Actor-Centric Relation Network

Chen Sun, Abhinav Shrivastava, Carl Vondrick,
Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid

Google Research

Abstract. Current state-of-the-art approaches for spatio-temporal action localization rely on detections at the frame level and model temporal context with 3D ConvNets. Here, we go one step further and model spatio-temporal relations to capture the interactions between human actors, relevant objects and scene elements essential to differentiate similar human actions. Our approach is weakly supervised and mines the relevant elements automatically with an actor-centric relational network (ACRN). ACRN computes and accumulates pair-wise relation information from actor and global scene features, and generates relation features for action classification. It is implemented as neural networks and can be trained jointly with an existing action detection system. We show that ACRN outperforms alternative approaches which capture relation information, and that the proposed framework improves upon the state-of-the-art performance on JHMDB and AVA. A visualization of the learned relation features confirms that our approach is able to attend to the relevant relations for each action.

Keywords: spatio-temporal action detection, relation networks

1 Introduction

Robust human action understanding will have a large impact in applications across robotics, security, and health. However, despite significant progress in visual recognition for objects and scenes [16, 27, 41, 64], performance on action recognition remains relatively low. Now that we have large, diverse, and realistic datasets such as AVA [13], SLAC [62], and Charades [49], why has action recognition performance not caught up?

Models for spatio-temporal action localization from the last few years have been mainly based on architectures for recognizing objects [12, 37, 58], building on the success of R-CNN style architectures [9, 10, 40]. However, unlike objects which can be identified solely by their visual appearance, in many cases actions can not be identified by the visual appearance of actors alone. Rather, action recognition often requires reasoning about the actor's relationship with objects and other actors, both spatially and temporally. To make this point, Figure 1 shows two actors performing different actions. Even for humans, by just looking at the cropped boxes, it is difficult to tell what actions are being performed. It is from the actors' interactions with a ball in the scene that we can tell that these

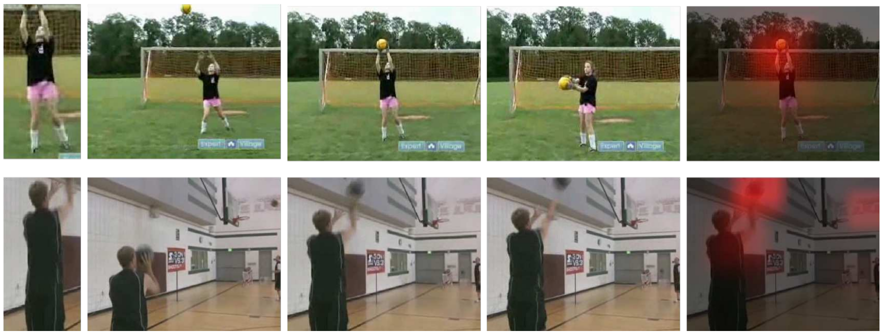


Fig. 1: Action detection is challenging even for humans without relation reasoning from the context. Only by extracting the relationship between the actor and the object (ball), and understanding how this relationship evolves over time, can one tell that the first action is catching a ball, while the second action is shooting a ball. The last column visualizes the relational heat maps generated by our algorithm.

are sports actions, and only by temporal reasoning of the relative positions, can we tell that the first actor is catching a ball and the second is shooting a ball.

Although the basic idea of exploiting context for action recognition is not new, earlier works [5, 32, 56] largely focused on the *classification* task (label each trimmed clip with an action label). For detection, where we want to assign different labels to different actors in the same scene, *actor-centric* relationships need to be extracted. Training this in a fully supervised manner would require detailed labeling of actors and relevant objects [4, 15]; such annotations can be very expensive to obtain. Therefore, we aim to build an action detection system that can infer actor-object spatio-temporal relations automatically with only actor-level supervision.

In this paper, we propose an action detection model that learns spatio-temporal relationships between actors and the scene. Motivated by the recent work of Santoro et al. [44] on visual question answering, we use neural network to compute pair-wise relation information from the actor and scene features, which enables the module to be jointly trained with the action detector. We simplify the search space of scene features to be individual cells on a feature map, and pool the actor feature to be 1×1. These simplifications allow us to compute relation information efficiently with 1×1 convolutions. A set of 3×3 convolutions are then used to accumulate relation information from neighboring locations. We refer to this approach as actor-centric relation network (ACRN). Finally, we also use the temporal context as inputs to ACRN. Such context is captured by 3D ConvNets as suggested by [13].

We evaluate our approach on JHMDB [22] and the recently released AVA dataset [13]. Experimental results show that our approach consistently outperforms the baseline approach, which focuses on the actor, and alternative approaches that employ context information. We also visualize the relation heat maps with classification activation mapping [63]. Figure 1 shows two examples of such visualization. It is evident that ACRN learns to focus on the ball and its motion over time (flattened into 2D).

The primary contribution of this paper is to learn actor-centric spatio-temporal relationships for action detection in video. The rest of the paper describes our approach and experiments in detail. In section 2, we first review related work. In section 3, we present our approach to detect human actions. In section 4, we discuss several experiments on two datasets where we obtain state-of-the-art action detection performance.

2 Related work

Action recognition. Action recognition has traditionally focused on classifying actions in short video clips. State-of-the-art methods rely either on two-stream 2D ConvNets [25, 50], 2D ConvNets with LSTMs [7, 34] or 3D ConvNets [3, 54]. While action classification in videos has been successful, it is inherently limited to short trimmed clips. If we want to address long untrimmed videos, temporal localization is necessary in addition to action classification. This requires an additional step of determining the start and end time of each action instance. Many recent state-of-the-art methods [2, 5, 60] rely on temporal proposals and classification approaches similar in spirit to recent methods for object detection [40].

