Classifying Collisions with Spatio-Temporal Action Graph Networks

Roei Herzig\textsuperscript{1*}, Elad Levi\textsuperscript{2*}, Huijuan Xu\textsuperscript{3*}, Eli Brosh\textsuperscript{2}, Amir Globerson\textsuperscript{1}, Trevor Darrell\textsuperscript{2,3}

\textsuperscript{1}Tel Aviv University, \textsuperscript{2}Nexar, \textsuperscript{3}UC Berkeley

\textsuperset{1}{roeiherz, gamir}@mail.tau.ac.il \textsuperset{2}{elad, elibrosh}@getnexar.com \textsuperset{3}{huijuan, trevor}@cs.berkeley.edu

Abstract

Events defined by the interaction of objects in a scene often are of critical importance, yet such events are typically rare and available labeled examples insufficient to train a conventional deep model that performs well across expected object appearances. Most deep learning activity recognition models focus on global context aggregation and do not explicitly consider object interactions inside the video, potentially overlooking important cues relevant to interpreting activity in the scene. In this paper, we show that a new model for explicit representation of object interactions significantly improves deep video activity classification for driving collision detection. We propose a Spatio-Temporal Action Graph (STAG) network, which incorporates spatial and temporal relations of objects. The network is automatically learned from data, with a latent graph structure inferred for the task. As a benchmark to evaluate performance on collision detection tasks, we introduce a novel data set based on data obtained from real life driving collisions and near-collisions. This data set reflects the challenging task of detecting and classifying accidents in a richly varying but yet highly constrained setting, that is very relevant to the evaluation of autonomous driving and alerting systems. Our experiments confirm that our STAG model offers significantly improved results for collision activity classification.

1. Introduction

Recognition of rare events in natural scenes poses a challenge for deep learning approaches to activity recognition, since an insufficient number of training examples are typically available to learn all required conditions. E.g., in driving scenarios critical events are often a function of the spatial relationship of prominent objects, yet are rare enough that they will not be experienced across a sufficient range of object appearances. Conventional deep models work well when they are supervised across the range of conditions they will experience, and conversely, may perform poorly on new object-appearance combinations. We desire an activity recognition model which can generalize effectively across object appearance and intra-object relationships.

With the advent of deep learning techniques, models have generally been limited to holistic representations, directly applying convolutional filters to full video frames. Typical methods included an LSTM conditioned on per-frame CNN features [7], 3-dimensional convolutional networks (3D ConvNets) [37] and two-Stream convolutional networks [35]. None of these models explicitly leverage object interactions to help activity classification, though most activities contain common object interaction patterns which are a clear cue for the activity category.

Historically, objects and their interactions have been considered valuable features for understanding activities, represented by spatio-temporal tubes (e.g., [12, 44]), spatially-aware embeddings [28], and/or spatio-temporal graphical models (e.g., [30, 3, 13]). But generally, when learning-based approaches were considered, the addition of explicit object detection often did not show significant improvements in real-world evaluation settings. The most

*Equal Contribution.
successful SVM-based video classification methods were largely based on handcrafted and holistic features including bag of word descriptors, dense trajectories, and motion boundary descriptors [39, 41, 40].

Recently, Wang et al. addressed the role of object temporal properties in deep activity classification [42]. They build a temporal graph to model object evolution, and combine this graph feature encoding with the holistic convolutional encoding of the whole video to improve video classification. While they show temporal graph modeling of boxes offers significant improvement on activity classification, their model did not consider the spatial relation of boxes in each frame. In addition, in [42] the temporal graph connection is defined by two metrics, the similarity relation of correlated objects and the spatial-temporal relations between nearby objects in adjacent frames. However, the two metrics may be too rigid to find the most suitable temporal connection for a specific dataset.

In this paper, we explore whether spatio-temporal object relationships offer improved activity recognition performance. We propose a Spatio-Temporal Action Graph (STAG) with both explicit spatial and explicit temporal connectivity between objects. STAGs not only model the object appearance relationship on a per-frame base but also models the temporal evolution of object relations. We learn the temporal relation structures from data, which may be more suitable for specific datasets and/or tasks. In Fig. 1, we demonstrate our approach on a driving scene for collision event detection.

