Abstract

State of the art visual relation detection methods have been relying on features extracted from RGB images including objects’ 2D positions. In this paper, we argue that the 3D positions of objects in space can provide additional valuable information about object relations. This information helps not only to detect spatial relations, such as standing behind, but also non-spatial relations, such as holding. Since 3D information of a scene is not easily accessible, we propose incorporating a pre-trained RGB-to-Depth model within visual relation detection frameworks. We discuss different feature extraction strategies from depth maps and show their critical role in relation detection. Our experiments confirm that the performance of state-of-the-art visual relation detection approaches can significantly be improved by utilizing depth map information.

1 Introduction

Relational learning is an established research area in machine learning. State-of-the-art approaches [16] describe relations as triples of the form (subject, predicate, object), such as (Man, rides, Bike). The triples, which all together form a knowledge graph, are typically extracted from structured data such as the infoboxes in Wikipedia and other sources.

In the last years, the detection of relations from images, i.e., visual relation detection, has also gained a lot of interest. The reason is that understanding relations between entities can play an important role in decision making. For example, detecting whether a man is on a bike or next to a bike is a crucial issue in autonomous driving. State-of-the-art works in this area utilize features extracted from RGB images, including their relative 2D positions, and propose different models to capture their relations. In this paper, we argue that relation detection can additionally benefit from 3D information. This information can help to distinguish between many relations such as standing behind, standing in front and even improve detection in situations where the objects are nearby such as standing next to.

Unfortunately, most available datasets lack this 3D information since acquiring them is a cumbersome task requiring specialized hardware. As depth maps can provide the objects’ distance from the camera, in this paper, we propose to incorporate a pre-trained RGB-to-Depth
network within relation detection frameworks. In particular, we exploit existing knowledge learned about the correspondences between RGB images and depth maps from separate large corpora of RGB-D pairs, i.e., NYU-Depth-v2 dataset [14].

In the next step, we extract features from the depth maps, which, jointly with the RGB-features, are the basis for relation detection. Depth maps have already been widely employed in other tasks, e.g., image classification and segmentation. In these applications, it is common to simply encode a depth image as a rendered RGB image and extract features using a pre-trained convolutional neural network, such as VGG [19]. However, this might be sub-optimal, as these are two modalities with different information. Indeed, in our experimental results in Section 4.5 we show that only if the feature extraction network had particularly been trained on depth maps, visual relation detection can be improved.

In summary, our contributions are as follows:

1. We are the first to utilize 3D information in visual relation detection. To compensate for the lack of 3D data, we propose to incorporate a pre-trained RGB-to-Depth model within relation detection frameworks.
2. We discuss and empirically investigate different strategies to extract features from depth maps for relation detection.
3. We study the quantitative and qualitative benefits of incorporating depth maps. We show in our empirical evaluations using the VRD dataset [12], that models using depth maps can outperform competing methods by a margin of up to 3% points, even those using information extracted from external language sources.

2 Related Works

Knowledge Graph Learning In Knowledge Graph learning, the aim is typically to find embeddings or latent representations for entities and predicates, which then can serve to predict the probability of unseen triples. These methods mostly differ in how they model relations. In RESCAL [12] each relation is defined as a linear transformation in the embedding space of entities, producing a triple probability. TransE [3] employs a similar idea but limits each relation to a translation. In comparison to RESCAL, it has fewer parameters; as a disadvantage, it cannot model symmetric relations. DistMult [22] considers each relation as a vector, similar to TransE, but minimizes the trilinear dot product of subject, predicate
and object vector. DistMult can also be understood as a form of RESCAL, where the transformation matrix is diagonal. ComplEx \[21\] extends DistMult to complex-valued vectors of embeddings. A multilayer perceptron (MLP) architecture \[4\] extends these methods to non-linear transformations and has shown to be competitive to the other discussed approaches on most benchmarks \[16, 20\].