However, a more detailed understanding of actions in video requires localization not only in time, but also in space. This is particular true in the case of multiple actors [13]. Many existing state-of-the-art approaches for spatio-temporal action localization [11, 37, 43, 51, 57] employ state-of-the-art object detectors [28, 40] to discriminate between action classes at the frame level. Recently, some approaches incorporate temporal context from multiple frames. This is particularly important for disambiguating actions such as “stand up” and “sit down”, which may appear identical at the single frame level. The tubelet approach [24] concatenates SSD features [28] over spatio-temporal volumes and jointly estimates classification and regression over several frames. T-CNN [18] uses 3D convolutions to estimate short tubes, micro-tubes rely on two successive frames [42] and pose-guided 3D convolutions add pose to a two-stream approach [65]. Gu et al. [13] rely on inflated 3D ConvNet (I3D) convolutions [3] for Faster R-CNN [40] region proposals and show that the use of I3D over relatively long temporal windows [55] improves the performance. The spatio-temporal separable 3D ConvNet (S3D) [59] improves the I3D architecture by observing that the 3D convolutions can be replaced by separable spatial and temporal convolutions without loss in accuracy, and that using such convolutions in higher layers of the network results in faster and more accurate models. We use the S3D [59] model as the baseline approach in this paper.

Whereas several recent approaches for spatio-temporal action localization do take into account temporal information, they ignore spatial context such as interaction with humans, objects and the surrounding scene. This results in confusion of similar actions and interactions, such as jumping and shooting a basketball.

We demonstrate that augmenting a state-of-the-art action localization approach with spatial context generates a significant performance improvement.

Context in vision. The use of context information to improve visual recognition has been extensively studied in computer vision. Early work showed that context can help scene classification [35], object detection [15, 17, 33, 39, 53], and action recognition in images [61]. In these cases, context often provides a strong prior that enables more robust recognition. While these models of context were largely hand-designed, recent investigations have studied how to learn context with deep convolutional networks [46, 47]. Spatial context has also been studied in self-supervised learning for learning unsupervised visual representations [6, 36]. Beyond images, context has been leveraged in video, in particular for recognizing actions with hand-crafted features [32] and learned representations [8, 45]. While we are also interested in recognizing human actions with context, this paper focuses on the role of context for *detection*. Importantly, since recognizing actions from crops is challenging even for humans, we believe context should play a critical role for learning robust action detection models.

Modeling the relations between objects [38, 31] and more specifically between humans and objects [12, 14] has been shown to improve the performance of recognizing relations in static images. Recent work [12] obtains state-of-the-art performance for human-action-object recognition on V-COCO [14] and HICO-DET [4]. In contrast to our approach, their model is only applied to static images and relies on full supervision of actor, action and objects as annotated in V-COCO [14] and HICO-DET [4].

3 Action detection with actor-centric relation network

This section describes our proposed action detection framework. The framework builds upon the recent success of deep learning methods for object and action detection from static images [40] and videos [37]. We note that the relational information between the actor of interest and other actors or objects are important to identify actions, but are typically ignored by recent action detection methods [37, 24]; such annotations could be time consuming to collect, and are not provided by many of the recent action recognition datasets [26, 13, 48]. Our proposed frameworks aims at explicitly modeling relations with weak actor-level supervision, with an actor-centric relation network module. Once trained, the framework can not only detects human actions with higher accuracy, but can also generate spatial heat maps of the relevant relations for each actor and action. An overview of the approach can be found in Figure 2.

3.1 Action detection framework

Our goal is to localize actions in videos. We follow the popular paradigm of frame-based action detection, where the model produces bounding-box predictions for

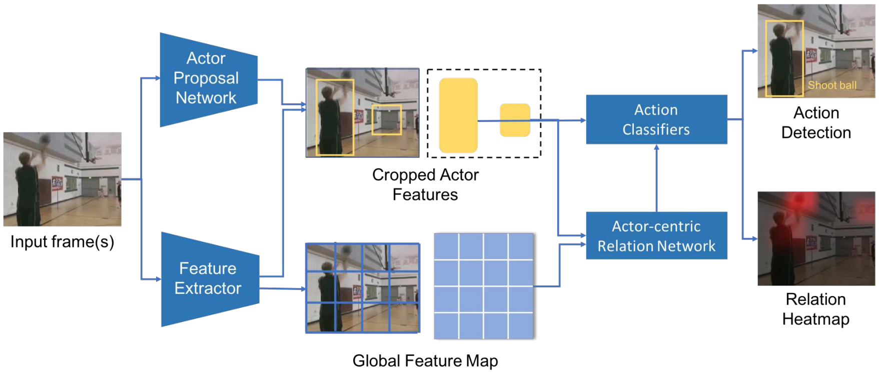


Fig. 2: Overview of our proposed action detection framework. Compared to a standard action detection approach, the proposed framework extracts pairwise relations from cropped actor features and a global feature map with the actor-centric relation network (ACRN) module. These relation features are then used for action classification.

actions on each frame individually [37, 11], and then links them into tubes as a post-processing step.