To evaluate our model on realistic critical driving scenarios, we introduce a challenging dataset based on real-world dashcam data. We collect 803 videos from more than 10 million rides, focusing on rare events of collision and near-collision scenarios experienced by the (dashcam) vehicle, typically involving its contact with a fixed or moving object such as another vehicle or pedestrian. Such kinds of activity tasks, which involve recognition of interaction with a specific well detected object, seem to fit well to our proposed approach.

Our experimental results validate the thesis that both spatial and temporal relations of objects are important for activity classification, and that explicitly modeling interactions offers advantages over one based on implicit temporal modeling.

With an arbitrary amount of training data, end-to-end learning with a suitably high-capacity network could implicitly learn object relationships relevant to recognition tasks. But such amounts of training data are not realizable in practice. Our results confirm that explicit object representations can offer better generalization performance for deep activity recognition in realistic conditions with limited training data.

The following section reviews relevant work on activity classification and graph models. Sec. 3 describes our approach to object-aware activity recognition using a STAG architecture. Sec. 5 shows the experimental results on Collisions dataset. Sec. 6 concludes the paper.

2. Related Work

Video Classification Early deep learning activity classification systems are in the spirit of bag of words models: per-frame features are pooled over an entire video sequence [22]. Simply pooling the per-frame features will ignore temporal structure in the video. Later work used a recurrent neural network, such as an LSTM, to temporally aggregate per-frame features [7, 48].

Another line of activity classifiers use 3D spatio-temporal filters to create hierarchical representations of the whole input video sequence [19, 36, 37, 38]. While spatio-temporal filters can benefit from large video datasets, they cannot be pretrained on images. Two-stream networks are proposed to leverage image pretraining in RGB as well as capturing fine low-level motion in another optical flow stream [35, 9]. I3D[4] is designed to inflate 2D kernels to 3D to learn spatio-temporal feature extractors from video while leveraging image pretrain weights. In all these aforementioned activity classifiers, whole frame video features are extracted without looking into the object details inside each frame. Recently some works have investigated the effect of either temporal or spatial object relations in activity classification [43, 26]. Our paper follows this direction and investigates the spatio-temporal relationship of objects for activity classification.

Graph Models Traditional graph models include Bayesian Networks and Conditional Random Field (CRF). Scene Graphs were first used for rich image representation in the image retrieval tasks [21]. To apply deep learning to graph models, various graph convolution operations and architectures have been proposed [16, 23]. One unique characteristic of graph models is their relational reasoning ability [2, 49, 1]. Graph models have been applied on many applications, including image generation [20] and robotics [34]. Models for automatically constructing scene graphs based on a Faster R-CNN architecture has been proposed in [46]. In this paper, we build an object graph model to improve performance on video activity classification tasks.

Object Detection Faster R-CNN [32], extends R-CNN [11] and Fast R-CNN [10] object detection approaches, incorporating region of interest (RoI) pooling and a region proposal network. More recent object detection models include SSD [25], R-FCN [5] and YOLO [31] which achieve faster and better object detection. Mask R-CNN [14] extends Faster R-CNN by adding one extra branch for predicting object masks in parallel with original bounding box predictions. It efficiently detects objects
Figure 2. Object-aware Activity Recognition architecture. The input consists of a sequence of 20 frames. (1) A backbone of ResNet50 and RPN produces a set of bounding boxes proposals for each frame. (2) A RoiAlign Single layer extracts $z_t^i$ features from the backbone using the boxes. In parallel, every pair of box proposals is used for computing a union box, and pairwise features $z_t^{i,j}$ extracted similar to the $z_t^i$ features. (3) The $z_t^i$ and $z_t^{i,j}$ are used as inputs to an Spatio-Temporal Action Graph Network Module which outputs a video representing $1 \times d$ processed by an attention over frames and objects from the new and improved features using graph networks approach. (4) Last, the model outputs collision classification.

in an image while simultaneously generating a high-quality instance segmentation mask. In this paper we take the high quality proposals from Mask R-CNN to build the spatial-temporal graph over them for activity classification. Object detection proposals serve as a foundation for several computer vision tasks like visual grounding [18], robot learning [6] etc. Here we try to leverage the object proposals for activity classification task.

