# Multi View Scene Graph Generation in Videos

Yichao Lu *
Layer6 AI
yichao@layer6.ai

Cheng Chang *
Layer6 AI
jason@layer6.ai

Himanshu Rai *
Layer6 AI
himanshu@layer6.ai

Guangwei Yu
Layer6 AI
guang@layer6.ai

Maksims Volkovs
Layer6 AI
maks@layer6.ai

## 1. Abstract

This paper outlines our approach for the Scene Graph Generation task of the Activity Net's Home Action Genome competition in CVPR 2021. Scene Graph Generation is an important task in computer vision aimed at improving the semantic understanding of the visual world. Previous works in constructing scene graphs in images have largely focused on pairwise models that lack global context information which can be crucially important to disambiguate complex scenes. To address this problem recent approaches have incorporated relationship graphs between objects. However, the structure of these graphs is pre-set ahead of time and not modified during training, so any introduced errors can propagate into the prediction stage and affect accuracy. In this work we incorporate information from videos as well as individual frames to improve performance for this task. We incorporate prior information from the egocentric views into our box classifier to improve the classification accuracy. We then propose to dynamically infer relationship graphs using a novel form of attention. Unlike previous approaches, we don't assume any pre-existing structure or order, and attend over all detected objects. Training the attention layers end-to-end enables the model to learn how to optimally extract contextual information for the target task. We combine this with relevant features extracted from a boosting model to obtain state of the art performance.

## 2. Introduction

Visual relationship detection [19, 13] is an important task in computer vision aimed at improving the semantic understanding of the visual world. While tasks such as recognition [2, 21, 7] and detection [12, 4, 18] focus on identifying and localizing objects, visual relationship detection additionally asks to predict relationships between pairs or groups of objects. Relationship prediction is a challenging task since it requires an in-depth understanding of a given scene and interactions between objects in it. However, it is also a major component of how we perceive and understand visual information [10], so successfully solving it would be a major step towards visually intelligent systems.

Visual relationship is formulated as a $\langle$*subject*, *predicate*, *object*$\rangle$ triplet, for example $\langle$*man*, *ride*, *horse*$\rangle$. Accurately detecting such relationships can be difficult due to the nuanced and often vague interactions between objects. Even seemingly straightforward relationships between easily recognizable objects such as *man* and *horse* can be difficult to identify in complex scenes.

Prior work in the this area has largely focused on images.Recent works like Action Genome [8] and Home Action Genome [16] offer annotated frames in videos across different views and hence provide extra signals which can be beneficial in better localization as well as scene graph construction. In this paper we build on these extra signals to improve our classification on exocentric(third person view). Further we use ensembling at different pyramid levels of classification to capture objects of different size.

Previous work in scene graph construction has largely focused on multi-stage pipelines [13, 27, 28, 32]. First, object detector is applied to identify all objects in a given image. Then relationship model is used to predict relationship predicates for pairs of detected objects. Relationship model is typically structured as pairwise prediction, and takes as input features from pairs of objects for which the relationship is predicted [13, 11]. Pairwise prediction has a significant drawback where by only focusing on a specific object pair the model can miss important global context information. This can lead to scene misinterpretation and incorrect prediction.

To address this problem recent work has explored ways to incorporate global contextual information into relationship prediction. The majority of proposed approaches in this category define a graph over detected objects, and prop-

*Authors contributed equally and order is determined randomly.

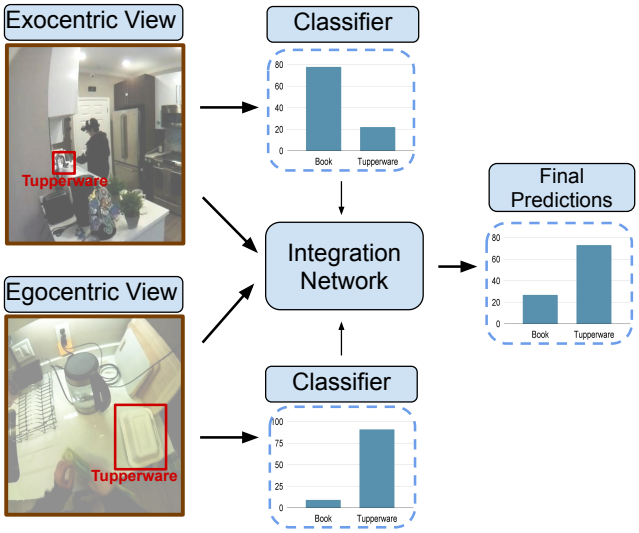



Figure 1: Pipeline for classification module.Individual ensemble of classifiers are applied on each views. The classifiers are applied on boxes in exocentric views while for egocentric view classifiers are applied on entire frame. The resultant softmax scores are then fed into Integration network along with features from both frames to get final predictions.

