Object Level Visual Reasoning in Videos

Fabien Baradel[1,2], Natalia Neverova[3], Christian Wolf[1,4], Julien Mille[5,6], Greg Mori[7]

[1]LIRIS [2]INSA-Lyon [3]Facebook AI Research [4]INRIA [5]INSA CVL [6]LI Tours [7]SFU

Abstract. Human activity recognition is typically addressed by training models to detect key concepts like global and local motion, features related to object classes present in the scene, as well as features related to the global context. The next open challenges in activity recognition require a level of understanding that pushes beyond this, requiring fine distinctions and a detailed comprehension of the interactions between actors and objects in a scene. We propose a model capable of learning to reason about semantically meaningful spatio-temporal interactions in videos. Key to our approach is the choice of performing this reasoning on an object level through the integration of state of the art object instance segmentation networks. This allows the model to learn detailed spatial interactions that exist at a semantic, object-interaction relevant level. We evaluated our method on three standard datasets: the Twenty-BN Something-Something dataset, the VLOG dataset and the EPIC Kitchens dataset, and achieve state of the art results on both. Finally, we also show visualizations of the interactions learned by the model, which illustrate object classes and their interactions corresponding to different activity classes.

1 Introduction

Video understanding is a diverse field in computer vision. At one end of the spectrum is video classification, with a goal of recognition of high level concepts in diverse internet video, exemplified by the YouTube-8M dataset [1]. On the other is close range 3D human activity recognition based on specialized RGB-D sensors and laboratory data collection [2]. Powerful models exist for analyzing internet videos. As in object detection, large-scale datasets have been introduced [3–8], which allow the training of high-capacity models from large amounts of data. These models enable the detection of key concepts present in videos, capturing concepts such as global and local motion, the presence of various object categories in a video, and global scene-level information. Impressive performance in recognizing high-level, often abstract concepts in diverse internet videos has been achieved.

However, recent attention has been directed toward a more thorough understanding of human-focused activity in diverse internet videos. These efforts range from atomic human actions [8] to fine-grained object interactions [5] to everyday,

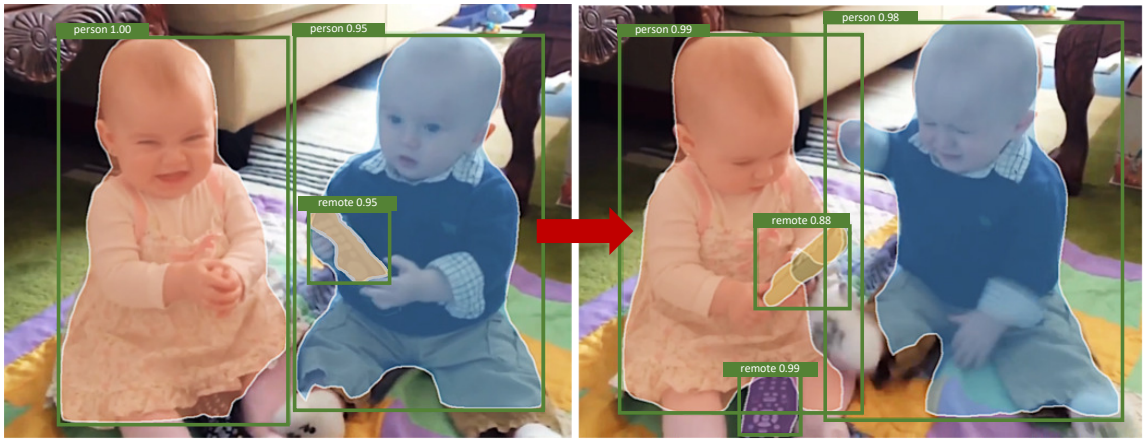

Fig. 1. Reasoning on what happened in a video requires higher-level reasoning, which our method performs on object level through an integrated mask predictor.

commonly occurring human-object interactions [6]. This returns us to a human-centric viewpoint of activity recognition where it is not only the presence of certain objects / scenes that dictate the activity present, but the manner, order, and effects of human interaction with these scene elements that are necessary for understanding. In a sense, this is akin to the problems in current 3D human activity recognition datasets [2], but requires the more challenging reasoning and understanding of diverse environments common to internet video collections.

Humans are able to infer what happened in a scene given a few sample images only. In particular, they can infer complex activities happening between pairs of frames. This faculty is called *reasoning* and is a key component of human intelligence. As an example we can consider the pair of images in Figure 1, which shows a complex situation involving articulated objects (babies), the change of possession of an object and a change in emotional expression. For humans it is straightforward to draw a conclusion on what happened (the remote control was taken away and created unhappiness). Humans have this extraordinary ability of performing visual reasoning on very complicated tasks while it is currently very hard for contemporary computer vision algorithms [9, 10].

The ability to perform *visual reasoning* in computer vision algorithms is still an open problem. Attempts have been made for learning interactions between different entities in images with promising results on Visual Question Answering problems with solutions ranging from prior-less data normalization [11] to structuring networks modeling relationships [12, 13] up to complex attention based mechanisms [14]. However, studies have shown that many visual reasoning methods obtain their results based by exploiting dataset bias and do not perform real reasoning [15].

We extend these efforts to *object level reasoning in videos*. Since a video is a temporal sequence we leverage the time as an explicit causal signal to identify causal object relations. Our approach is related to the concept of the *"arrow of the time"* [16] involving the "one-way direction" or "asymmetry" of time. Causal event occurs before the event it affects ($A \rightarrow B$). In Figure 1 the remote control was taken before the unhappiness of the baby on the right side. For a

video classification problem, we want to identify a causal event A happening in a video that affects its label B. But instead of identifying this causal event directly from pixels we want to identify it from an object level perspective. We believe that such an approach would be able to learn causal signals.

Following this hypothesis we propose to make a bridge between object detection and activity recognition. Object detection allows us to extract low-level information from a scene with all the present object instances and their semantic meanings. However, detailed activity understanding requires reasoning over these semantic structures, determining which objects were involved in interactions, of what nature, and what were the results of these. To compound problems, the semantic structure of a scene may change during a video (e.g. a new object can appear, a person may make a move from one point to another one of the scene).

We propose an **Object Relation Network** (ORN), a neural network module for reasoning between detected semantic object instances through space and time. The ORN has potential to address these issues and conduct relational reasoning over object interactions for the purpose of activity recognition. A set of object detection masks ranging over different object categories and temporal occurrences is input to the ORN. The ORN is able to infer pairwise relationships between objects detected at varying different moments in time.

2 Related work

Action Recognition — Action recognition has a long history in computer vision. Pre-deep learning approaches focused on handcrafted spatio-temporal features including space-time interest points like SIFT-3D, HOG3D, IDT and aggregated them using bag-of-words techniques. Some hand-crafted representations, like dense trajectories [17], still give competitive performance and are frequently combined with deep learning.