**Objects for activities** [24] classifies the event in the image as well as providing a number of semantic labels to the objects and scene environment in the image. They show that using scenes or objects alone cannot achieve good performance. [29] is the early work to exploit human motion and object context to perform action recognition and object classification. Since many activity classes are related to humans, a variety of papers aim to model the interaction of human and objects for activity recognition [33, 13, 47]. [8] exploits the consistent appearance of objects, hands, and actions from the egocentric view for activity recognition. They also show that joint modeling of activities, actions, and objects leads to superior performance compared to the case where they are considered independently. [45] proposes an approach for activity recognition based on detecting and analyzing the temporal sequence of objects. We investigate spatio-temporal object interactions in the car-road scene for collision classification, and demonstrate the benefit of explicit modeling object relationships through space and time.

3. Spatio-Temporal Action Graph

In what follows we describe our model for object-aware recognition using a Spatio-Temporal Action Graph Network (STAG). The key insight of the Spatio-Temporal Action Graph module is that actions are characterized by the way different objects interact in the image across space and time. To capture these, our model is constructed as follows. First, a detector module is applied to the image to extract bounding boxes of objects of interest (see Section 3.1). This results in a set of vector representations for the bounding boxes, and pairs of bounding boxes. This is done for each frame in the sequence.

Next, these representations are re-processed via a graph network, resulting in a new representation per box, where now each such representation has incorporated information from all other boxes. Therefore, the new representation can take context into account. Again, this is done per time-frame. Additionally, at each time frame, a per-frame representation is calculated.

Finally, information is aggregated across frames, by using a variant of an attention module. This results in a representation of the entire sequence, which is then used for classifying the action. Those, design choices in the above architectures are specifically targeted at modeling spatio-temporal patterns, whereby information is gradually aggregated along both these axes.

We next describe each of the components in more detail. In what follows we use the following constants: $T$ is the number of frames in the sequence to be classified, $N$ is the maximum number of objects (i.e., bounding boxes) per
Figure 3. A schematic representation of the Spatio-Temporal Action Graph Network Module (a) First, the features $z_{i,j}^{t}$ are processed element-wise by $\alpha$. (b) Features are summed using an attention over the neighbors per object to create a vector $s_i$. (c) A representation of the entire graph per frame is created by applying $\beta$ and summing the created vector using an attention over the objects. (d) An improved object features $z_{i,j}^{t}$ are finally processed by $\gamma$, together with a concatenated $z_{i}^{t}$ features with an expanded graph features per frame. (e) An attention over frames and objects is been applied to get a feature sized $1 \times d$ which represent a video.

Thus overall, we have a tensor of size $T \times N \times d$ for single object features, and a tensor of size $T \times N \times N \times d$ for object-pair features.

### 3.2. STAG Module

In order to model the interaction between multiple objects in the same frame, we use a graph network that integrates information across all boxes and pairs of boxes in a single frame. The network, inspired by the architecture in [17], is shown in Fig. 3. It progressively aggregates over information to eventually obtain a vector in $\mathbb{R}^d$ describing the whole sequence. At the first, the features of $z_{i,j}^{t}$ are processed element-wise by $\alpha$. Then, a reduction step a tensor of size $T \times N \times N \times d$ is reduced to $T \times N \times d$ by having each box “attend” to all the other boxes, thereby integrating information from across the frame. After that, those features, are processed by $\beta$, and in the next reduction phase, $T \times N \times d$ is transformed to $T \times d$ via a soft-attention over boxes per time frame. This results in a feature vector describing each complete frame. At this point, we reprocess the single box features $z_i^{t}$ using the complete frame features by $\gamma$ to get an improved features $z_i^{t}$. To obtain a representation for the whole sequence we reduce over $T \times N$ resulting in a vector $\mathbb{R}^d$ for the entire sequence. Finally, we apply a multi-layer neural network to this vector, with a softmax classifier. As we described later in Sec. 5.1, $\alpha$, $\beta$ and $\gamma$ are functions mapping features from one dimension to another.

### 3.3. Object Alignment

The motion of the objects is a critical key in order to correctly recognize the action in a video clip. Since the order of the detected objects will typically vary between different frames (as it just corresponds to the order in which the RPN ranks these), there is a need to align the objects in order to mix the objects information across the temporal dimension. In cases where there is a need to recognize activity of one main object (such as car collisions), the network architecture can exploit this property and concentrate on one object.
at each frame.

The way we implement this idea in the network architecture is by using soft attention on the objects (see model description above), in order to reduce the object axis, allowing the network to concentrate on specific objects. Thus, instead of explicitly modeling all object interactions between frames (which requires a fixed object ordering across all frames), we first aggregate object relationships, then model temporal relationships. After the dimensions reduction, we left with one feature vector per frame. We then combine all the frames together, reduced the temporal dimension and predict the action.