Action detection model. Our base model has two key components: actor localization and action classification. These two components are trained jointly in an end-to-end fashion. This action detection model was proposed in [13], motivated by the success of applying end-to-end object detection algorithms to action detection [37, 24]. The inputs to the base model include the key frame to generate action predictions, and optionally neighboring frames of the key frame as temporal context. The outputs of the base model include 2D bounding boxes of localized actions for the key frame. The overall architecture of the base model largely resembles the Faster R-CNN detection algorithm. For actor localization, our method uses the region proposal network (RPN) from Faster R-CNN to generate 2D actor proposals. For action classification, we use deep representations extracted from the key frame and (optionally) neighboring frames. Unlike Faster R-CNN, we do not require the actor proposal network to share the same extracted features as the action classification network, although such sharing is possible. This allows more freedom to the choice of action classification features without adding much computation overhead, as classification feature computation is usually dominated by the neighboring frames.

Incorporating temporal context. We adopt 3D ConvNets [3, 59] as used by [13] to incorporate larger temporal context from neighboring frames. We found that 3D ConvNets consistently outperform alternative approaches such as channel-wise stacking of frames at the input layer or average pooling at the output layer. The output feature map from 3D ConvNets has an extra time dimension, which is inconsistent with the 2D bounding box proposals generated by RPN. We address this issue by *flattening* the 3D feature map with a $\mathtt{t} \times 1 \times 1$ temporal convolution, where $\mathtt{t}$ is the size of the time dimension. The flattened 2D feature map can then be provided to a standard differentiable ROIPooling operation [21, 19] to produce cropped actor features. The cropped actor features are then inflated back to 3D, to allow reusing the pre-trained 3D ConvNets weights for classification. Empirically, we find that the flattening approach gives on par or better accuracy than keeping the temporal dimension with 3D ROIPooling.

Architecture details. For accurate actor bounding-box locations, we follow [13] and use a 2D ResNet-50 model [16] trained on key frames with action bounding-box annotations. For action classification, we use gated separable 3D network (S3D-G) [59]. Compared with I3D [3] used in [13], S3D-G replaces full 3D convolutions with separable spatial and temporal convolutions, and employs spatio-temporal feature gating layers. Overall, S3D-G is faster, has fewer parameters, provides higher accuracy compared to other 3D ConvNet models, and has a flexible design which makes it ideal for the large-scale action detection setup.

Following the recommendation from [59], we use the *top-heavy* configuration and use 2D convolutions without gating until the `Mixed_4b` block (we follow the same naming conventions as the Inception networks [52]), and switch to separable 3D convolutions with gating onwards. To combine RGB and optical flow input modalities, we use early fusion at the `Mixed_4f` block instead of late fusion at the logits layer. With these changes, we observed a 1.8× speed-up in our action detection model without loosing performance. We use the features from the fused `Mixed_4f` ($\mathtt{t} \times \mathtt{h} \times \mathtt{w} \times \mathtt{c}$) block for action classification. These features have a spatial output stride of 16 pixels and a temporal output stride of 4 frames. Regions in `Mixed_4f` corresponding to actor RPN proposals are temporally flattened and used as the input for the action classification network. We will refer to the $\mathtt{h} \times \mathtt{w} \times \mathtt{c}$ feature map as $\mathcal{F}$ going forward. For each RPN proposal generated by a potential actor ($b_i = (x_1^i, y_1^i, x_2^i, y_2^i)$)), we crop and resize the feature within b_i from $\mathcal{F}$ using ROIPooling to obtain a fixed-length representation $\mathcal{F}(b_i)$ of size $7 \times 7 \times 832$. This feature representation is used by the action classifier, which consists of `Mixed_5b` and `Mixed_5c` blocks (that output $7 \times 7 \times 1024$ feature), and an average pooling layer which outputs $1 \times 1 \times 1024$ feature. This feature is then used to learn a linear classifier for actions and a regressor for bounding-box offsets. We refer to the action detection model described above as our **Base-Model** throughout this paper. As shown in the experiments, the Base-Model by itself obtains state-of-the-art performance for action detection on the datasets explored in this work.

3.2 Actor-centric relations for action detection

A key component missing in the Base-Model is reasoning about relations outside the cropped regions. It is important to model such relations, as actions are in many cases defined by them (see Figure 1). Here, we propose to extract actor-centric relation features and to input them to the action classification network. Our approach performs relation reasoning given only action annotations, and automatically retrieves the regions that are most related to the action. Thus, we refer to our approach as an actor-centric relation network (ACRN).

Actor-centric relations. Given an input example I with a set of actors $\mathcal{A} = \{A_1, A_2, ..., A_M\}$ and objects $\mathcal{O} = \{O_1, O_2, ..., O_N\}$, we define a pair-wise relation feature between actor A_i and object O_j as $g_\theta(a_i, o_j)$, where $g_\theta(\cdot)$ is a feature extractor function parameterized by θ, and a_i and o_j are the feature representations of actor A_i and object O_j respectively.

The actor-centric relation feature for actor A_i can be computed by

$$\text{ACR}(A_i) = f_\phi\left(\left\{g_\theta(a_i, o_j) : O_j \in \mathcal{O}\right\}\right), \tag{1}$$

where $f_\phi(\cdot)$ is a function that aggregates features from all pair-wise relations, parameterized by ϕ.

To use actor-centric relations for action detection, we need to define actors, objects and their relations. We treat each actor proposal generated by RPN as one actor. However, objects and their relations are not explicitly defined or annotated for the action detection task. We adopt a workaround that treats each individual feature cell in the convolutional feature map $\mathcal{F}$ as an *object* O_i, which naturally gives the object representation o_i. This simplification avoids the need of generating object proposals, and has been shown to be effective for video classification [56] and question answering [44] tasks. However, as we show in the experiment section, directly applying this simplification does not improve the performance of the Base-Model. We compute ACR(A_i) with neural networks, this allows the module to be end-to-end trained with the Base-Model. In particular, both g_θ and f_ϕ can be implemented with standard convolutions and pooling operations.

