate this by minimizing a sum of two loss components:

- Video loss: $L_{\text{CE}}(Y_{\text{video}}, M(\hat{Y}))$, where $M : T_{\text{out}} \times N_{\text{class}} \rightarrow N_{\text{class}}$ is a mean-pooling operation across the temporal dimension of the network outputs.

- Frame loss: $L_{\text{CE}}(Y_{\text{frame}}, U(\hat{Y}))$, where $U : T_{\text{out}} \times N_{\text{class}} \rightarrow T_{\text{in}} \times N_{\text{class}}$ is a bilinear interpolation operation that upsamples the logit outputs to match the input temporal dimension.

5.3 Implementation

Saliency. We explore several bounding box sizes for the Charades validation set and find that for our network with expected spatial inputs $H_{\text{in}} = W_{\text{in}} = 224$, $B_h, B_w = 128$ performs the best. These zoomed-in boxes enclose roughly 30% of the original area.

Training. First, we finetune the baseline model from Kinetics to the Charades dataset without using focused inputs. We then apply the proposed approach to further improve the recognition performance.

Charades-Base: We initialize the model with *Kinetics-Base* weights. We cover temporal extents T_{in} of 16, 32 and 64. We use mini-batch SGD with momentum set to 0.9 and batch sizes are set accordingly to 20, 16, 12 for $T_{\text{in}} = 16, 32, 64$ respectively. Training lasts for 50 epochs, with an initial learning rate 0.375 that is decreased by a factor of 10 at epochs 15 and 40.

Multi-modality models: From the *Charades-Base* model we tune several attention and ablation models for additional 25 epochs, in both *temporal concatenation* and *late fusion* versions. The initial learning rate is 0.02, the batch size is 8, and the learning rate is dropped by a factor of 10 at epoch 15. We finetune BN layers, small batch size notwithstanding. At $T_{\text{in}} = 64$ GPU memory requirements for *late fusion* and *temporal concatenation* are 24GB and 32GB respectively. We thus train *late fusion* models on 2 Titan X GPUs and *temporal concatenation* models on 2 Tesla P100s.

The *Repeat* model is tuned from *Charades-Base*, but no attention step is executed, instead the V_{in} inputs are repeated in the temporal dimension. The *Random crop* and *Center crop* experiments perform attention step without utilizing saliency, i.e., during training a random or a centered $B_h \times B_w$ cuboid respectively is cropped from the input volume. When training *Attention* models, a choice of saliency function can be made at both training and evaluation stages. To denote this choice, we utilize a two-index notation L, where the first index corresponds to the saliency function used during training, the second to the function used during evaluation. We train $L_{\text{E / E}}$, $L_{\text{CE / CE}}$ and $L_{\text{CE / E}}$ *Attention* models. We emphasize that the $L_{\text{CE / CE}}$ experiment only serves as a reference for what attention guided by training loss can achieve, since utilizing L_{CE} during evaluation involves accessing ground truth data.

During training, a left-right flip is applied with $p = 0.5$, while the transform F amounts to resizing the video to 256×256 and randomly cropping a 224×224 cuboid.

Evaluation. During evaluation, F is composed of resizing the video to 256×256 and then obtaining a 224×224 center crop. At test time we sample 10 clips per video and combine the predictions using max pooling, following prior work [33, 36].

6 Results

In this section we experimentally evaluate the models on the Charades action recognition task. We show that our baseline model performs on par with the state-of-the art. Most importantly, we show that, when trained on videos augmented by focused attention, the performance of the model improves over the baseline. Finally, we perform several ablation studies.

6.1 Baseline model

In Table 1 we show the performance of our *Charades-Base* baseline at different temporal extents T_{in} as well as other approaches. We include neural network approaches that were state-of-the-art for Charades before appearance of I3D [20, 33, 42] and show that I3D outperforms them. Next, we consider a baseline I3D network from a recent publication [33] and demonstrate our network performing at the same level, despite being pretrained on only

Method	Model	Pretraining	mAP
2-Stream [20]	VGG16	ImageNet	18.6
Asyn-TF [33]	VGG16	ImageNet	22.4
Multiscale TRN [42]	Inception	ImageNet	25.2
I3D baseline ($T_{\text{in}} = 32$) [33]	ResNet50-I3D	ImageNet+Kinetics	31.8
Ours ($T_{\text{in}} = 16$)	ResNet50-I3D	Kinetics	27.8
Ours ($T_{\text{in}} = 32$)	ResNet50-I3D	Kinetics	30.3
Ours ($T_{\text{in}} = 64$)	ResNet50-I3D	Kinetics	31.6

Table 1: Baseline performance. We show the mean Average Precision (mAP%).

Kinetics dataset. Recent works utilizing attention report higher performance on this task, but employ long-range temporal techniques such as spatiotemporal graphical models [33] or memory banks [36].

6.2 Attention

Method	Temp. concat.	Late fusion
Charades-Base	31.6	31.6
Repeat	31.7	31.6
Random crop	31.4	30.6
Center crop	31.6	31.0
Attention $L_{\text{CE / CE}}$	33.2	32.9
Attention $L_{\text{CE / CE}}$ (upscaled)	33.2	32.7
Attention $L_{\text{CE / E}}$	33.2	33.0
Attention $L_{\text{CE / E}}$ (upscaled)	33.1	32.8
Attention $L_{\text{E / E}}$	33.3	33.1

Table 2: Our performance at $T_{\text{in}} = 64$, mAP%

In Table 2 we present the experimental results of the proposed attention approach. *Charades-Base* and *Repeat* are the two baseline experiments, where *Charades-Base* corresponds to the baseline performance of the model without using additional modality and *Repeat* allows us to reason about the stacked model performance in the absence of the attention transform A.