In the recent past, work on video understanding has shifted to deep learning. Early attempts have been made to adapt 2D convolutional networks to videos through temporal pooling and 3D convolutions, which were used early on [18, 19]. 3D convolutions are now widely adopted for activity recognition with further recent advances through the introduction of feature transfer from image classification models pre-trained on ImageNet/ILSVRC [20] to activity recognition by inflating 2D convolutional kernels through 3D kernels after pre-training [3]. The downside of 3D kernels is their computational complexity and the large number of learnable parameters, leading to the introduction of 2.5D kernels, i.e. separable filters in the form of a 2D spatial kernel followed by a temporal kernel [21]. An alternative to temporal convolutions are Recurrent Neural Networks (RNNs) in their various gated forms (GRUs, LSTMs) [22, 23].

Karpathy et al. [24] presented a wide study on different ways of connecting information in spatial and temporal dimensions through convolutions and pooling. They also showed that there was a small margin between classifying individual frames and classifying videos with more sophisticated temporal ag-

gregation. However, these results were obtained on very general datasets with coarse activity classes.

Simoyan et al. [25] proposed a widely adopted two-stream architecture for action recognition which extracts two different types of features using two different streams, one processing raw RGB input and a second stream processing pre-computed optical flow images. The method outperformed the state of the art, but relies on rather small scale optical flow computations.

In slightly narrower settings, prior information on the video content can allow more fine-grained models. Articulated pose is widely used in cases where humans are guaranteed to be present [2]. Pose estimation and activity recognition as a joint (multi-task) problem has recently shown to improve either tasks [26]. Somewhat related to our work, Structural RNNs [27] perform activity recognition by integrating features from semantic objects and their relationships. However, they handle the temporal evolution of tracked objects in videos with a set of RNNs, each of which corresponds to cliques in a graph which models the spatio-temporal relationships between these objects. This graph is hand-crafted manually for each application, though related work provides learnable connections via gating functions [28].

Attention models are a way to structure deep networks in an often generic way. They are able to iteratively focus attention to specific parts in the data without requiring prior knowledge about part or object positions. In activity recognition, they have gained some traction in recent years, either as soft-attention on articulated pose (joints) [29], on feature map cells [30, 31], on time [32] or on parts in raw RGB input through differentiable crops [33].

When raw video data is globally fed into deep neural networks, they focus on extracting spatio-temporal features and perform aggregations. It has been shown that these techniques fail on challenging fine-grained datasets, which require learning long temporal dependencies and human-object interactions. A concentrated effort has been made to create large scale datasets to overcome these issues [5–8].

Relational Reasoning — Relational reasoning is a well studied field for many applications ranging from visual reasoning [12] to reasoning about physical systems [34]. Battaglia et al. [34] introduce a fully-differentiable network physics engine called Interaction Network (IN). IN learns to predict several physical systems such as gravitational systems, rigid body dynamics, and mass-spring systems. It shows impressive results; however, it learns from a virtual environment, which provides access to virtually unlimited training examples. Following the same perspective, Santoro et al. [12] introduced Relation Network (RN), a plug-in module for reasoning in deep networks. RN shows human-level performance in Visual Question Answering (VQA) by inferring pairwise "object" relations. However, in contrast to our work, the term "object" in [12] does not refer to semantically meaningful entities, but to discrete cells in feature maps. The number of interactions therefore grows with feature map resolutions, which makes it difficult to scale. Furthermore, a recent study [15] has shown that some

of these results are subject to dataset bias and do not generalize well to small changes in the settings of the dataset.

In the same line, recent work [35] has shown promising results on discovering objects and their interactions in an unsupervised manner using training examples from virtual environments. In [36], attention and relational modules are combined on a graph structure. From a different perspective, [11] show that relational reasoning can be learned in a data driven way without any prior using conditional batch normalization. They use a feature-wise affine transformation based on conditioning information and show that this operation is effective for visual reasoning. In an opposite approach, a strong structural prior is learned in the form of a complex attention mechanism: in [14], an external memory module combined with attention processes over input images and textural questions performs iterative reasoning for VQA.

While most of the discussed work has been designed for VQA and for predictions on physical systems and environments, extensions have been proposed for video understanding. Reasoning in videos on a mask or segmentation level has been attempted for video prediction [37], where the goal was to leverage semantic information to be able predict further into the future. Zhou et al [38] have recently shown state-of-the-art performance on challenging datasets by extending Relation Network to video classification. Their chosen entities are frames, on which thy employ RN to reason on a temporal level only through pairwise frame relations. The approach is promising, but restricted to temporal contextual information without an understanding on a local object level, which is provided by our approach.

Reasoning over sets of objects is somewhat related to reasoning from unstructured data points. A connection can be mode to PointNet [39], which was designed to learn from unordered sets of points as input, while respecting the permutation invariance of points in an efficient manner. PointNet shares many properties with DeepSet [40] which is a more general framework for extracting information from sets of objects. To some extent, our work is related to PointNet, as we handle unordered sets of objects in a permutation invariant way. However, we have an object relation viewpoint that directly reasons over relationships between these semantic entities.

3 Object-level Visual Reasoning in Space and Time

Our goal is to extract multiple types of cues from a video sequence: interactions between predicted objects and their semantic classes, as well as local and global motion in the scene. We formulate this objective as a neural architecture with two heads: an *activity head* and an *object head*. Figure 2 gives a functional overview of the model. Both heads share common features up to a certain layer shown in red in the figure. The *activity head*, shown in orange in the figure, is a CNN-based architecture employing convolutional layers, including spatio-temporal convolutions, able to extract global motion features. However, it is not able to extract information from an object level perspective. We leverage the

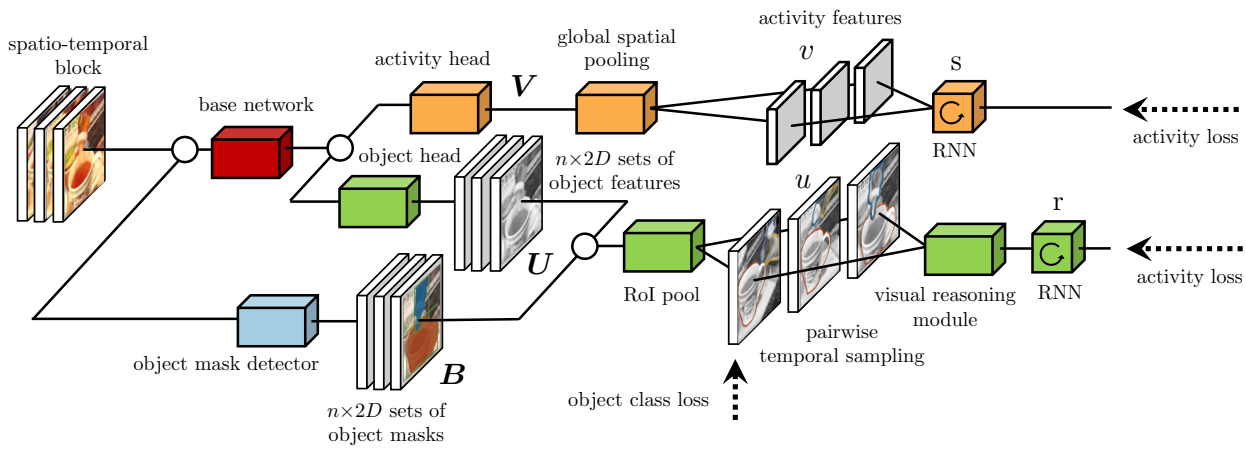

Fig. 2. A functional overview of the model. A global convolutional model extracts features and splits into two heads trained to predict, respectively activity classes and object classes. The latter are predicted by pooling over object instance masks, which are predicted by an additional convolutional model. The object instances are passed through a visual reasoning module.

object head to perform reasoning on the relationships between predicted object instances.