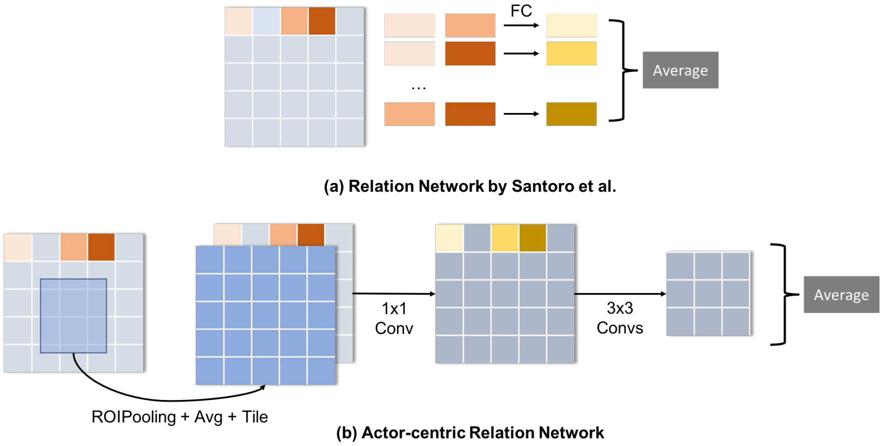
Action detection with actor-centric relation network (ACRN). We now discuss how to incorporate ACRN into our action detection framework. Given N frames from a video $(\mathcal{V})$, we first extract the fused feature map $\mathcal{F}_v$ of size $\mathtt{h} \times \mathtt{w} \times \mathtt{c}$ and a set of bounding-boxes generated by the actor RPN $(\mathcal{B} = (b_i, ..., b_R))$. For each box b_i, we follow the procedure described in Section 3.1 to get an actor feature f_i^a of size $1 \times 1 \times 1024$. Note that this is the same feature used by the Base-Model for action classification.

To extract pair-wise relation features, we follow the relation network module used by Santoro et al. [44] and implement g_θ as a single fully-connected layer. The inputs to g_θ are set as the concatenation of features from one actor proposal and one object location, along with their locations:

$$a_i = \left[f_i^a; b_i\right], \text{ and } o_{j,k} = \left[\mathcal{F}_v(j,k); l\,(j,k)\right], \tag{2}$$

where $\mathcal{F}_v(j,k)$ is the $1 \times 1 \times 832$ feature extracted at feature location (j,k), and $l = (j/H, k/W)$.

In practice, we can efficiently compute $g_\theta(a_i, o_{j,k})$ for all (j,k) locations using convolution operations. The actor appearance feature f_i^a is duplicated to $h \times w \times 1024$ feature and concatenated with $\mathcal{F}$ channel-wise, along with the box and



(a) Relation Network by Santoro et al.

(b) Actor-centric Relation Network

Fig. 3: Comparison of our proposed ACRN (**b**) with the relation network used by Santoro et al. [44] (**a**). We compute relation feature maps by duplicating actor features and applying a 1×1 convolution. A set of 3×3 convolutions are then applied on relation feature map to accumulate information from neighboring relations.

location embeddings. Next, g_θ is computed using a 1×1 convolution layer, which outputs $\mathcal{F}^\theta(a_i)$ of size $\mathtt{h} \times \mathtt{w} \times 832$. These operations are illustrated in Figure 3 (b). Since a_i and $o_{j,k}$ come from different layers with varying feature amplitude, we follow the design decisions from [1, 29, 46] to normalize and scale the features when combining them as inputs to the relational reasoning modules.

After $\mathcal{F}^\theta(a_i)$ is computed, the model needs to aggregate all g_θ with f_ϕ. One option is to directly apply average pooling as in [44], which works for synthetic data with relative simple scene [23]. However, for action recognition the relevant relational information could be very sparse, and averaging will dilute such information. Moreover, since the relation features are computed locally, information about bigger objects could be lost after average pooling. Instead, we propose to apply convolution operations on $\mathcal{F}^\theta(a_i)$ before average pooling, which allows relational information to accumulate over neighboring locations. In practice, we use `Mixed_5b` and `Mixed_5c` blocks of the S3D-G network and an average pooling layer (similar to the action classifier network) to output a $1 \times 1 \times 1024$ feature (f_i^{RN}). Finally, we concatenate actor feature f_i^a and spatio-temporal relation feature f_i^{RN} to get a $1 \times 1 \times 2048$ representation which is used for action classification and bounding-box regression for a given actor box b_i (see Figure 2). The same process is repeated for all proposed regions $\mathcal{B}$.

To handle objects of various scales, we further extend the ACRN module to allow feature maps other than $\mathcal{F}$ to be used, each of which can be resized to different scales. For example, smaller objects might be better represented in the earlier layers of the network at a larger scale feature map, while bigger objects might be better represented in the higher layers at a smaller scale. In Section 4.2, we study the impact of using different layers and their combination for relation reasoning.

4 Experiments

In this section, we perform design and ablation analysis of our method and visualize what relations are captured by the actor-centric relation network. Finally, we evaluate it on the task of spatio-temporal action localization and demonstrate consistent and significant gain across multiple benchmarks.