**Visual Relation Detection** Visual relation detection received a huge boost by the availability of large corpora of annotated images such as the Visual Relation Detection (VRD) \[12\] and the Visual Genome \[10\], containing the visual form of entities, and relations. In VRD \[12\], Word2Vec representations of the subject, object, and the predicate were used to train a model jointly with the corresponding image section describing the predicate. In particular, they consider the joint bounding box of subject and object as the image representation for the predicate. Follow-up work achieved improved performance by incorporating a knowledge graph, constructed from the annotated triples in the training set \[1\]. In general, the distribution of the predicate bounding box in these works is much more long-tailed than of objects alone. Therefore, separating the models for objects and predicates, as employed in VTransE \[26\] reduces the complexity of training such a model. VTransE is a generalization of TransE to visual relation detection in which the last convolutional layer of the image detector, together with the location of entities and their class labels, is taken as the input vector to the TransE algorithm. More recently, Yu et al. \[24\] proposed a teacher-student model to distill external language knowledge to improve visual relation detection. Neural Motifs \[25\] and Graph R-CNN \[23\] incorporate context within each prediction using LSTMs and graph convolutions respectively.

**Depth Maps** While several works have leveraged depth maps to improve object detection \[2, 5, 7\], to the best of our knowledge this is the first time that depth maps are used in the relation detection task.

### 3 Framework

Let \(\xi = \{e_1, e_2, ..., e_n\}\) be the set of all entities, including subjects (s) and objects (o), and \(P = \{p_1, p_2, ..., p_m\}\) the set of all predicates. Each occurring entity \(e_i\) is represented by an image \(I_i\), depth map \(D_i\) and a class label \(c_i\). From the RGB images and depth maps, we extract features denoted by \(v_i\) and \(d_i\). In the following, we will describe the depth map generation, feature extraction and the relation model.

#### 3.1 Depth Map Generation

We incorporate an **RGB-to-Depth** model within our visual relation detection framework. As shown in Figure 2, this is a fully convolutional neural network (CNN) that takes an RGB image as input and generates its predicted depth map. This model can be pre-trained on any datasets containing pairs of RGB and depth maps. It enables us to work with currently available visual relation detection datasets without requiring to collect additional data, and also mitigates the need for specialized hardware in real-world applications.
Figure 2: We propose utilizing 3D information in visual relation detection by incorporating an RGB-to-Depth model within relation detection frameworks. On the left side, we see the RGB image (top) and its transformation into the corresponding depth map (bottom) by the RGB-to-Depth network. The extracted object bounding boxes from the region proposal network and their depth maps are fed into separate feature extractors. RGB features ($v$) and depth features ($d$) of each subject entity and object entity together with their location features ($l$) and class labels ($c$) are fed into fully connected relation detection layers to estimate the predicate. The color opacity of arrows for object are decreased to improve the readability.

3.2 Feature Extraction

**Feature Extraction from RGB ($v_i, c_i, l_{ij}$):** Extracting features from RGB images follows a typical approach of employing a CNN pre-trained and fine-tuned for image classification, and extracting the object bounding boxes, embeddings from the last convolutional (or fully connected layer), and the final classification output. In particular, $(x_i, y_i, w_i, h_i)$ is the coordinates of the bounding box for $e_i$, $v_i$ is the embedding feature vector, and $c_i$ is the class probability distribution.

We define a scale-invariant location feature for each pair of entities $e_i$ and $e_j$ similarly to [26] as $l_{ij} = (t_x, t_y, t_w, t_h)$ with: $t_x = (x_i - x_j) / w_j, t_y = (y_i - y_j) / h_j, t_w = \log(w_i / w_j), t_h = \log(h_i / h_j)$.

**Feature Extraction from Depth Maps ($d_i$):** Depth maps have been employed in tasks such as object detection and segmentation [5, 8]. In those works, it is common to simply render a depth map as an RGB image and extract depth features using a CNN pre-trained for RGB images. There, it has been argued that the edges in depth maps might yield better object contours than the edges in cluttered RGB images and that one may combine edges from both RGB and depth to obtain more information [8]. Therefore, they aimed to get similar, complementary features from both modalities.