agate information through this graph to extract global context [28, 24]. The graph is pre-computed ahead of time as another stage in the pipeline following object detection, and is not modified after that. Incorporating information from the graph does provide additional context that can aid prediction. However, fixed structure can contain significant errors that the relationship model is unable to fix. Missing and/or incorrect vertices can propagate wrong information that can be amplified during the prediction stage.

In this work we take a different approach and instead *learn* a graph representation between objects by applying multiple layers of self-attention. Given an object, self-attention allows the model to focus on relevant other objects within the image. The process can be thought of as creating "soft" edges between objects in a directed weighted graph. Unlike previous approaches, we don't assume any pre-existing structure or order, and attend over all objects found during the detection phase. During training, the self-attention layers are optimised for relationship prediction, so the model learns how to optimally extract contextual information for the target task. We found that that a significant portion of performance gain is achieved by adding multi-head self-attention.

# 3. Related Work

## 3.1. Visual Relationship Detection

The visual relationship detection task is formalized first by Lu *et al.* [13]. In the current literature, this task is also treated as a related task to scene graph parsing, where a scene graph is the visually grounded graph with localized objects as nodes and pairwise predicates as the edges. In this case, the visual relationship detection task is constrained to generating valid scene graphs. This is referred to as the graph constraint where given an object pair, only one predicate class is predicted. We follow most existing work and investigate both tasks simultaneously by evaluating with and without the graph constraint.

Most work in this area follow the initial pipeline from [13] to apply a standard object detection pipeline with off-the-shelf fine-tuned weights to predict objects, followed by predicate classification [13, 33, 29, 27, 1, 30, 11, 23, 28, 25]. We follow this protocol to disentangle object detection error with relationship detection and focus on reasoning over the relationships. Many of these works then apply pairwise classification model similar to [13] which, given two proposed objects from the detection pipeline, independently make a prediction for each predicate class. Some exception include [31] which learn embeddings of objects and predicates and map their features to a shared space for inference. Among these Associative Embedding use a graph contrastive loss to train object embedding that capture graph information[15]. Recently, similar work improves over this by applying insights particular to visual relationships and introducing new loss functions to learn over both positive and negative relationship examples, pushing performance to new state-of-art[32]. Here we investigate an orthogonal direction and instead focus on learning global context. We show that self-attention is sufficient to capture the global context and achieve competitive performance using standard pairwise relationship classifier with cross entropy loss.

## 3.2. Context

Prior to visual relationship detection, the use of context information has been studied by numerous works in object understanding for recognition [6, 14] and detection [3, 20, 26]. context in generating sentence from image[5]. Since the formal definition of visual relationship detection by [13], recent works depart from this initial paradigm [15, 28, 24, 31] to incorporate contextual information. [28] show the importance of semantic prior and sub-structures that exist within a scene graph and propose to use a recurrent model (LSTM) to learn new embedding that captures the contextual information. Similarly, Graph R-CNN apply graph convolutional network (GCN) to capture contextual information [24]. However both rely on pre-defined structure (order in[28] and neighbor graph in [24]). In contrast,

we use Transformer[22] encoder's self-attention to learn how to capture the context.

### 3.3. Attention

Our work leverages the Transformer [22] self-attention mechanism from natural language processing. We adopt ideas such as scaled dot-product attention and multi-head attention. Prior work [24, 17] that apply attention in visual relationship detection start by defining a nearest neighbor graph. Attention is used to capture information about this graph structure by encoding it similar to graph convolutional network (GCN)[9]. In particular Graph R-CNN[24] adds attention to GCN and generates new embedding for each object. Here, the attention is a predetermined set of weights generated from pairwise similarity of the object features. Graph self-attention [17] embed a pair of object features and linguistic relationships jointly using attention mechanisms. However, their attention is also computed over a neighborhood graph similar to Graph R-CNN. In contrast, we use attention as a mechanism to directly extract useful information. Our model is based on self-attention without using pre-defined graph structure.