4.1 Experimental Setup

Datasets and metrics. We report results on the JHMDB [22] and AVA [13] action detection benchmarks. JHMDB [22] consists of 928 temporally trimmed clips with 21 action classes and has three training/validation splits. Unless noted otherwise, we follow the standard setup and report results by averaging over all three splits. We report the frame-level and video-level mean average precision (frame-AP and video-AP) with an intersection-over-union (IOU) threshold of 0.5. For video-AP, we link per-frame action detection results into tubes using the algorithm from [13]. We use the AVA version 2.1 benchmark [13]. It consists of 211k training and 57k validation examples labeled at 1FPS over 80 action classes. We follow their baseline setup and report results on 60 action classes which have at least 25 validation examples per class. We report frame-AP for AVA.

Implementation details. For the Base-Model, we use the ResNet-50 [16] RGB model for actor localization and the S3D-G [59] two-stream model for action classification. The detailed architecture is described in Section 3.1. Optical flow for the two-stream network is extracted using FlowNet2 [20]. As is standard practice, the ResNet-50 model is pre-trained on ImageNet and S3D-G RGB+Flow streams are pre-trained on Kinetics. The classification head (`Mixed_5b`, `Mixed_5c`) are initialized from RGB stream pre-trained on Kinetics for both RN and actor classification, but they are updated separately (weights are not shared). The whole pipeline (actor localization, actor-centric RN, and action classification) is trained jointly end-to-end.

We train the model for 200K and 1.2M steps for JHMDB and AVA respectively, and use start asynchronous SGD with a batch-size of 1 per GPU (11 GPUs in total), mini-batch size of 256 for actor RPN and 64 for action classifier (following [13]). We warm-start the learning rate from 0.00001 to 0.001 in 40K steps using linear annealing and then use cosine learning rate decay [30]. To stabilize training, the batch-norm updates are disabled during training and we apply a gradient multiplier of 0.01 to gradients from RN to the feature map.

Model	frame-AP
Base-Model	75.2
Resize+Concat [1, 29, 46]	74.8
Santoro et al. [44]	75.1
ACRN	**77.6**

(a) Relation reasoning modules.

Frames	Base-Model	ACRN
1	52.6	54.0
5	66.1	69.8
10	70.6	74.9
20	75.2	77.6

(b) Temporal context.

Feature	`Conv1a`	`Conv2c`	`Mixed_3b`	`Mixed_4b`	`Mixed_4f`	`Conv2c`, `Mixed_3c`, `Mixed_4f`
Scale 0.5	76.4	**77.2**	76.2	76.2	77.1	**77.9**
Scale 1.0	76.6	**77.9**	76.4	76.6	77.6	77.5

(c) Feature layers and scales.

Table 1: Frame-AP evaluating the impact of different parameters of ACRN on JHMDB dataset (3 splits).

4.2 Design and ablation analysis

We perform a number of ablation experiments to better understand the properties of our actor-centric relation network and its impact on the action detection performance. The results are shown in Table 1 and Table 2.

Importance of relation reasoning modules. Table 1a compare the performance between the Base-Model which only uses actor features, and three different relation reasoning modules which take global feature maps as additional inputs. We L2-normalize the actor appearance feature f_i^a and scene context feature `Mixed_4f`, concatenate the features together, scale the L2-norm of the concatenated feature back to the L2-norm of f_i^a, and use a 1×1 convolution layer to reduce the number of channels to be same as f_i^a. We study the following relational reasoning modules:

Resize+Concat [1, 29, 46]: resize the global feature map and actor feature map at `Mixed_4f` to have same size and directly concatenate channel-wise. The concatenated feature maps are fed into the classification head (`Mixed_5b-5c`).

Santoro et al. [44]: global and actor feature maps are used to compute g_θ, which are then averaged to compute f_ϕ with one fully-connected layer.

ACRN: global and actor feature maps are used to compute relation feature maps by ACRN, which are fed into the classification head.

Table 1a shows performance comparisons in frame-AP on JHMDB. Our proposed ACRN imrpoves by **2.4** over the Base-Model. However, Resize+Concat and Santoro et al. fail to outperform the baseline despite having access to global feature maps. The gap highlights the importance of designing an appropriate relation reasoning module.

Impact of temporal context. Table 1b studies the impact of temporal context on the Base-Model and the proposed ACRN framework by varying the number input frames, and show the results in . As observed by [13], using more input

Action	Mixed_4f	Conv2c	Gap	Action	Mixed_4f	Conv2c	Gap
answer phone	56.0	50.3	5.7	drive	15.3	19.5	-4.2
jump	6.8	4.0	2.8	fight	36.2	40.4	-4.2
swim	35.1	33.2	1.9	kiss	15.6	19.4	-3.8
read	10.3	8.6	1.7	play instrument	7.1	10.9	-3.8
dance	32.7	31.6	1.1	touch	24.2	26.7	-2.5

Table 2: AVA actions with biggest performance gaps when different features are used by ACRN.

frames generally helps our models. ACRN consistently improves over the Base-Model across all temporal lengths.

Comparison of feature layers and scales. ACRN can take feature maps from the different layers of a ConvNet as inputs. Each feature map can be resized to have different scales. Choices of feature layer and scale may have pros and cons: intuitively, features from higher layers (e.g. Mixed_4f) encode more semantic information, but have lower resolution; features from lower layers (e.g. Conv2c) have higher resolution but are less semantically meaningful. Similarly, feature map with larger scale potentially helps to identify interactions involving smaller objects, but also increases the number of relations to aggregate.