Our main contribution is a new structured module called **Object Relation Network** (ORN), which is able to perform spatio-temporal reasoning between detected object instances in the video. ORN is able to reason by modeling how objects move, appear and disappear and how they interact between two frames.

In this section, we will first describe our main contribution, the ORN network. We then provide details about object instance features, about the activity head, and finally about the final recognition task. In what follows, lowercase letters denote 1D vectors while uppercase letters are used for 2D and 3D matrices or higher order tensors. We assume that the input of our system is a video of T frames denoted by $\mathbf{X}_{1:T} = (\mathbf{X}_t)_{t=1}^{T}$ where $\mathbf{X}_t$ is the RGB image at timestep t. The goal is to learn a mapping from $\mathbf{X}_{1:T}$ to activity classes $\mathbf{y}$.

3.1 Object Relation Network

ORN (Object Relation Network) is a module for reasoning between semantic objects through space and time. It captures object moves, arrivals and interactions in an efficient manner. We suppose that for each frame t, we have a set of objects k with associated features $\mathbf{o}_t^k$. Objects and features are detected and computed by the object head described in Section 3.2.

Reasoning about activities in videos is inherently temporal, as activities follow the *arrow of time* [16], i.e. the causality of the time dimension imposes that past actions have consequences in the future but *not* vice-versa. We handle this by sampling by running a process over time t, and for each instant t, sampling a second frame t' with $t'<t$. Our network reasons on objects which interact between pairs of frames and their corresponding sets of objects $\mathbf{O}_{t'} = \left\{\mathbf{o}_{t'}^k\right\}_{k=1}^{K'}$

and $\mathbf{O}_t = \left\{\mathbf{o}_t^k\right\}_{k=1}^K$. The goal is to learn a general function defined on the set of all input objects from the combined set of both frames:

$$\mathbf{g}_t = g(\mathbf{o}_{t'}^1, \ldots, \mathbf{o}_{t'}^{K'}, \mathbf{o}_t^1, \ldots, \mathbf{o}_t^K). \tag{1}$$

The objects in this set are unordered, regardless of the frame they belong to. This task is related to a problem raised in the PointNet algorithm [39] discussed in Section 2. PointNet approximates a general function g over an item set $\mathcal{S} = \{x_1, x_2, \ldots, x_N\}$ as a symmetric function g' on transformed elements of the set

$$g(x_1, x_2, \ldots, x_N) \approx g'(h(x_1), h(x_2), \ldots, h(x_N)). \tag{2}$$

In [39], g' is a max pooling operation. The authors show that it allows universal approximation of continuous sets of functions given that the hidden representation (the output of the mapping $h(\cdot)$) is of sufficiently high dimension.

We argue that the approximation in (2) can be extended as follows:

$$g(x_1, x_2, \ldots, x_N) \approx f_\phi \left(\sum_{c \in \mathcal{C}} h_c \left(\cup_{i \in c}\, x_i \right) \right) \tag{3}$$

where $\mathcal{C}$ is the set of cliques of a graph defined over the item set $\mathcal{S}$, $\cup$ is the concatenation operator and we chose the sum operator as symmetric function. The input dimension of the non-linearity $h_c(\cdot)$ depends on the size of clique c but maps to a fixed output dimension H. In the case where $\mathcal{C}$ is composed of unary cliques only, form (3) decomposes like (2) with the exception of a different symmetry operator (sum instead of max pooling). Choosing different graphical structures through $\mathcal{S}$ will lead to different terms in the summation and allows modeling different types of interactions between items in the item set.

Note that the interactions between items in (3) are not exclusively modeled through $h_c(\cdot)$. Indeed, it is interesting to note, that the graphical decomposition provided by $\mathcal{C}$ leads to interactions which are different from the interactions the same decomposition would provide when used in a probabilistic graphical model, like for instance a Markov Random Field (MRF). In particular, a decomposition into unary terms only, as given in equation (2), does *not* lead to independence between items, whereas an MRF with unary terms only is equivalent to a distribution over independent random variables. This is a consequence of the global mapping $f_\phi(\cdot)$, which is defined on the sum over all direct interactions. Higher-order interactions between several items not directly modeled through a non-linearity $h_c(\cdot)$ can eventually be learned by the model through the joint output space of all $h_c(\cdot)$, provided that the dimensionality H of this space is high enough to incorporate all interactions. However, whereas the mapping $h_c(\cdot)$ provides a direct model of interactions between pairs of items, learning interactions between two items (j, k), which are not directly captured through a clique c and its corresponding $h_c(\cdot)$, requires learning a corresponding subspace in the common output space spanned by all $h_c(\cdot)$.

This leads to the question of how to define the trade-off between the complexity of the decomposition $\mathcal{C}$ and the output dimensionality H of the mapping

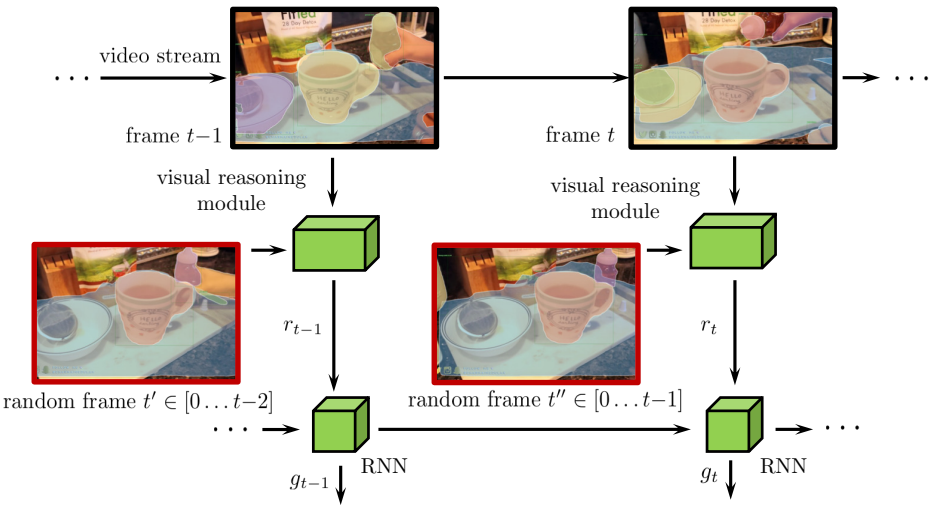

Fig. 3. ORN in the object head operating on detected instances of objects.