## 4. Approach

Visual relationship detection/scene graph construction is the joint task of detecting objects and relationships between them in the form of triplets $\langle$*subject*, *predicate*, *object*$\rangle$. Both *subject* and *object* are objects in the traditional detection setting and share a common set of object classes, while *predicates* are represented by a separate set of relationship classes. Our approach, called **C**lassify, **A**ttend and **P**redict (CAP), aims to dynamically infer relationship graph between objects by applying self-attention. Information from this graph is then encoded into representation of each object. Updated representations contain relevant context information, and are used to predict relationships between objects.The full architecture for our approach is shown in Figure 2.

Given a video and $n$ labelled objects , in the annotated frames, the task is to correctly label the boxes and identify relationship between the human subject and the object, if there exists one. We use $[x_1, ..., x_n]$ to denote the given object bounding boxes where each $x_i \in \mathsf{R}^d$ is represented by a set of features. Following the classification stage , we apply multiple layers of self-attention to encode relevant global context information. Each self-attention layer can be thought of as inferring a directed weighted graph between objects with edge weights given by the attention softmax coefficients. Graph construction is dynamic, and the model learns what information to focus on during the training phase. This is in contrast with previous work where graph is prebuilt in a separate stage and kept fixed during model training [28, 24]. After self-attention, updated representations $[\hat{x}_1, ..., \hat{x}_n]$ are passed to the pairwise relationship classifier to get predicate probability. These predicated probabilities are then fed into a gradient boosting model that also accepts features from the raw boxes and video features extracted from a 3d CNN model. In the following sections we describe each stage in detail.

### 4.1. Box Classification in Multi Views

We follow the existing literature in object classification to predict the labels for objects in both egocentric (first person view) as well as exocentric(third person view). We trained individual classifiers first on each of the views. As shown in figure 2, the classofier trained on exocentric views could have a difficult time in identifying tiny and/or obfuscated objects. Additionally side views in exocentric views might provide additional useful information for classification. So a joint classification module as shown in figure 2 learns from both the views and fixes the mistakes from each of the individual views. We use an ensemble of classifiers for each of the views. We then take the SoftMax scores from both the classifiers and pass them through our integration network. The integration network comprises of a bunch of mlp layers with additional non-linearity. The integration network also looks at the image features from both the images .It thus learns from both the views and can classify objects with a much higher success rate. The final confidence from this model as well as the spatial visual and semantic features extracted from the boxes are fed into the relationship module.

**Visual Features.** Visual features are taken from the CNN classifiers . These features capture visual information about the object by encoding part of the image that corresponds to object's bounding box from the exocentric view as well as entire image from the egocentric view. We pool the features to have consistent size to be fed in the relationship modules.

**Spatial Features.** Spatial features capture geometric information of the object within an image. Relationships such as *on*, *under* and *inside_of* are highly dependent on object positioning in a given scene. To incorporate this information we compute features such as coordinates of the bounding box, its height and width, total area etc.

**Semantic Features.** Recent work found that semantic features generated from statistics between object classes and predicates can significantly improve model performance [28, 32]. . When limited training data is available, incorporating prior information on object-predicate co-occurrence can both accelerate training and make the model more robust. We use features such as the predicted object class and statistics on predicates that appear with this class, to summarize relevant semantic information.