4. The Collisions Dataset

To evaluate our approach on realistic driving scenes, we introduce a dataset comprising of real-world driving videos of rare events. These events encompass collisions scenarios, i.e., scenarios involving the contact of the (dashcam) vehicle with a fixed or moving object; and near-collisions, i.e., scenarios requiring an evasive maneuver to avoid a crash. Such driving scenarios most often contain interactions between two vehicles, or between a vehicle and a pedestrian or a bike, and thus are suitable for modeling object interactions. The dataset contains rare events from diverse driving scenes including urban and highway areas in several large cities in the US.

**Data collection.** The data was collected from a large-scale deployment of connected dashcams. Each vehicle is equipped with a dashcam and a companion smartphone app that continuously captures and uploads sensor data such as GPS, IMU and gyroscope readings. Overall the vehicles collected more than 10 million rides. Rare events are automatically detected using a triggering algorithm based on the IMU and gyroscope sensor data. The events are then manually validated by human annotators based on visual inspection to identify edge case events of collisions and near-collisions, as well as safe driving events. Of all the detected triggers, our subset contains 743 collisions and 60 near-collisions, all from different drivers. For each event, we maintain a video, and the time of impact (or near-impact), which typically occurs in the middle of the video. Each video is approximately 40 seconds in length and has a resolution of 1280 × 720.

**The full and few-shot datasets** To investigate the effect of the proposed graph model, we use the events to create two datasets: full and few-shot. The full dataset contains 803 videos. We use 732 videos as training data (44 of them are near-collision) and 71 as test data (6 of them are near-collision). The few-shot dataset contains only 125 videos, 25 videos used as training data and 100 as test data. We temporally downsample the videos to 5 frames per second to avoid feeding near-duplicate frames into our model. We use segments of the videos for model training to fully leverage our data. Specifically, we split each video into three segments, two non-overlapping segments of 20 frames each from before the time of impact, and one segment starting at the time of impact. So that we have two negative segments (non-risky driving scenes) and one positive segment (a collision scene) for each collision event, and three negatives for each near-collision event. After this processing, we have a total 2409 video segments, out of which 1656 are negative examples and 753 are positives.

**Diversity** Our dataset is collected to learn to recognize collisions in natural driving scenes. The coverage of the dataset includes various types of scenes, the parties involved, and lighting conditions. Fig. 5 shows different examples of collision scenes under various weather and lighting conditions. We use visual inspection to analyze the identity of 3rd parties involved in the collisions. We find that most of the data (85%) consists of crashes involving two vehicles, and collisions involving pedestrians and cyclists tally up to 6% each. Table 1 shows the distribution of the 3rd party types involved, as well as weather and lighting conditions.

Finally, we will release a publicly available challenge based on this data upon acceptance of paper.

5. Experiments

We next describe the results of applying our model to the dataset in Sec. 4, compare it a state of the art activity recognition classifier, and study model variants.

5.1. Model Details

We chose the following parameters: \( T = 20 \) (number of frames per video; we use a randomly sub-sampled segment with \( T \) frames from the dataset, as described in Sec. 4.), \( N = 12 \) (number of boxes per frame) and \( d = 512 \) (feature dimension per box).

**Detector.** We used Mask R-CNN with ResNet50 as a backbone to train a sequence of \( T \) frames corresponding to one video. The FPN was trained with strides \((4, 8, 16, 32, 64)\). The RPN trained with anchor scales \((32, 64, 128, 256, 512)\) and aspect ratios \((0.5, 1, 2)\) and anchor stride of 1. The input image was resized with padding to 256, which is the

| Party type | dist. | Weather | dist. | Lighting | dist. |
|-----------|-------|---------|-------|----------|-------|
| Vehicle   | 85%   | Clear   | 93%   | Day      | 62%   |
| Bike      | 6%    | Rain    | 5.3%  | Night    | 38%   |
| Pedestrian| 6%    | Snow    | 1.7%  |          |       |
| Road object| 1%    |         |       |          |       |
| Motorcycle| 1%    |         |       |          |       |

Table 1. Rare event dataset statistics: 3rd party, weather, and lighting conditions.
Figure 5. Different examples of the dataset: (a) extreme weather conditions such as snow and heavy raining. (b) near collision of a truck and a bicycle rider. (c) day and night collisions.

maximum dimension. The RPN proposals were filtered by non-maximum suppression with IoU threshold of 0.7. The model then subsampled the most likely 12 ROIs from an initial 2000 ROIs of the RPN. Using a higher number of ROIs could potentially be a major issue for time and space complexity, however, meaningful scene could be represented by only 12 objects because they capture sufficient information of the scene.