$h_c(\cdot)$, both of which will determine the complexity of the modeled interactions. Increasing the size of cliques in $\mathcal{C}$ will increase the input dimension (and therefore the capacity) of the mapping $h_c(\cdot)$ as well as the computational complexity of the sum operation.

Inspired by relational networks [12], we chose to directly model inter-frame interactions between pairs of objects (j, k) and leave modeling of higher-order interactions to the output space of the mappings h_θ and the global mapping f_ϕ:

$$\mathbf{g}_t = \sum_{j,k} h_\theta(\mathbf{o}^j_{t'}, \mathbf{o}^k_t) \tag{4}$$

In order to better directly model long-range interactions, we make the global mapping $f_\phi(\cdot, \cdot)$ recurrent, which leads to the following form:

$$\mathbf{r}_t = f_\phi(\mathbf{g}_t, \mathbf{r}_{t-1}) \tag{5}$$

where $\mathbf{r}_t$ represents the recurrent *object reasoning state* at time t and $\mathbf{g}_t$ is the global inter-frame interaction inferred at time t such as described in Equation 4. In practice, this is implemented as a GRU, but for simplicity we omitted the gates in equation (5). The pairwise mappings $h_\theta(\cdot, \cdot)$ are implemented as an MLP. Figure 3 provides a visual explanation of the object head's operating through time.

ORN differs from [12] in three main points:

Objects have a semantic definition — we model relationships with respect to semantically meaningful entities (object instances) instead of feature map cells which do not have a semantically meaningful spatial extent. We will show in the experimental section that this is a key difference.

Objects are selected from different frames — we infer object pairwise relations only between objects present in two different sets. This is a key design choice which allows our model to reason about changes in object relationships over time.

Long range reasoning — integration of the object relations over time is recurrent by using a RNN for $f_\phi(\cdot)$. Since reasoning from a full sequence cannot be done by inferring the relations between two frames, $f_\phi(\cdot)$ allows long range reasoning on sequences of variable length.

3.2 Object instance features

The object features $\mathbf{O}_t = \left\{\mathbf{o}_t^k\right\}_{k=1}^K$ for each frame t used for the ORN module described above are computed and collected from local regions predicted by a mask predictor. Independently for each frame $\mathbf{X}_t$ of the input data block, we predict object instances as binary masks $\mathbf{B}_t^k$ and associated object class predictions $\mathbf{c}_t^k$, a distribution over C classes. We use Mask-RCNN [41], which is able to detect objects in a frame using region proposal networks [42] and produces a high quality segmentation mask for each object instance.

The objective is to collect features for each object instance, which jointly describe its appearance, the change in its appearance over time, and its shape, i.e. the shape of the binary mask. In theory, appearance could also be described by pooling the feature representation learned by the mask predictor (Mask R-CNN). However, in practice we choose to pool features from the dedicated *object head* such as shown in Figure 2, which also include motion through the spatio-temporal convolutions shared with the activity head:

$$\mathbf{u}_t^k = \text{ROI-Pooling}(\mathbf{U}_t, \mathbf{B}_t^k) \tag{6}$$

where $\mathbf{U}_t$ is the feature map output by the *object head*, $\mathbf{u}_t^k$ is a D-dimensional vector of appearance and appearance change of object k.

Shape information from the binary mask $\mathbf{B}_t^k$ is extracted through the following mapping function:

$$\mathbf{b}_t^k = g_\phi(\mathbf{B}_t^k) \tag{7}$$

where $g_\phi(\cdot)$ is a MLP. Information about object k in image $\mathbf{X}_t$ is given by a concatenation of appearance, shape, and object class:

$$\mathbf{o}_t^k = [\ \mathbf{b}_t^k \quad \mathbf{u}_t^k \quad \mathbf{c}_t^k\ ] \tag{8}$$

3.3 Global Motion and Context

Current approaches in video understanding focus on modeling the video from a high-level perspective. By a stack of spatio-temporal convolution and pooling they focus on learning global scene context information. Effective activity recognition requires integration of both of these sources: global information about the entire video content in addition to relational reasoning for making fine distinctions regarding object interactions and properties.

In our method, local low-level reasoning is provided through object head and the ORN module such as described above in Section 3.1. We complement

this representation by high-level context information described by $\mathbf{V}_t$ which are feature outputs from the activity head (orange block in Figure 2).

We use spatial global average pooling over $\mathbf{V}_t$ to output T D-dimensional feature vectors denoted by $\mathbf{v}_t$, where $\mathbf{v}_t$ corresponds to the context information of the video at timestep t.

We model the dynamics of the context information through time by employing a RNN $f_\gamma(\cdot)$ given by:

$$\mathbf{s}_t = f_\gamma(\mathbf{v}_t, \mathbf{s}_{t-1}) \tag{9}$$

where $\mathbf{s}$ is the hidden state of $f_\gamma(\cdot)$ and gives cues about the evolution of the context though time.

3.4 Recognition

Given an input video sequence $\mathbf{X}_{1:T}$, the two different streams corresponding to the activity head and the object head result in the two representations $\mathbf{h}$ and $\mathbf{r}$, respectively where $\mathbf{h} = \sum_t \mathbf{h}_t$ and $\mathbf{r} = \sum_t \mathbf{r}_t$. Each representation is the hidden state of a respective GRU, which were described in the preceding subsections. Recall that $\mathbf{h}$ provides the global motion context while $\mathbf{r}$ provides the object reasoning state output by the ORN module. We perform independent linear classification for each representation:

$$\mathbf{y}^1 = \mathbf{W}\,\mathbf{h} \tag{10}$$

$$\mathbf{y}^2 = \mathbf{Z}\,\mathbf{r} \tag{11}$$

where $\mathbf{y}^1, \mathbf{y}^2$ correspond to the logits from the *activity head* and the *object head*, respectively, and $\mathbf{W}$ and $\mathbf{Z}$ are trainable weights (including biases). The final prediction is done by averaging logits $\mathbf{y}^1$ and $\mathbf{y}^2$ followed by softmax activation.

4 Network Architectures and feature dimensions

The input RGB images $\mathbf{X}_t$ are of size $\mathbf{R}^{3\times W\times H}$ where W and H correspond to the width and height and are of size 244 each. The object and activity heads (orange and green in Figure 2) are a joint convolutional neural network with Resnet50 architecture pre-trained on ImageNet/ILSVRC [20], with Conv1 and Conv5 blocks being inflated to 2.5D convolutions [21] (3D convolutions with a separable temporal dimension). This choice has been optimized on the validation set, as explained in Section 6 and shown in Table 5.