However, this practice of employing a model pre-trained on a particular source modality, e.g. RGB, and applying it on a different target modality, e.g. depth map, is sub-optimal in many applications. Specifically, in relation detection, one of the main reasons to employ depth maps is utilizing the relative 3D distance of objects to each other. The objects in an image are often spread over wider regions of the image while the early processing layers in object detector CNNs, trained with RGB images, are highly specialized in differentiating

---

1One should also keep in mind that even fine-tuning some layers of a network does not change the very early convolutional filters.
pixels in a local neighborhood. Therefore, these networks might lack sufficiently many non-differentiating filters in the first layer, required in order to carry depth information up to higher layers. To this end, unlike most of the state-of-the-art works on depth-based image segmentation or classification, we train a depth map feature extractor model from scratch, using depth maps and specifically for relation detection. In Section 4 we show that empirical investigations of this effect support our proposal.

Note that depth map features were extracted from the same bounding boxes as the ones generated for the RGB-images. To compose the final entity representation, we create a full feature vector $\text{feat}_i$ for each entity by concatenating $v_i, d_i, c_i$ and $l_{ij}$.

3.3 Relation Model

We define a latent representation for the subject $e_s$ and for the object $e_o$ to embed the features of each entity in a joint space. The subject (and similarly object) embeddings and their interactions are modeled as follows:

$$
e_s = f(W[\text{feat}_s]),$$

(1)

where $\text{feat}_s = [l_s, v_s, c_s, d_s]$. The predicate is then modeled as

$$
e_p' = f'(W'[e_s; e_o]).$$

(2)

Here, $e_s, e_o$ are embedding vectors for subject and object, and $e_p'$ is a vector of logits for the predicate. $W$ and $W'$ describe linear transformations, $f$ and $f'$ are non-linear functions. We realize the latter as layers in a neural network with ReLU activations.

The predicate prediction model of Equation (2) is a generalization of the model used by [4] to construct knowledge graphs. In that paper, latent features for the predicates are also part of the network input, whereas here we use a separate output for each predicate which is easier to train and has fewer parameters.

To learn the parameters, we consider each relation $(s, p, o)$ with an associated Bernoulli variable $y_{spo}$ that takes 1 if the triple is observed and 0 otherwise. Given the set of observed triples $T$, the loss function is the categorical cross entropy between the one-hot targets and the distribution obtained by softmax over the network’s output defined as:

$$
\mathcal{L} = \sum_{(s, p, o) \in T} -\log \frac{\exp(e_p^T f'(W'[e_s; e_o]))}{\sum_{p' \in P} \exp(e_{p'}^T f'(W'[e_s; e_o]))}
$$

(3)

Here, we are following a locally closed world assumption [4].

4 Evaluation

4.1 Architectures

RGB-to-Depth Network: We use the RGB-to-Depth model introduced in [11]. The model is pre-trained on data from NYU Depth Dataset v2 [14]. Pre-training on the outdoor images from Make3D dataset [18] instead, did not show promising results. This observation is not surprising since Make3D images contain mostly outdoor scenes with too few objects.

RGB Feature Extraction: To extract embeddings and class probabilities of RGB images, we use the VGG-16 architecture [19] pre-trained on ImageNet [17] and fine-tuned to our data.
**Depth Map Feature Extraction:** For depth map extraction we use a convolutional neural network with the architecture proposed in [5]. We trained this model from scratch following the earlier discussions in Subsection 3.2. In the experimental section (Subsection 4.5), we show that this leads to much better solutions than using pre-trained convolutional layers. We trained this network on a pure depth-based, relation detection task using Adam [9], with a learning rate of $10^{-4}$ and batch size of 16 for eight epochs.