Features for the 12 objects and 12 · 11 relations are extracted as explained in Sec. 3 resulting in features $z_i$ for objects and $z_{i,j}$ for relations. Both the objects and relations ROIs are pooled to $7 \times 7$ followed by $1 \times 1$ convolution layer to be $T \times N \times d$ and $T \times N \times N \times d$, respectively. The Mask R-CNN was pretrained on the BDD dataset [27] using the default split as the author suggested.

Spatio-Temporal Action Graph Module. The $\alpha$, $\beta$ and $\gamma$ modules are implemented using 3 fully connected (FC) layers (see Fig. 3) with a $d$ dimensional output each. The attention is implemented in the standard way using another as a network like $\alpha$, $\beta$ and $\gamma$, receiving the same inputs, but using the output scores for the attention for reducing dimension in the model.

5.2. Training Details

Our model was trained using SGD with a momentum of 0.9 and a learning rate 0.01. The learning rate was decayed by a factor of 0.5 each epoch, and the gradient was clipped at norm 5. Each batch includes a video segment of $T$ frames that includes ground truth detection annotations per frame and collision annotation per segment. The detection annotations came from the pretrained Mask-RCNN, which trained on BDD [27]. The loss of the Object-aware Activity Recognition was a sum of the following two components: the Mask-RCNN detector and the Spatio-Temporal Action Graph network. We trained the Object-aware Activity Recognition with a ratio of 1:3 of positive to negatives, meaning the Spatio-Temporal Action Graph was able to learn both positive and negatives examples.

Detector. The Backbone and RPN are trained in with a loss that consisting of RPN classification and regression, and the classification and regression of the second stage, as is standard with two stage detectors. The two losses are given the same weight.

Spatio-Temporal Action Graph Classification Loss

The Spatio-Temporal Action Graph network (Sec. 3.2) has two logits at its output, corresponding to the two possible actions ("collision" or "no collision"). Since we have the ground-truth for this class, we simply use a binary cross entropy loss between the logits and the ground-truth label.

5.3. Model Variants

We refer to our model as a Spatio-Temporal Action Graph or STAG (see Sec. 5.2). STAG progressively processes the RPN features $z_i^t$, $z^t_{i,j}$ into a single vector describing the whole frame sequence. Through this process, it can capture the spatio-temporal aspects of the
sequence. Another approach to capturing temporal structure are recursive neural nets and in particular LSTM. Here we describe two LSTM variants that use some aspects of STAG, but replace the temporal processing in STAG with an LSTM model. We considered the following models:

1) LSTM Boxes - An LSTM model whose input at time \( t \) are the RPN features for the \( B \) boxes. This model does not use any component in Fig. 3. The final state of the LSTM is used for predicting the binary action label “collision” or “no collision”.

2) LSTM Spatial Graph - An LSTM model as above, but whose input at time \( t \) is the \( z_{it}' \) vector in Fig. 3 (i.e., the output of \( \gamma \) function). Thus, this model replaces the final attention-based reduction step in STAG with LSTM reduction.

5.4. A C3D Baseline

We compare our approach with the C3D [37] approach, which uses 3D ConvNet to extract rich spatio-temporal feature hierarchies from a given input video buffer. We adopt the convolutional layers (conv1a to conv5b) and fc layers(fc6 to fc8) of C3D model. The input to our model is a sequence of RGB video frames with dimension \( \mathbb{R}^{3 \times L \times H \times W} \). The height (\( H \)) and width (\( W \)) of the frames are taken as 112 each following [37]. The number of frames \( L \) is set as 16 frames sampled from each video. We use the Sports-1M pretrained C3D weights to initialize the model used in our training. In order to show the complementary effect of the video feature encoding using spatio-temporal filters, we fuse the prediction from the C3D model and our proposed STAG model by averaging the classification scores from both models. Similarly, C3D model is fused with other model variants LSTM Boxes and LSTM Spatial Graph (listed in Sec. 5.3).