The last *conv5* layers have been split into two different heads (activity head and object head). The intermediate feature representations $\mathbf{U}_t$ and $\mathbf{V}_t$ are of dimensions $2048\times T\times 7\times 7$ and $2048\times T\times 14\times 14$, respectively. We provide a higher spatial resolution for the feature maps $\mathbf{U}_t$ of the object head to get more precise local descriptors. This can be done by changing the stride of the initial *conv5* layers from 2 to 1. Temporal convolutions have been configured to keep the same time temporal dimension through the network.

Global spatial pooling of activity features results in a 2048 dimensional feature vector fed into a GRU with 512 dimensional hidden state $\mathbf{s}_t$. ROI-Pooling of object features results in 2048 dimensional feature vectors $\mathbf{u}_t^k$. The encoder of the binary mask is a MLP with one hidden layer of size 100 and outputs a mask embedding $\mathbf{b}_t^k$ of dimension 100. The number of object classes is 80, which leads in total to a 2229 dimensional object feature vector $\mathbf{o}_t^k$.

The non-linearity $h_\theta(\cdot)$ is implemented as an MLP with 2 hidden layers each with 512 units and produces an 512 dimensional output space. $f_\phi(\cdot)$ is implemented as a GRU with a 256 dimension hidden state $\mathbf{r}_t$. We use ReLU as the activation function after each layer for each network.

5 Training

We train the model with three different losses:

$$\mathcal{L} = \mathcal{L}_1\Big(\frac{\hat{\mathbf{y}}^1 + \hat{\mathbf{y}}^2}{2}, \mathbf{y}\Big) + \sum_t \sum_k \mathcal{L}_2(\hat{\mathbf{c}}_t^k, \mathbf{c}_t^k). \tag{12}$$

where $\mathcal{L}_1$ and $\mathcal{L}_2$ are the cross-entropy loss.

The first term correspond to the supervised activity class losses comparing two different activity class predictions to the class ground truth: $\hat{\mathbf{y}}^1$ is the prediction of the activity head, whereas $\hat{\mathbf{y}}^2$ is the prediction of the object head, as given by equations (10) and (11), respectively.

The third term is a loss which pushes the features $\mathbf{U}$ of the object towards representations of the semantic object classes. The goal is to obtain features related to, both, motion (through the layers shared with the activity head), as well as as object classes. As ground-truth object classes are not available, we define the loss as the cross-entropy between the class label $\mathbf{c}_t^k$ predicted by the mask predictor and a dedicated linear class prediction $\hat{\mathbf{c}}_t^k$ based on features $\mathbf{u}_t^k$, which, as we recall, are RoI-pooled from $\mathbf{U}$:

$$\mathbf{c}_t^k = \mathbf{R}\ \mathbf{u}_t^k \tag{13}$$

where $\mathbf{R}$ trainable parameters (biases integrated) learned end-to-end together with the other parameters of the model.

We found that first training the object head only and then the full network was performing better. A ResNet50 network pretrained on ImageNet is modified by inflating some of its filters to 2.5 convolutions (3D convolutions with the time dimension separated), as described in Section 4; then by fine-tuning.

We train the model using the Adam optimizer [43] with an initial learning rate of 10^{-4} on $\sim$ 30 epochs and use early-stopping criterion on the validation set for hyper-parameter optimization. Training takes $\sim$50 minutes per epoch on 4 Titan XP GPUs with clips of 8 frames.

6 Experimental results

We evaluated the method on three standard datasets, which represent difficult fine-grained activity recognition tasks: the Something-Something dataset, the VLOG dataset and the recently released EPIC Kicthens dataset.

Something-Something (SS) — is a recent video classification dataset with 108,000 example videos and 157 classes [5]. It shows humans performing different actions with different objects, actions and objects being combined in different ways. Solving SS requires common sense reasoning and standard state-of-the-art methods in activity recognition tend to fail, which makes this dataset very challenging.

VLOG — has been recently released with 114,000 videos and 30 classes [6]. The underlying task is a multi-label binary classification of human-object interactions. Classes correspond to objects, and labels of a class are 1 if a person has touched a certain object during the video, otherwise they are 0. It has recently been shown, that state-of-the-art video based methods [3] are outperformed on VLOG by image based methods like ResNet-50 [44], although these video methods outperform image based ResNet-50 on large-scale video datasets like the Kinetics dataset [3]. This suggests a gap between traditional datasets like Kinetics and the fine-grained dataset VLOG, making it particularly difficult.

EPIC Kitchens (EPIC) — is an egocentric video dataset recently released containing 55 hours recording of daily activities [45]. This is the largest in first-person vision and the activities performed are non-scripted, which makes the dataset very challenging and close to real world data. The dataset is densely annotated and several tasks exist such as object detection, action recognition and action prediction. We focus on action recognition with 39'594 action segments in total and 125 actions classes (i.e verbs). Since the test set is not available yet we conducted our experiments on the training set (28'561 videos). We use the videos recorded by person 01 to person 25 for training (22'675 videos) and define the validation set as the remaining videos (5'886 videos).

For all datasets we rescale the input video resolution to 256×256. While training, we crop space-time blocks of 224×224 spatial resolution and L frames, with L=8 for the SS dataset and L=4 for VLOG and EPIC. We do not perform any other data augmentation. While training we extract L frames from the entire video by splitting the video into L sub-sequences and randomly sampling one frame per sub-sequence. The output sequence of size L is called a *clip*. A clip aims to represent the full video with less frames. For testing we aggregate results of 10 clips.

The ablation study is done by using the train set as training data and we report the result on the validation set. We compare against other state-of-the-art approaches on the test set. For the ablation studies, we slightly decreased the computational complexity of the model: the base network (including activity and object heads) is a ResNet-18 instead of ResNet-50, a single clip of 4 frames is extracted from a video at test time.

Comparison with other approaches — Table 1 shows the performance of the proposed approach on the VLOG dataset. We outperform the state of the art

	mAP	bag	bed	bedding	book/papers	bottle/tube	bowl	box	brush	cabinet	cell-phone	clothing	cup	door	drawers	food	fork	knife	laptop	microwave	oven	pen/pencil	pillow	plate	refrigerator	sink	spoon	stuffed animal	table	toothbrush	towel
R50 [44]	40.5	29.7	68.9	65.8	64.5	58.2	33.1	22.1	19.0	23.9	54.0	45.5	28.6	49.2	**28.7**	49.6	19.4	37.5	62.9	48.8	23.0	36.9	39.2	12.5	**55.9**	58.8	31.1	**57.4**	26.8	39.6	22.9
I3D [3]	39.7	24.9	71.7	**71.4**	62.5	57.1	27.1	19.2	**33.9**	20.7	50.6	45.8	24.7	54.7	19.1	**50.8**	19.3	**41.9**	54.0	27.5	21.4	37.4	42.9	12.6	42.5	60.4	33.9	46.0	23.5	59.6	34.7
Ours	**44.7**	**30.2**	**72.3**	70.7	**64.9**	**59.8**	**38.2**	**24.6**	26.3	22.4	**64.5**	**47.2**	**35.4**	**57.9**	25.2	48.5	**24.5**	40.2	**72.0**	**54.1**	**26.5**	**39.9**	**48.6**	**15.2**	53.5	**60.7**	**36.8**	52.8	**27.9**	**64.0**	**37.6**