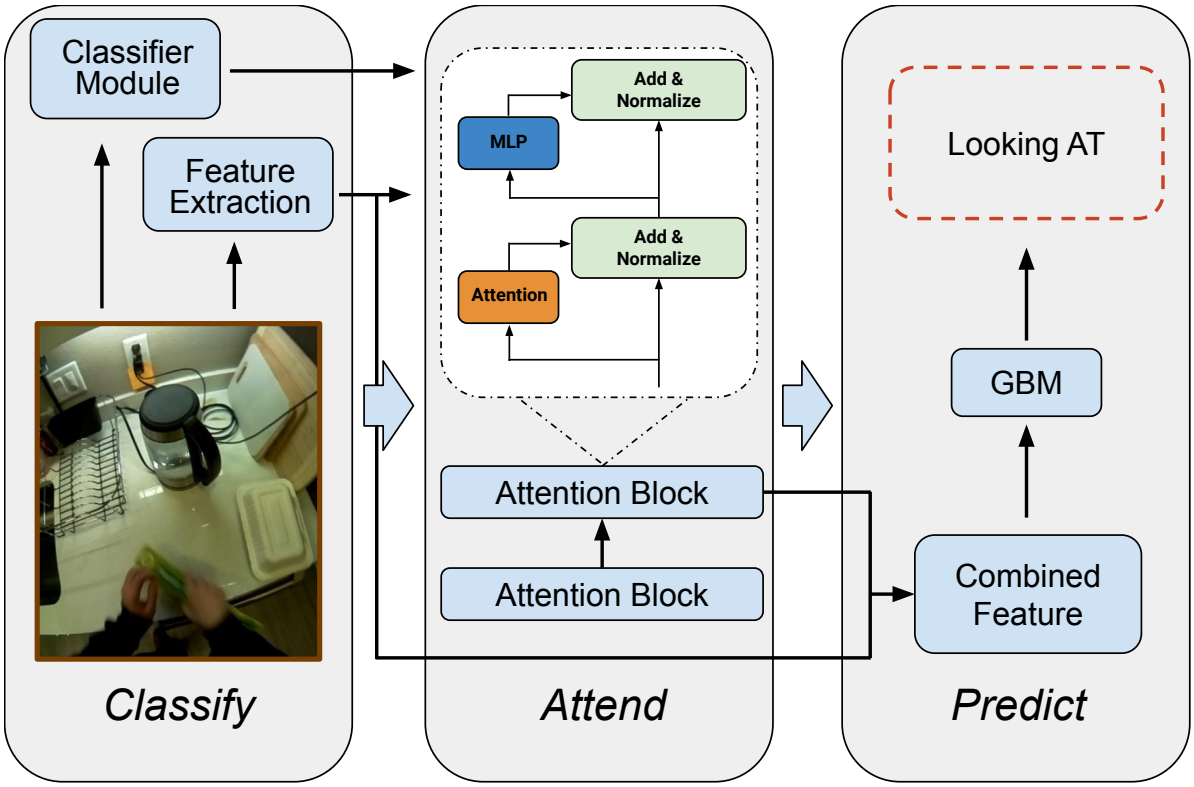


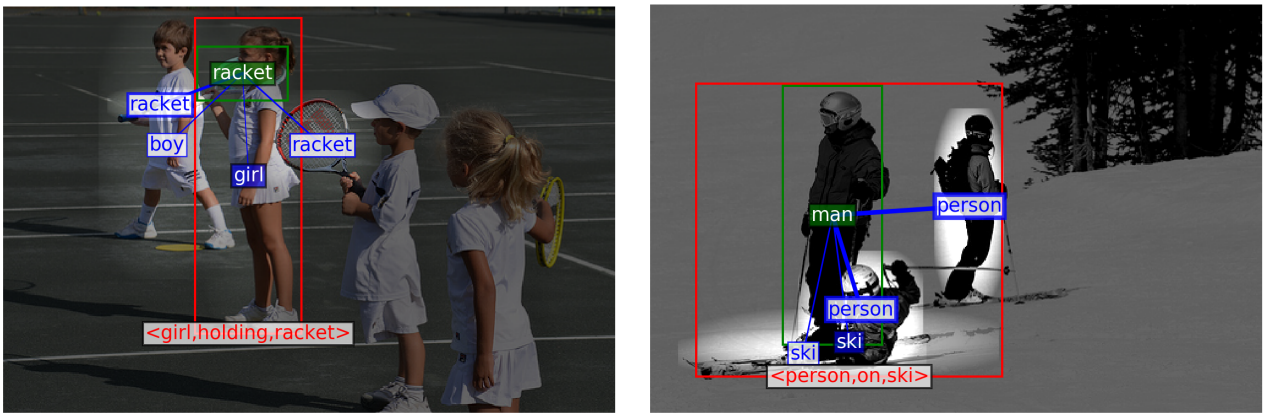

Figure 2: **Classify**, **Attend** and **Predict** model architecture. **Classify**: object classification is done on bounding boxes by running classifier module on exocentric and egocentric views. Visual, spatial and semantic features are extracted for each bounding box to get a representation . **Attend**: multiple layers of multi-head self-attention are applied to get updated representations. Each self-attention layer dynamically creates a weighted directed graph between detected objects with edge weights given by attention softmax coefficients. **Predict**: updated representations are passed to the relationship classifier to predict relationship class for each candidate pair of objects. These results along with other extracted features from I3D as well as other features are passed to XGB model for final predictions

### 4.2. Global Context Through Self-Attention

The majority of proposed models for visual relationship detection operate on pairs of objects found during the detection stage. Object features are generated independently of each other, so the pairwise model is only able to capture information local to the two objects. As we discussed, such architecture has a significant drawback where by focusing only on specific object pair, the model loses global context information. This is particularly problematic in cluttered scenes where many objects are in close proximity to one another and have common relationships. Pairwise models are unable to jointly reason about nearby objects, and this can lead to incorrect predictions. Zhang et al., [32] found that this is the predominant mode of failure for many such models.

We address this limitation in the attend stage. The main idea behind this stage is to treat all detected objects as global context information. However, given a target object, only part of this context is generally relevant for relationship prediction. Consequently, the model also needs to selectively focus on specific objects within the context to extract the salient information. We propose to achieve this by using multiple layers of self-attention. Each self-attention layer is applied to all detected objects, and softmax attention weights determine which objects contain useful information for relationship prediction. This information is then aggregated together and encoded into object representation. Updated representations contain both information about the object, and the relevant context around it. Stacking multiple layers of self-attention enables the model to encode increasingly more complex interactions between objects.