5.5. Results

As described in Sec. 4, we have two datasets; the full dataset and the few-shot dataset. Table 2 reports classification accuracy on these two datasets, for our Spatio-Temporal Action Graph (STAG) model as well as two of its variants and the C3D model. First, it can be seen that STAG outperforms all the other models. Second, the LSTM BOXES model performs very poorly, highlighting the importance of using more expressive per-frame representations. This, LSTM still underperforms STAG even when using part of the STAG representation, highlighting the advantage of the attention based reduction in Fig. 3. Fourth, on the Few-Shot data the performance of all models decreased significantly, as expected. However, in these cases the STAG model still outperforms the others.

Table 3 reports accuracy for models fused with C3D. It can be seen that fusion improves the STAG models in all settings.

5.6. Qualitative Analysis

Visual inspection of success and failure cases reveals an interesting pattern. We observe that STAG outperforms C3D on “near-collision” cases such as that in the upper row of Fig. 6. Correctly classifying such cases requires understanding the relative configuration of the objects in order to determine if there was an incident or not. On the other hand C3D may put too much weight on events such as speed-change which are not always predictive of an accident.

The C3D model outperforms STAG on cases where objects are not clearly visible. For example in the lower row of Fig. 6.

Because C3D and STAG have complementary strengths, their fusion results in a stronger model, as shown in Table 3.

5.7. Attention Visualization

Our STAG model uses attention across space and time to reduce a video into a single vector. One advantage of attention models is their interpretability. Specifically, one can view the attention map that the model uses to understand which parts of the input have more influence on the decision. For the case of collision detection, this is clearly an important feature. Fig. 7 provides a nice illustration of the insight that attention maps provide. The figure shows heat maps per scene for the attention maps used in Fig. 3. It can be seen that objects that pose more danger to the driver tend to receive higher attention values.

6. Conclusion

In this paper, we introduce an approach to action recognition that specifically focuses on objects and their spatio-
Figure 6. Success and failure cases for STAG. **Top Row**: a near-collision example (i.e., the correct label is “no collision”) where the STAG model correctly classifies, and the C3D model does not. **Middle Row**: a collision example (i.e., the correct label is “collision”), where the C3D model is correct, while the STAG module errs. **Bottom Row**: a collision example (i.e., the correct label is “collision”), where the STAG module which was trained on the Few-shot dataset is correct.

Figure 7. **Explainability via Attention**. A sequence of frames, with superimposed heat-map as calculated using the attention component in Fig. 3. It can be seen that more “dangerous” objects tend to receive more attention.

temporal interaction. Our approach integrates context from across the image, and then aggregates information using multiple uses of attention. Empirical results show that our STAG model outperforms the C3D model, as well as LSTM based approaches.

The use of attention turns out to be particularly useful in terms of explainability, since the model chooses to attend to risk factors in the video.

We believe these results highlight the importance of modeling objects for activity recognition. One promising extension of STAG is to consider semantic segmentation of objects, which may be particularly important for modeling pedestrian behavior. Another is to extend the model to longer time sequences, where more elaborate spatio-temporal patterns may be relevant. Finally, an exciting challenge is to integrate such prediction networks in autonomous driving systems, where the controller can use input from the collision predictor.

Our proposed model architecture mainly addresses the use case where there are prominent objects and their relations in a driving scene which determine the activity class. In future work we would like to address more complex sce-
scenarios of activities and scenes which contain high order of complexity between multiple objects. An example for such a use case for activity recognition in the driving field could be understanding third party collisions, which could be useful in a real life scenarios.

References

[1] F. Baradel, N. Neverova, C. Wolf, J. Mille, and G. Mori. Object level visual reasoning in videos. *arXiv preprint arXiv:1806.01657*, 2018. 2

[2] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. 2

[3] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. In *Computer vision (ICCV), 2011 IEEE international conference on*, pages 778–785. IEEE, 2011. 1

[4] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017. 2

[5] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In *Neural Information Processing Systems*, pages 379–387, 2016. 2

[6] C. Devin, P. Abbeel, T. Darrell, and S. Levine. Deep object-centric representations for generalizable robot learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7111–7118. IEEE, 2018. 3

[7] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015. 1, 2

[8] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In *Computer vision (ICCV), 2011 IEEE International Conference on*, pages 407–414. IEEE, 2011. 3

[9] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016. 2

[10] R. Girshick. Fast R-CNN. In *IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 2

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *IEEE CVPR*, pages 580–587, 2014. 2