Table 1. Results on Hand/Semantic Object Interaction Classification (Average precision in % on the test set) on VLOG dataset. R50 and I3D implemented by [6].

on this challenging dataset by a margin of ≈4.2 points (44.7% accuracy against 40.5% by [44]). As mentioned above, traditional video approaches tend to fail on this challenging fine-grained dataset, providing inferior results. Performance on the Something-Something dataset is given in Table 3. We outperform the state of the art given by very recent methods. On EPIC we re-implement standard baselines such CNN-2D and I3D based on a Resnet-18 and report results on the validation set (Table 4) since the test set is not available. Our full method reports an accuracy of 40.89 and outperforms baselines by a large margin (≈+6.4 and ≈+7.9 points respectively for against CNN-2D and I3D).

Effect of object-level reasoning — Table 2 shows the importance of reasoning on the performance of the method. The baseline corresponds to the performance obtained by the activity head trained alone (inflated ResNet, in the ResNet-18 version for this table). No object level reasoning is present in this baseline. The proposed approach (third line) including an object head and the ORN module gains 0.8, 2.5 and 2.4 points compared to our baseline respectively on SS, on EPIC and on VLOG. This indicates that the reasoning module is able to extract complementary features compared to the activity head.

Using *semantically defined objects* proved to be important and led to a gain of 2 points on EPIC and 2.3 points on VLOG for the full model (6/12.7 points using the object head only) compared to an extension of Santoro *et al* [12] operating on pixel level. This indicates importance of object level reasoning. The gain on SS is smaller (0.7 point with the full model and 7.8 points with the object head only) and can be explained by the difference in spatial resolution of the videos. Object detections and predictions of the binary masks are done using the initial video resolution. The mean video resolution for VLOG is 660×1183 and for EPIC is 640×480 against 100×157 for SS. Mask-RCNN has been trained on images of resolution 800×800 and thus performs best on higher resolutions. The quality of the object detector is important for leveraging object level understanding then for the rest of the ablation study we focus on EPIC and VLOG datasets.

The function f_ϕ in Equation (3) is an important design choice in our model. In our proposed model, f_ϕ is recurrent over time to ensure that the ORN module captures long range reasoning over time, as shown in Equation (5). Removing the recurrence in this equation leads to an MLP instead of a (gated) RNN, as evaluated in row 4 of Table 2. Performance decreases by 1.1 point on VLOG and 1.4 points on EPIC. The larger gap for EPIC compared to VLOG and can arguably be explained by the fact that in SS actions cover the whole video,

Method	Object type	EPIC		VLOG		SS	
		obj. head	2 heads	obj. head	2 heads	obj. head	2 heads
Baseline	-	-	*38.33*	-	*35.03*	-	*31.31*
ORN	pixel	23.71	38.83	14.40	35.18	2.51	31.43
ORN	**COCO**	**29.94**	**40.89**	**27.14**	**37.49**	10.26	**32.12**
ORN-mlp	COCO	28.15	39.41	25.40	36.35	-	-
ORN	COCO-visual	28.45	38.92	22.92	35.49	-	-
ORN	COCO-shape	21.92	37.16	7.18	35.39	-	-
ORN	COCO-class	21.96	37.75	13.40	35.94	-	-
ORN	COCO-intra	29.25	38.10	26.78	36.28	-	-
ORN clique-1	COCO	28.25	40.18	26.48	36.71	-	-
ORN clique-3	COCO	22.61	37.67	27.05	36.04	-	-

Table 2. Ablation study with ResNet-18 backbone. Results in %: Top-1 accuracy for EPIC and SS datasets, and mAP for VLOG dataset.

Methods	Top1
C3D + Avg [5]	21.50
I3D [5]	27.63
MultiScale TRN [38]	33.60
Ours	**35.97**

Table 3. Experimental results on the Something-Something dataset (classification accuracy in % on the test set).

Methods	Top1
R18 [44]*	32.05
I3D-18 [3]*	34.20
Ours	**40.89**

Table 4. Experimental results on the EPIC Kitchens dataset (accuracy in % on the validation set – methods with * have been re-implemented).

while solving VLOG requires detecting the right moment when the human-object interaction occurs and thus long range reasoning plays a less important role.

Visual features extracted from object regions are the most discriminative, however object shapes and labels also provide complementary information.

Finally, the last part of Table 2 evaluates the effect of using pairwise cliques in the clique set $\mathcal{S}$ in Eq. (3) and shows that pairwise cliques outperform cliques of size 1 and 3. We would like to recall, that even with unary cliques only, interactions between objects are still modeled. However, the model needs to find subspaces in the hidden representations associated to each interaction.

CNN architecture and kernel inflations — The convolutional architecture of the model was optimized over the validation set of the SS dataset, as shown in Table 5. The architecture itself (in terms of numbers of layers, filters etc.) is determined by pre-training on image classification. We optimized the choice of filter inflations from 2D to 2.5D or 3D for several convolutional blocks. This has been optimized for the single head model and using a ResNet-18 variant to speed up computation. The hyper-parameter search indicates that the differences in performance are relatively large, with performance increases being up to 100% w.r.t. to the pure 2D baselines. This indicates, without surprise, that motion is a strong cue. Inflating kernels to 2.5D on the input side and on the output side provided best performances, suggesting that temporal integration

Conv1			Conv2			Conv3			Conv4			Conv5			Aggreg		SS
2D	3D	2.5D	2D	3D	2.5D	2D	3D	2.5D	2D	3D	2.5D	2D	3D	2.5D	GAP	RNN	
✓	-	-	✓	-	-	✓	-	-	✓	-	-	✓	-	-	✓	-	15.73
✓	-	-	✓	-	-	✓	-	-	✓	-	-	✓	-	-	-	✓	15.88
-	✓	-	-	✓	-	-	✓	-	-	✓	-	-	✓	-	✓	-	31.42
-	-	✓	-	-	✓	-	-	✓	-	-	✓	-	-	✓	✓	-	27.58
✓	-	-	✓	-	-	✓	-	-	✓	-	-	-	✓	-	✓	-	31.28
✓	-	-	✓	-	-	✓	-	-	-	✓	-	-	✓	-	✓	-	32.06
✓	-	-	✓	-	-	-	✓	-	-	✓	-	-	✓	-	✓	-	32.25
✓	-	-	✓	-	-	✓	-	-	✓	-	-	-	-	✓	✓	-	31.31
✓	-	-	✓	-	-	✓	-	-	-	-	✓	-	-	✓	✓	-	32.79
✓	-	-	✓	-	-	-	-	✓	-	-	✓	-	-	✓	✓	-	**33.77**
-	✓	-	✓	-	-	✓	-	-	✓	-	-	✓	-	-	✓	-	28.71
-	✓	-	-	✓	-	✓	-	-	✓	-	-	✓	-	-	✓	-	31.42
-	-	✓	✓	-	-	✓	-	-	✓	-	-	✓	-	-	✓	-	20.05
-	-	✓	-	-	✓	✓	-	-	✓	-	-	✓	-	-	✓	-	22.52

Table 5. Effect of the CNN architecture (choice of kernel inflations) on a single head ResNet-18 network. Accuracy in % on the validation set of Something-Something is shown. 2.5D kernels are separable kernels: 2D followed by a 1D temporal.