A number of attention architectures have been developed, in this work we focus on the recently proposed Transformer encoder with multi-head self-attention [22]. Using $X \in \mathsf{R}^{n \ d}$ to denote the concatenated together representations from all objects $[x_1, ..., x_n]$, Transformer self-attention maps $X$ to a new representation $\hat{X}$ of the same




(a) (b)

Figure 3: Example self-attention graphs inferred by our model. In each figure, edges with largest attention weights are shown for a target detected object (green box). Edges are indicated as blue lines with thickness is proportional to softmax weights. Softmax activations are taken from one head in the first attention block. The predicted relationship triplet for each target object is shown in red.

size and order. Formally, this mapping is computed as:

$$\begin{aligned} \hat{X} &= \text{LayerNorm}\Big(\text{MLP}(Z) + Z\Big) \\ Z &= \text{LayerNorm}\Big(\text{MultiHead}(X) + X\Big) \\ \text{MultiHead}(X) &= \big(\text{head}_1||...||\text{head}_h\big)W^O \\ \text{head}_i(X) &= \text{softmax}\left(\frac{(XW_i^Q)(XW_i^K)^\top}{\sqrt{d}}\right)\left(XW_i^V\right) \end{aligned} \tag{1}$$

where $W_i^Q, W_i^K, W_i^V \in \mathsf{R}^{d\ \ d}$ and $W^O \in \mathsf{R}^{hd\ \ d}$ are the learned weights for multi-head self-attention with $h$ heads, $||$ denotes concatenation and MLP is a feed forward layer. Similar to the Transformer encoder [22], we apply residual connections after attention and MLP operations, and set all layer dimensions to $d$. The dimensionality can be readily changed by applying another MLP projection. For each attention head, global context is presented by the linearly transformed object representations $XW_i^V$. Softmax attention then enables the model to focus on specific objects within this context. To draw parallels with previous work on graph-based relationship prediction, the $n \times n$ attention tensor can be thought of as an adjacency matrix for a weighted densely connected relationship graph. Information from this graph is encoded into the representation of each object by combining vertex representations with softmax edge weights. During training the model learns how to effectively focus on relevant objects in each scene to maximise relationship prediction accuracy.