In Table 1c, we report frame-mAP on the JHMDB dataset by varying the feature layers and scales of the global feature map. We observe that Conv2c is the best performing single feature, followed by Mixed_4f. The performance is relatively stable for different scales. We note that combining features from multiple layers not necessarily results in better overall performance. In Table 2, we list the AVA categories with highest performance gap when using Mixed_4f and Conv2c. We can see that the two feature layers are clearly complimentary for many actions. In the following experiments, we report ACRN results based on the best single feature layer and scale.

4.3 Comparison with the state of the art

We compare our best models with the state-of-the-art methods on JHMDB and AVA. For the state-of-the-art methods, we use the same experimental setup and quote the results as reported by the authors. We fix the number of input frames to 20 for I3D, Base-Model and ACRN.

As shown in Table 3, our Base-Model already outperforms all previous methods, and the proposed ACRN algorithm further achieves a gain over this Base-Model. We also look into the per-class performance breakdown: on the JHMDB dataset, ACRN outperforms the Base-Model significantly for catch (12%), jump (6%), shoot gun (5%) and wave (10%). The gain is smaller when the performance of the Base-Model is almost saturated (e.g. golf, pullup and pour). The Base-Model performs only slightly better on pick, throw and run. When visualizing the relation heatmaps, we can see that ACRN has difficulty attending to the right relations for these actions.

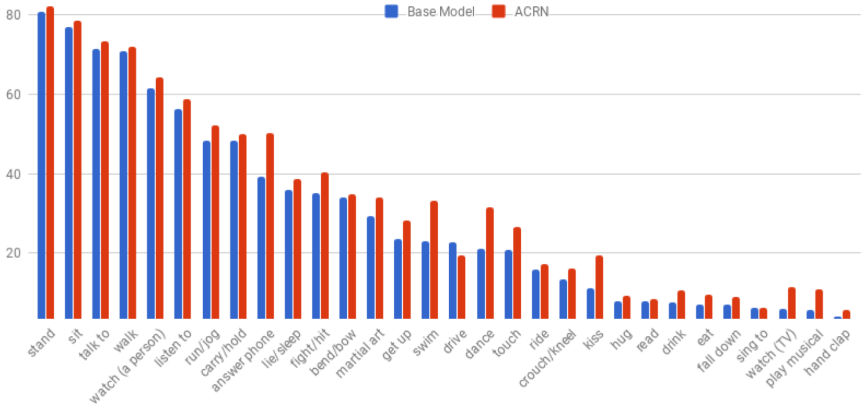


Fig. 4: Per-category frame-AP comparison between the Base-Model and ACRN on AVA. Labels are sorted by descending Base-Model performance, only the top 30 categories are shown.

On the AVA dataset, the per-class performance breakdown for the 30 highest performing categories can be found in Figure 4. We can discover that the biggest gains are achieved for answer phone (11%), fight (5%), swim (10%), dance (10%), touch (6%), kiss (8%) and play musical instruments (5%), most of which involve human-human or human-object interactions.

Model	frame-AP	video-AP
Peng et al. [37]	58.5	73.1
ACT [24]	65.7	73.7
I3D [13]	73.3	78.6
Base-Model	75.2	78.8
ACRN	**77.9**	**80.1**

(a) JHMDB (3 splits)

Model	frame-AP
Single frame [13]	14.2
I3D [13]	15.1
Base-Model	15.5
ACRN	**17.4**

(b) AVA (version 2.1)

Table 3: Comparison with state of the art on (a) the JHMDB dataset and (b) AVA . For JHMDB, we report average precision over 3 splits.

4.4 Qualitative Results

To qualitatively verify what relations are learned by ACRN, we apply the class activation map (CAM) [63] method to visualize the per-category relation heatmaps based on ACRN outputs. We modify the inference network by removing the average pooling operation after the ACRN branch, and apply the final action classifier as 1 × 1 convolutions on the relation feature map. This allows us to

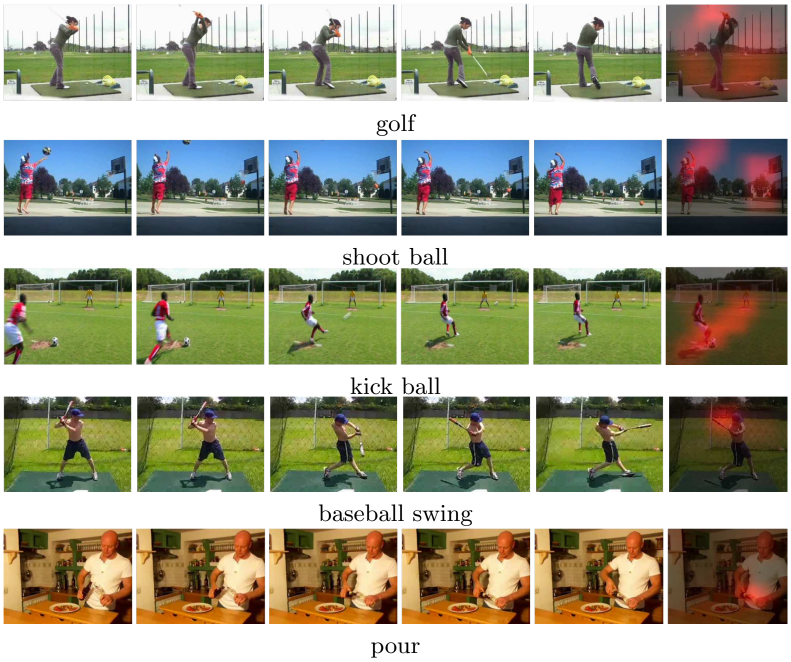


Fig. 5: Visualization of relation heatmap on JHMDB dataset.