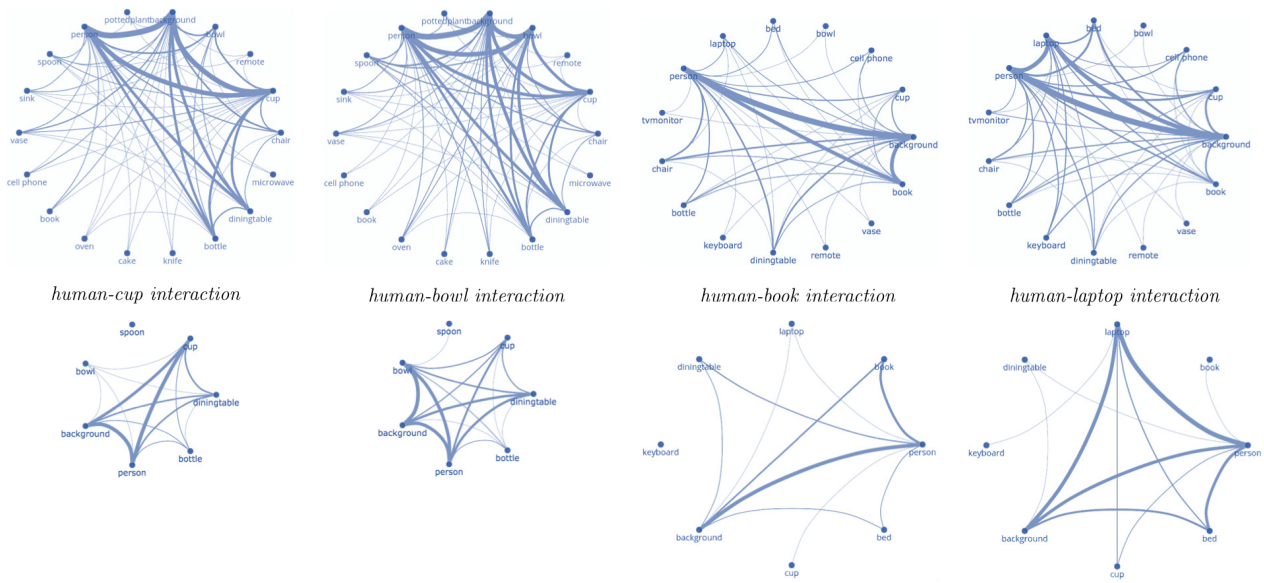

Fig. 4. Example of object pairwise interactions learned by our model on VLOG for four different classes. Objects co-occurrences are at the top and learned pairwise objects interactions are at the bottom. Line thickness indicates learned importance of a given relation. Interactions have been normalized by the object co-occurrences.

is required at a very low level (motion estimation) as well as on a very high level, close to reasoning. Our study also corroborates recent research in activity recognition, indicating that 2.5D kernels provide a good trade-off between high-capacity and learnable numbers of parameters. The choice of a (gated) RNN for temporal integration of the activity head features proved important (see Table 5) compared to global average pooling (GAP) over space and time.

Visualizing the learned object interactions — Figure 4 shows visualizations of the pairwise object relationships the model learned from data, in

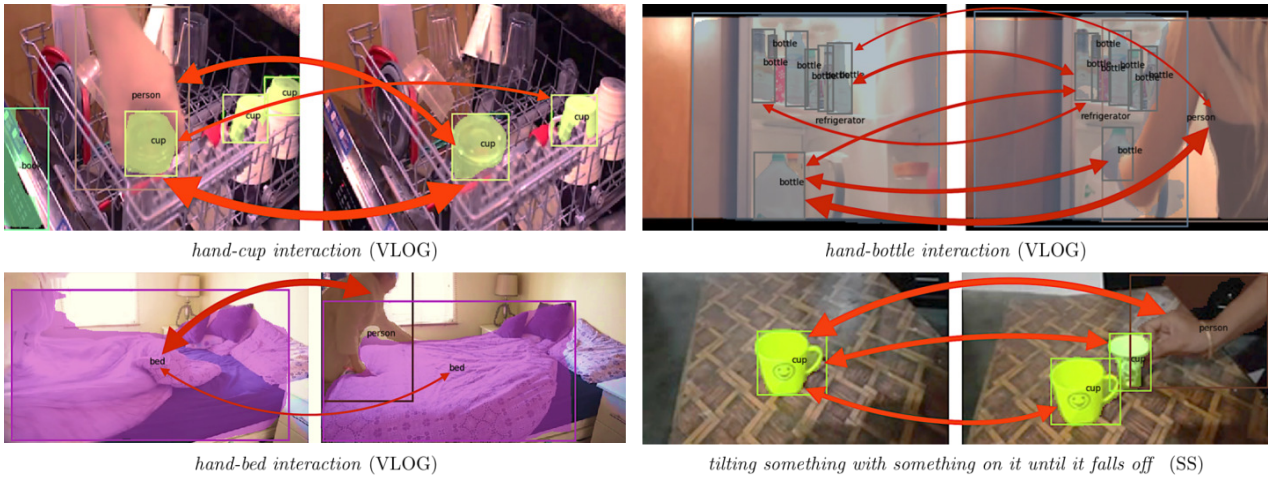

Fig. 5. Example frame pairs of the VLOG dataset and the Something-Something dataset shown with overlaid object instance masks. Line thickness indicates learned importance of a given relation.

Fig. 6. Examples of failure cases – a) small sized objects (on the left). Our model detects a *cell phone* and a *person* but fails to detect *hand-cell-phone contact*; b) confusion between semantically similar objects (on the right). The model falsly predicts *hand-cup contact* instead of *hand-glass-contact* even though the *wine glass* is detected.

particular from the VLOG dataset. Each graph is computed for a given activity class, and strong arcs between two nodes in the graph indicate strong relationships between the object classes, i.e. the model detects a high correlation between these relationships and the activity. The graphs were obtained by thresholding the summed activations of each pairwise relationship (j, k) in equation (4). Each pair (j, k) can be assigned a pair of object classes $\mathbf{c}_t^j$ and $\mathbf{c}_t^k$ through the predictions of the object instance mask predictor. Integrating over all samples of the dataset for a given class leads to the visualizations in Figure 4. We can see that the object interactions are highly relevant to the detected activities: the *person-touches-bed* activity is correlated to interactions between relevant object classes *person* and *bed*. Similarly, activities *human-bowl interaction* and *human-cup interaction* show interactions with the respective objects *bowl* and *cup*. Moreover, other recovered relationships are highly correlated to the scene (for example, *dining-table* and *bowl* for activity *human-bowl interaction*).