### 4.3. Relationship Prediction

Our relationship prediction module consists of two stages. In the first stage we use an attention based model to predict relationship scores between a subject and an object. In the second stage these scores along with other features described in the previous section as well as I3D features extracted from video are fed into an gradient boosting tree to make the final predictions. We now describe these stages in details below. Following the attention stage, we apply pairwise prediction to infer relationships for each pair of objects. In contrast to existing pairwise models, we use the updated representations $\hat{x}_i$ that incorporate global context information. We define the probability of a relationship triplet $\langle x_i, k, x_j \rangle$ as:

$$p(x_i, k, x_j) = p(k|\hat{x}_i, \hat{x}_j)p(x_i)p(x_j) \tag{2}$$

where $p(k|\hat{x}_i, \hat{x}_j)$ is the probability of predicate class $k$ given updated representations $\hat{x}_i$ and $\hat{x}_j$, and $p(x_i)$, $p(x_j)$ are object class probabilities from the detection model. This model thus favours predictions where both object and predicate probabilities are high.

There are many possible architecture choices for $p(k|\hat{x}_i, \hat{x}_j)$, in this work we use an MLP with a softmax output layer that generates probabilities for all relationship predicate classes. This model takes as as input concatenated representations from $\hat{x}_i$ and $\hat{x}_j$, together with additional pairwise spatial and semantic features. The features introduced in Section 4.1 are extracted for each object individually, and can be enhanced when pairs of objects are considered. For example, for spatial features we can now compute distance between corresponding bounding boxes

and their overlap. Similarly, we can narrow down possible relationship predicates given the two object classes and estimate their likelihoods from training data counts. These features provide additional prior information that can aid prediction, and we pass them to the pairwise classifier together with updated object representations.

During training we jointly optimize attention and pairwise classifier weights. For each training image we first pass it through the classification model to get features for all ground truth objects and their bounding boxes. We then use the target relationship triplets $\langle x_i, k, x_j \rangle$ to optimize the model with the log likelihood objective:

$$\mathcal{L} = -\sum_{hx_i;k;x_j i} \log\left(p(k|\hat{x}_i, \hat{x}_j)\right) \quad (3)$$

The gradients from this loss are back-propagated to jointly update the classifier and self-attention layers.

At inference, we first use the classification model to get scores for the bounding boxes. We then extract visual, spatial and semantic features for each object, aggregate them together and pass through self-attention layers to get updated representations. Finally, the pairwise relationship classifier is applied to all object pairs to get predicate class probabilities. These class probabilities are then passed onto an XgBoost Classifier.

Now we describe our approach to incorporating frame-level and video-level representations into the relationship prediction module. We begin by fine-tuning the Kinetics pre-trained I3D on the Home Action Genome dataset for the human action recognition task. Then we obtain frame-level and video-level representations by feeding each video into the trained I3D. The frame level representations are obtained from the 3D ConvNets in both streams, and the video-level representations are the pre-softmax logits from the penultimate layer of I3D.

The frame-level and video-level representations, together with the relationship prediction scores predicted by the self-attention network, are then fed into the second stage XGBoost to obtain the final relationship prediction. We additionally add several manually designed features to XGBoost, including (1) spatial features that encode the bounding box coordinates of subjects and objects, their relative position, IOU, etc., and (2) semantic features that encode the frequency of subjects and objects and the frequency of each predicate given a specific subject-object pair.

Compared to conventional scene graph generation datasets, where the predictions are evaluated using individual subject-predicate-object triplets, the Home Action Genome dataset requires the model to accurately identify each possible predicate between subject-object pairs. Specifically, each subject-object pair may have more than one predicates and even sets of predicates, and the model needs to correctly predict all these predicates in order to obtain full score.

Table 1: Results on the Scene Graphh Generation Task in Home Action Genome

| LeaderBoard - Top3 Teams | | | |
|---|---|---|---|
| Team | Score | recall@10 | recall@20 |
| IMBA | 0.76569 | 0.72183 | 0.80955 |
| Layer6 | 0.68437 | 0.63398 | 0.73476 |
| AIST & DENSO | 0.65797 | 0.59636 | 0.71958 |

The predicted relationship triplets are sorted according to the joint object-predicate probability $p(x_i, k, x_j)$ (see Equation 2) and top triplets are used as the final prediction.