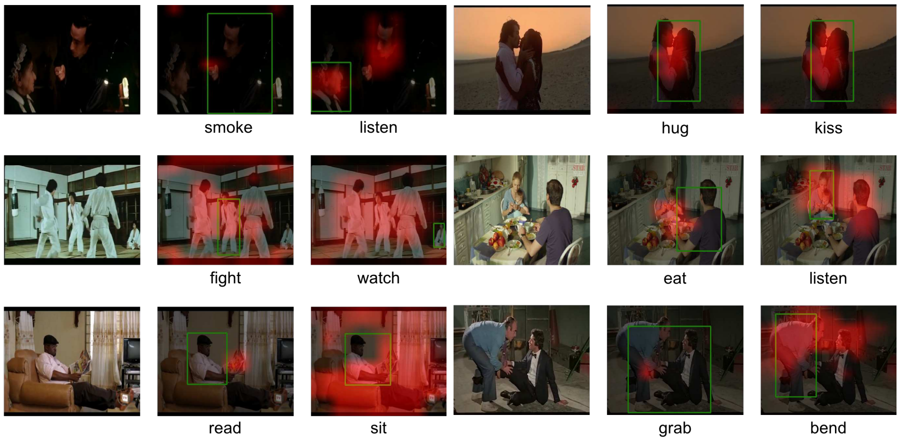


Fig. 6: Visualization of relation heatmap on AVA. The actor corresponding to the heatmap is marked in green, and its action is shown below the example. Notice how the heatmap varies depending on the action categories.

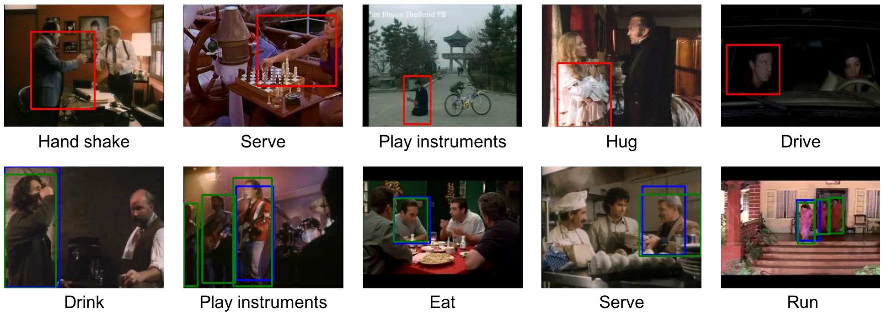


Fig. 7: (*Top row*) False alarm detections from the Base-Model (red boxes) that are removed by ACRN. (*Bottom row*) Miss detections (green) of the Base-Model captured by ACRN (blue).

generate spatially localized per-category activations, which illustrates the relations important to identify a certain action. Note that the spatial heatmaps also encode temporal information, as the input features are flattened from 3D to 2D.

Figure 5 and Figure 6 show the visualizations of the top-1 and top-2 highest scoring detections on JHMDB and AVA respectively. We render the bounding boxes and their associated relation heatmap in green and red respectively. We can see that ACRN is able to capture spatio-temporal relations beyond the actor bounding box, and its output depends on actor and action. Finally, Figure 7 illustrates examples for which the false alarms of the Base-Model are removed by ACRN (top row) and the missing detections are captured by ACRN (bottom row).

5 Conclusion

This paper presents a novel approach to automatically determine relevant spatio-temporal elements characterizing human actions in video. Experimental results for spatio-temporal action localization demonstrate a clear gain and visualizations show that the mined elements are indeed relevant. Future work includes a description of an actor by more than one feature, i.e., a number of features representing different human body parts. This will allow to model relations not only with an actor, but also the relations to relevant human parts. Another line of work could be to look at higher-order spatio-temporal relations.

Acknowledgement: We thank Chunhui Gu, David Ross and Jitendra Malik for discussion and comments.

References

1. Bell, S., Lawrence Zitnick, C., Bala, K., Girshick, R.: Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In: CVPR (2016)
2. Buch, S., Escorcia, V., Shen, C., Ghanem, B., Niebles, J.C.: SST: Single-stream temporal action proposals. In: CVPR (2017)
3. Carreira, J., Zisserman, A.: Quo vadis, action recognition? A new model and the Kinetics dataset. In: CVPR (2017)
4. Chao, Y.W., Liu, Y., Liu, X., Zeng, H., Deng, J.: Learning to detect human-object interactions. In: WACV (2018)
5. Dai, X., Singh, B., Zhang, G., Davis, L.S., Chen, Y.Q.: Temporal context network for activity localization in videos. In: ICCV (2017)
6. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: ICCV (2015)
7. Donahue, J., Hendricks, L.A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: CVPR (2015)
8. Girdhar, R., Ramanan, D.: Attentional pooling for action recognition. In: NIPS (2017)
9. Girshick, R.: Fast R-CNN. In: ICCV (2015)
10. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR (2014)
11. Gkioxari, G., Malik, J.: Finding action tubes. In: CVPR (2015)
12. Gkioxari, G., Girshick, R., Dollár, P., He, K.: Detecting and recognizing human-object intaractions. In: CVPR (2018)
13. Gu, C., Sun, C., Ross, D.A., Vondrick, C., Pantofaru, C., Li, Y., Vijayanarasimhan, S., Toderici, G., Ricco, S., Sukthankar, R., Schmid, C., Malik, J.: AVA: A video dataset of spatio-temporally localized atomic visual actions. In: CVPR (2018)
14. Gupta, S., Malik, J.: Visual semantic role labeling. arXiv:1505.04474 (2015)
15. Gupta, S., Hariharan, B., Malik, J.: Exploring person context and local scene context for object detection. arXiv:1511.08177 (2015)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
17. Heitz, G., Koller, D.: Learning spatial context: Using stuff to find things. In: ECCV (2008)
18. Hou, R., Chen, C., Shah, M.: Tube convolutional neural network (T-CNN) for action detection in videos. In: ICCV (2017)
19. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K.: Speed/accuracy trade-offs for modern convolutional object detectors. In: CVPR (2017)
20. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: FlowNet 2.0: Evolution of optical flow estimation with deep networks. In: CVPR (2017)
21. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: NIPS (2015)
22. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M.: Towards understanding action recognition. In: ICCV (2013)
23. Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C.L., Girshick, R.B.: CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In: CVPR (2017)