Finally, Figure 5 shows example frames of the datasets and masks predicted by Mask R-CNN and Figure 6 shows some failure cases, which are either due to errors done by the object mask prediction (Mask R-CNN) or by the ORN itself.

7 Conclusion

We presented a method for activity recognition in videos which leverages object instance detections for visual reasoning on object interactions over time. Our model is differentiable and fully trainable and learns interactions from training data. The choice of reasoning over semantically well-defined objects is key to our approach and outperforms state of the art methods which reason on grid-levels, such as cells of convolutional feature maps. Object features are learned with a two-headed neural model, integrating losses on activity recognition and object recognition. Temporal dependencies and causal relationships are dealt with by integrating relationships between different time instants. We evaluated the method on two difficult datasets, on which standard approaches do not perform well, and report state-of-the-art results. We also show visualizations of the learned object-level interactions and demonstrate that they are highly relevant to the underlying activity classes.

References

1. Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., Vijayanarasimhan, S.: Youtube-8m: A large-scale video classification benchmark. arxiv:1609.08675 (2016)
2. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In: CVPR. (2016)
3. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR. (2017)
4. Monfort, M., Zhou, B., Bargal, S.A., Andonian, A., Yan, T., Ramakrishnan, K., Brown, L., Fan, Q., Gutfruend, D., Vondrick, C., Oliva, A.: Moments in time dataset: one million videos for event understanding. In: arxiv:1801.03150. (2018)
5. Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., Hoppe, F., Thurau, C., Bax, I., Memisevic, R.: The "something something" video database for learning and evaluating visual common sense. In: ICCV. (Oct 2017)
6. Fouhey, D.F., Kuo, W., Efros, A.A., Malik, J.: From lifestyle vlogs to everyday interactions. In: CVPR. (2018)
7. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalanditis, Y., Li, L.J., Shamma, D.A., Bernstein, M., Fei-Fei, L.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International Journal of Computer Vision (IJCV) **123** (2017) 32–73
8. 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. Arxiv (2017)
9. Stabinger, S., Rodríguez-Sánchez, A., Piater, J.: 25 years of CNNs: Can we compare to human abstraction capabilities? In: ICANN. (2016)
10. Fleuret, F., Li, T., Dubout, C., Wampler, E.K., Yantis, S., Geman, D.: Comparing machines and humans on a visual categorization test. Proceedings of the National Academy of Sciences of the United States of America **108 43** (2011) 17621–5
11. Perez, E., Vries, H.D., Strub, F., Dumoulin, V., Courville, A.: Learning visual reasoning without strong priors. In: ICML 2017's Machine Learning in Speech and Language Processing Workshop. (2017)

12. Santoro, A., Raposo, D., Barrett, D.G., Malinowski, M., Pascanu, R., Battaglia, P., Lillicrap, T.: A simple neural network module for relational reasoning. In: NIPS. (2017)
13. Watters, N., Zoran, D., Weber, T., Battaglia, P., Pascanu, R., Tacchetti, A.: Visual interaction networks: Learning a physics simulator from video. In: NIPS. (2017) 4542–4550
14. Hudson, D., Manning, C.: Compositional attention networks for machine reasoning. In: ICLR. (2018)
15. Kim, J., Ricci, M., Serre, T.: Not-so-CLEVR: Visual relations strain feedforward neural networks (2018)
16. Pickup, L.C., Pan, Z., Wei, D., Shih, Y., Zhang, C., Zisserman, A., Scholkopf, B., Freeman, W.T.: Seeing the arrow of time. In: CVPR. (2014)
17. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action Recognition by Dense Trajectories. In: CVPR. (2011)
18. Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., Baskurt, A.: Sequential deep learning for human action recognition. In: HBU. (2011)
19. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: ICCV. (2015)
20. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., Fei-Fei, L.: Imagenet large scale visual recognition challenge. IJCV **115**(3) (2015) 211–252
21. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning for video understanding. Pre-print: arxiv:1712.04851 (2017)
22. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8) (1997) 1735–1780
23. 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)
24. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR. (2014)
25. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NIPS. (2014)
26. Luvizon, D., Picard, D., Tabia, H.: 2d/3d pose estimation and action recognition using multitask deep learning. In: CVPR. (2018)
27. Jain, A., Zamir, A.R., Savarese, S., Saxena, A.: Structural-RNN: Deep Learning on Spatio-Temporal Graphs. In: CVPR. (2016)
28. Deng, Z., Vahdat, A., Hu, H., Mori, G.: Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In: CVPR. (2016)
29. Song, S., Lan, C., Xing, J., Zeng, W., Liu, J.: An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data. In: AAAI Conf. on AI. (2016)
30. Sharma, S., Kiros, R., Salakhutdinov, R.: Action recognition using visual attention. ICLR Workshop (2016)
31. Sun, L., Jia, K., Chen, K., Yeung, D., Shi, B.E., Savarese, S.: Lattice long short-term memory for human action recognition. In: ICCV. (2017)
32. Yeung, S., Russakovsky, O., Jin, N., Andriluka, M., Mori, G., Fei-Fei, L.: Every moment counts: Dense detailed labeling of actions in complex videos. arXiv preprint arXiv:1507.05738 (2015)
33. Baradel, F., Wolf, C., Mille, J., Taylor, G.: Glimpse clouds: Human activity recognition from unstructured feature points. In: CVPR. (2018)

34. Battaglia, P.W., Pascanu, R., Lai, M., Rezende, D.J., Kavukcuoglu, K.: Interaction networks for learning about objects, relations and physics. In: NIPS. (2016)
35. van Steenkiste, S., Chang, M., Greff, K., Schmidhuber, J.: Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. ICLR (2018)
36. Velikovi, P., Cucurull, G., Casanova, A., Romero, A., Li, P., Bengio, Y.: Graph attention networks. In: ICLR. (2018)
37. Luc, P., , Neverova, N., Couprie, C., Verbeek, J., LeCun, Y.: Predicting deeper into the future of semantic segmentation. In: ICCV. (2017)
38. Bolei, Z., Zhang, A.A., Torralba, A.: Temporal relational reasoning in videos. arXiv preprint arXiv:1711.08496v1 (2017)
39. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: CVPR. (2017)
40. Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R.R., Smola, A.J.: Deep sets. In: NIPS. (2017) 3391–3401
41. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask r-cnn. In: ICCV. (2017)
42. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS. (2015) 91–99
43. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: ICML. (2015)
44. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)
45. Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., Wray, M.: Scaling egocentric vision: The epic-kitchens dataset. arXiv preprint arXiv:1804.02748 (2018)