## 5. Evaluation and Results

We evaluated the model on the code provided by the competition hosts and present the results in the table 1 on the Home Action Genome Dataset [16]. We present the recall@10 and recall@20 results.The training set comprises of 2624 videos comprising of 119,919 annotated frames having over 1 million annotated relationships.There are 25 different relationship classes and 85 different object classes in the dataset. The evaluation metric was PRDCLS which removes the detector bias allowing us to focus specifically on relationship classification.

## 6. Conclusion

We described our winning entry for the Home Action Genome competition 2021, where we proposed a highly effective and novel pipeline for the task of scene graph generation in videos.

## References


[1] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *CVPR*, 2017. 2

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[3] Santosh K Divvala, Derek Hoiem, James H Hays, Alexei A Efros, and Martial Hebert. An empirical study of context in object detection. In *CVPR*, 2009. 2

[4] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 1

[5] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV*, 2010. 2

[6] Carolina Galleguillos and Serge Belongie. Context based object categorization: A critical survey. *Computer vision and image understanding*, 114(6):712–722, 2010. 2

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[8] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10236–10247, 2020. 1

[9] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ICLR*, 2016. 3

[10] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017. 1

[11] Yikang Li, Wanli Ouyang, Xiaogang Wang, and Xiao'ou Tang. Vip-cnn: Visual phrase guided convolutional neural network. In *CVPR*, 2017. 1, 2

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1

[13] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *ECCV*, 2016. 1, 2

[14] Thomas Mensink, Efstratios Gavves, and Cees GM Snoek. Costa: Co-occurrence statistics for zero-shot classification. In *CVPR*, 2014. 2

[15] Alejandro Newell and Jia Deng. Pixels to graphs by associative embedding. In *NeurIPS*, 2017. 2

[16] Jingwei Ji Rishi Desai Kazuki Kozuka Shun Ishizaka Ehsan Adeli Juan Carlos Niebles Nishant Rai, Haofeng Chen. Home action genome: Cooperative compositional action understanding. 2021. 1, 6

[17] Mengshi Qi, Weijian Li, Zhengyuan Yang, Yunhong Wang, and Jiebo Luo. Attentive relational networks for mapping images to scene graphs. In *CVPR*, 2019. 3

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1

[19] Mohammad Amin Sadeghi and Ali Farhadi. Recognition using visual phrases. In *CVPR*, 2011. 1

[20] Ruslan Salakhutdinov, Antonio Torralba, and Josh Tenenbaum. Learning to share visual appearance for multiclass object detection. In *CVPR*, 2011. 2

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3, 4, 5

[23] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, 2017. 2

[24] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *ECCV*, 2018. 2, 3

[25] Guojun Yin, Lu Sheng, Bin Liu, Nenghai Yu, Xiaogang Wang, Jing Shao, and Chen Change Loy. Zoom-net: Mining deep feature interactions for visual relationship recognition. In *ECCV*, 2018. 2

[26] Ruichi Yu, Xi Chen, Vlad I Morariu, and Larry S Davis. The role of context selection in object detection. In *BMVC*, 2016. 2

[27] Ruichi Yu, Ang Li, Vlad I Morariu, and Larry S Davis. Visual relationship detection with internal and external linguistic knowledge distillation. In *ICCV*, 2017. 1, 2

[28] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, 2018. 1, 2, 3

[29] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In *CVPR*, 2017. 2

[30] Hanwang Zhang, Zawlin Kyaw, Jinyang Yu, and Shih-Fu Chang. Ppr-fcn: Weakly supervised visual relation detection via parallel pairwise r-fcn. In *ICCV*, 2017. 2

[31] Ji Zhang, Yannis Kalantidis, Marcus Rohrbach, Manohar Paluri, Ahmed Elgammal, and Mohamed Elhoseiny. Large-scale visual relationship understanding. In *AAAI*, 2019. 2

[32] Ji Zhang, Kevin J Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph parsing. In *CVPR*, 2019. 1, 2, 3, 4

[33] Bohan Zhuang, Lingqiao Liu, Chunhua Shen, and Ian Reid. Towards context-aware interaction recognition for visual relationship detection. In *ICCV*, 2017. 2