We see that *Random crop* and *Center crop* strategies do not provide any useful signal to the network, in fact randomly cropping the volume results in a lower performance. This indicates that the attention transform A should be guided by a reasonable, input-dependent saliency function.

The Attention models $L_{\text{CE / CE}}$, $L_{\text{E / E}}$ and $L_{\text{CE / E}}$ outperform the baselines and allow us to confirm the usefulness of the saliency-guided attention step. We see that L_{CE} and L_{E} saliencies are very similar, as evidenced by a very small difference in performances of $L_{\text{CE / CE}}$, $L_{\text{E / E}}$ and $L_{\text{CE / E}}$.

We also perform ablation studies "$L_{\text{CE / CE}}$ (upscaled)" and "$L_{\text{CE / E}}$ (upscaled)", for which we do not query the original video for the higher resolution inputs and instead simply upscale the focused video. These experiments show lower performance than $L_{\text{CE / CE}}$ and $L_{\text{CE / E}}$, proving that querying the original video is important.

7 Conclusion

In this paper we introduce a new approach to leverage attention for action recognition. Instead of being constrained within the bounds of input data dimensionality, we take a step further and leverage the original video data. We demonstrate that the preprocessing steps of a video action pipeline have a notable effect on the quality of results. Consequently, we allow the network-dependent saliency measure to guide the preprocessing operation to select regions of the data that are more useful for the recognition task. Crucially, these regions can be processed in higher detail, allowing to access additional information, which is not used by conventional attention approaches.

Acknowledgements. This work was supported in part by the Indo-French project EVEREST (no. 5302-1) funded by CEFIPRA, ERC advanced grant ALLEGRO, and a grant from ANR (AVENUE project ANR-18-CE23-0011). We gratefully acknowledge the support of NVIDIA with the donation of GPUs used for this work.

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, 2017.

[2] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. Potion: Pose motion representation for action recognition. In *CVPR*, 2018.

[3] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *Technical Report, University of Montreal*, 2009.

[4] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast Networks for Video Recognition. *arXiv:1812.03982*, 2018.

[5] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. In *NIPS*, 2017.

[6] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. Video Action Transformer Network. *arXiv:1812.02707*, 2018.

[7] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "Something Something" video database for learning and evaluating visual common sense. In *ICCV*, 2017.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[9] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[10] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015.

[11] Angelos Katharopoulos and François Fleuret. Processing Megapixel Images with Deep Attention-Sampling Models. In *ICML*, 2019.

[12] Zhenyang Li, Efstratios Gavves, Mihir Jain, and Cees G. M. Snoek. VideoLSTM convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 2018.

[13] Ji Lin, Chuang Gan, and Song Han. Temporal shift module for efficient video understanding. *arXiv:1811.08383*, 2018.

[14] Xiang Long, Chuang Gan, Gerard de Melo, Jiajun Wu, Xiao Liu, and Shilei Wen. Attention clusters: Purely attention based local feature integration for video classification. In *CVPR*, 2018.

[15] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Yan Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, and Aude Oliva. Moments in time dataset: one million videos for event understanding. *PAMI*, 2019.

[16] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization. In *ICCV*, 2017.

[17] Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov. Action recognition using visual attention. In *ICLR (workshop track)*, 2016.

[18] Gunnar A. Sigurdsson and Abhinav Gupta. PyVideoResearch. 2018. URL `https://github.com/gsig/PyVideoResearch`.

[19] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016.

[20] Gunnar A Sigurdsson, Santosh Divvala, Ali Farhadi, and Abhinav Gupta. Asynchronous temporal fields for action recognition. In *CVPR*, 2017.

[21] Gunnar A. Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Charades-ego: A large-scale dataset of paired third and first person videos. *arXiv:1804.09626*, 2018.

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

[23] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015.

[24] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv:1312.6034*, 2013.

[25] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR (workshop track)*, 2015.

[26] Swathikiran Sudhakaran and Oswald Lanz. Attention is All We Need: Nailing Down Object-centric Attention for Egocentric Activity Recognition. In *BMVC*, 2018.

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

[28] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[29] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *PAMI*, 2018.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *NIPS*, 2017.

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

[32] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *PAMI*, 2018.

[33] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, 2018.

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

[35] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3D LSTM: A Model for Video Prediction and Beyond. In *ICLR*, 2019.

[36] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krähenbühl, and Ross Girshick. Long-Term Feature Banks for Detailed Video Understanding. In *CVPR*, 2019.

[37] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, 2016.

[38] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015.

[39] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

[40] Haochen Zhang, Dong Liu, and Zhiwei Xiong. Two-Stream Oriented Video Super-Resolution for Action Recognition. *arXiv:1903.05577*, 2019.

[41] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization. In *CVPR*, 2016.

[42] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018.

[43] Mohammadreza Zolfaghari, Gabriel L. Oliveira, Nima Sedaghat, and Thomas Brox. Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection. In *ICCV*, 2017.