24. Kalogeiton, V., Weinzaepfel, P., Ferrari, V., Schmid, C.: Action tubelet detector for spatio-temporal action localization. In: ICCV (2017)
25. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR (2014)
26. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A.: The Kinetics human action video dataset. arXiv:1705.06950 (2017)
27. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
28. Liu, W., Anguelov, D., Erhan, D., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single shot MultiBox detector. In: ECCV (2016)
29. Liu, W., Rabinovich, A., Berg, A.C.: Parsenet: Looking wider to see better. arXiv:1506.04579 (2015)
30. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with restarts. In: ICLR (2017)
31. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. In: ECCV (2016)
32. Marszalek, M., Laptev, I., Schmid, C.: Actions in context. In: CVPR (2009)
33. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: CVPR (2014)
34. Ng, J.Y., Hausknecht, M.J., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: Deep networks for video classification. In: CVPR (2015)
35. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. IJCV (2001)
36. Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: CVPR (2016)
37. Peng, X., Schmid, C.: Multi-region two-stream R-CNN for action detection. In: ECCV (2016)
38. Peyre, J., Laptev, I., Schmid, C., Sivic, J.: Weakly-supervised learning of visual relations. In: ICCV (2017)
39. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in context. In: ICCV (2007)
40. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS (2015)
41. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV (2015)
42. Saha, S., G.Sing, Cuzzolin, F.: AMTnet: Action-micro-tube regression by end-to-end trainable deep architecture. In: ICCV (2017)
43. Saha, S., Singh, G., Sapienza, M., Torr, P., Cuzzolin, F.: Deep learning for detecting multiple space-time action tubes in videos. In: BMVC (2016)
44. Santoro, A., Raposo, D., Barrett, D.G.T., Malinowski, M., Pascanu, R., Battaglia, P., Lillicrap, T.P.: A simple neural network module for relational reasoning. In: NIPS (2017)
45. Sharma, S., Kiros, R., Salakhutdinov, R.: Action recognition using visual attention. arXiv:1511.04119 (2015)
46. Shrivastava, A., Gupta, A.: Contextual priming and feedback for faster R-CNN. In: ECCV (2016)

47. Shrivastava, A., Sukthankar, R., Malik, J., Gupta, A.: Beyond skip connections: Top-down modulation for object detection. arXiv:1612.06851 (2016)
48. Sigurdsson, G., Varol, G., Wang, X., Farhadi, A., Laptev, I., Gupta, A.: Hollywood in homes: Crowdsourcing data collection for activity understanding. In: ECCV (2016)
49. Sigurdsson, G.A., Varol, G., Wang, X., Farhadi, A., Laptev, I., Gupta, A.: Hollywood in homes: Crowdsourcing data collection for activity understanding. In: ECCV (2016)
50. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NIPS (2014)
51. Singh, G., Saha, S., Sapienza, M., Torr, P., Cuzzolin, F.: Online real-time multiple spatiotemporal action localisation and prediction. In: ICCV (2017)
52. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR (2015)
53. Torralba, A., Murphy, K.P., Freeman, W.T., Rubin, M.A.: Context-based vision system for place and object recognition. In: ICCV (2003)
54. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: ICCV (2015)
55. Varol, G., Laptev, I., Schmid, C.: Long-term temporal convolutions for action recognition. IEEE PAMI (2017)
56. Wang, X., Girshick, R.B., Gupta, A., He, K.: Non-local neural networks. In: CVPR (2018)
57. Weinzaepfel, P., Harchaoui, Z., Schmid, C.: Learning to track for spatio-temporal action localization. In: ICCV (2015)
58. Weinzaepfel, P., Martin, X., Schmid, C.: Towards weakly-supervised action localization. arXiv:1605.05197 (2016)
59. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning for video understanding. In: ECCV (2018)
60. Xu, H., Das, A., Saenko, K.: R-C3D: Region convolutional 3D network for temporal activity detection. In: ICCV (2017)
61. Yao, B., Fei-Fei, L.: Modeling mutual context of object and human pose in human-object interaction activities. In: CVPR (2010)
62. Zhao, H., Yan, Z., Wang, H., Torresani, L., Torralba, A.: SLAC: A sparsely labeled dataset for action classification and localization. arXiv:1712.09374 (2017)
63. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: CVPR (2016)
64. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: NIPS (2014)
65. Zolfaghari, M., Oliveira, G., Sedaghat, N., Brox, T.: Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection. In: ICCV (2017)