[12] G. Gkioxari and J. Malik. Finding action tubes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 759–768, 2015. 1

[13] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1775–1789, 2009. 1, 3

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017. 2

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV), 2017*. 4

[16] R. Herzig, M. Raboh, G. Chechik, J. Berant, and A. Globerson. Mapping images to scene graphs with permutation-invariant structured prediction. *arXiv preprint arXiv:1802.05451*, 2018. 2

[17] R. Herzig, M. Raboh, G. Chechik, J. Berant, and A. Globerson. Mapping images to scene graphs with permutation-invariant structured prediction. In *Advances in Neural Information Processing Systems (NIPS), 2018*. 4

[18] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4555–4564, 2016. 3

[19] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013. 2

[20] J. Johnson, A. Gupta, and L. Fei-Fei. Image generation from scene graphs. *arXiv preprint arXiv:1804.01622*, 2018. 2

[21] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3668–3678, 2015. 2

[22] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 2

[23] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2

[24] L.-J. Li and L. Fei-Fei. What, where and who? classifying events by scene and object recognition. 2007. 3

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single Shot MultiBox Detector. In *European Conference on Computer Vision*, pages 21–37, 2016. 2

[26] C.-Y. Ma, A. Kadav, I. Melvin, Z. Kira, G. AlRegib, and H. P. Graf. Attend and interact: Higher-order object interactions for video understanding. *arXiv preprint arXiv:1711.06330*, 2017. 2

[27] V. Madhavan and T. Darrell. The bdd-nexar collective: A large-scale, crowdsourced, dataset of driving scenes. Master’s thesis, EECS Department, University of California, Berkeley, May 2017. 6

[28] P. Mettes and C. G. Snoek. Spatial-aware object embeddings for zero-shot localization and classification of actions. In *Proceedings of the International Conference on Computer Vision, 2017*. 1

[29] D. J. Moore, I. A. Essa, and M. H. Hayes. Exploiting human actions and object context for recognition tasks. Technical report, Georgia Institute of Technology, 1999. 3
B. Packer, K. Saenko, and D. Koller. A combined pose, object, and feature model for action understanding. In CVPR, pages 1378–1385. Citeseer, 2012. 1

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016. 2

S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015. 2

M. S. Ryoo and J. K. Aggarwal. Hierarchical recognition of human activities interacting with objects. In 2007 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2007. 3

A. Sanchez-Gonzalez, N. Heess, J. T. Springenberg, J. Merel, M. Riedmiller, R. Hadsell, and P. Battaglia. Graph networks as learnable physics engines for inference and control. arXiv preprint arXiv:1806.01242, 2018. 2

K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In Advances in neural information processing systems, pages 568–576, 2014. 1, 2

G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In European conference on computer vision, pages 140–153. Springer, 2010. 2

D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE international conference on computer vision, pages 4489–4497, 2015. 1, 2, 7

G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. IEEE transactions on pattern analysis and machine intelligence, 40(6):1510–1517, 2018. 2

H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pages 3169–3176. IEEE, 2011. 2

H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. International journal of computer vision, 103(1):60–79, 2013. 2

H. Wang and C. Schmid. Action recognition with improved trajectories. In Proceedings of the IEEE international conference on computer vision, pages 3551–3558, 2013. 2

X. Wang and A. Gupta. Videos as space-time region graphs. arXiv preprint arXiv:1806.01810, 2018. 2

X. Wang and A. Gupta. Videos as space-time region graphs. In ECCV, 2018. 2

P. Weinzaepfel, X. Martin, and C. Schmid. Towards weaklysupervised action localization. arXiv preprint arXiv:1605.05197, 3(7), 2016. 1

J. Wu, A. Osumtogun, T. Choudhury, M. Philipose, and J. M. Rehg. A scalable approach to activity recognition based on object use. In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pages 1–8. IEEE, 2007. 3

J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh. Graph r-cnn for scene graph generation. arXiv preprint arXiv:1808.00191, 2018. 2

B. Yao and L. Fei-Fei. Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(9):1691–1703, 2012. 3

J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4694–4702, 2015. 2

V. Zambaldi, D. Raposo, A. Santoro, V. Bapst, Y. Li, I. Babuschkin, K. Tuyls, D. Reichert, T. Lillicrap, E. Lockhart, et al. Relational deep reinforcement learning. arXiv preprint arXiv:1806.01830, 2018. 2