**Relation Detection Network:** Finally, given the features extracted from previous models with the location features described in Subsection 3.2, we trained our relation detection model. We concatenated the VGG features, depth features, one-hot encoded labels and location features in a pairwise fashion for subject and object. We connected each feature pair with a fully connected hidden layer of 64, 200, 4096 and 20 neurons with the dropout of 0.1, 0.8, 0.8 and 0.1 and a scaling layer initialized with 1.0, 0.3, 0.5 and 1.0 as in [24], in the same order as the mentioned features. In the end, we fully connected the concatenated output of these layers to 4096 neurons with 0.2 dropout before connecting it to 70 neurons for each predicate. We trained this network by Adam [9], with a learning rate of $10^{-3}$. We used a batch size of 16 and six epochs of training. All of the layers were initialized with Xavier weights [6].

### 4.2 Dataset

We test our approach on the Visual Relation Detection (VRD) [12] dataset. It contains 100 objects categories and 70 predicates. It has 37,993 triples from which 6,672 are unique. Similar to other works [1, 12, 26], we split the data into 4000 training and 1000 test images, where 1,877 relationships are in the test set for zero-shot evaluations.

### 4.3 Metrics

Some relations might not be annotated in the test set while because of the model’s generalization, they might get higher prediction values than the annotated ones. Therefore, we report R@K where it tells us whether the specific predicate in test set ended up as one of the top K listed probable predicates [1, 12, 24]. It is important to note the multilabel nature of predicates and that multiple predicates can describe the relation between an entity pair. For example, the correct predicate, given entity pair (Man, Horse) could be at the same time above and riding. Therefore, given each subject and object pair, we do not only consider the predicate with the highest prediction score but also the ranked prediction scores of all possible 70 predicates as in [24].

### 4.4 Comparing Methods

We compare our results with earlier approaches of Lu's-V [12], which takes the joint bounding boxes of subject and object as the predicate’s image and applies an image classifier on it, Lu’s-VLK [12] that combines the previous approach with Word2Vec [13] embeddings, and Baier et al. [4] that constructs a knowledge graph with ComplEx [21] model using class labels. We also compare with VTransE [26] that takes visual embeddings and refines them using TransE. For a fair comparison, we implemented a basic version of the last two methods rather than reporting the results directly. The other reported results are from [24]. We consider their student network (Yu’s-S) which is trained given images of the VRD, and their full model (Yu’s-S+T) which uses also external language data from Wikipedia.
Table 1: Predicate prediction and zero-shot accuracies on VRD test set divided into three sets of rows. The first set contains results directly reported from the respected works. The results in the lower two rows of sets are reported from self-implementations. When the depth maps are utilized together with RGB features \((\text{Ours-c}_i, \text{v}_i, \text{l}_{ij}, d_i)\), we gain a large improvement. This improvement is almost as large as \(\text{Ours-c}_i\) to \(\text{Ours-c}_i, \text{v}_i, \text{l}_{ij}\), demonstrating the importance of depth features in relation detection. One can also see that even using depth maps alone \((\text{Ours-d}_i)\) gives a surprisingly significant detection accuracy. Additionally, comparing \(\text{Ours-c}_i, \text{v}_i, \text{l}_{ij}\) to VTransE reveals the model’s advantage.

| Task | Zero-shot Pre. | Predicate Pred. |
|------|----------------|-----------------|
|      | R@100 | R@50 | R@100 | R@50 |
| Lu’s-V [12] | 32.34 | 23.95 | 37.20 | 28.36 |
| Lu’s-VLK [12] | 50.04 | 29.77 | 84.34 | 70.97 |
| Yu’s-S [24] | 74.65 | 54.20 | 86.97 | 74.98 |
| Yu’s-S+T [24] | - - | - - | 94.65 | 85.64 |
| ComplEx [1] | 74.42 | 52.78 | 81.02 | 62.45 |
| VTransE [26] | 83.66 | 67.15 | 96.22 | 90.00 |
| Ours - d_i | 72.80 | 52.27 | 86.61 | 74.17 |
| Ours - l_i | 81.69 | 62.70 | 95.39 | 88.22 |
| Ours - c_i, v_i, l_{ij} | 84.94 | 70.06 | 96.12 | 90.34 |
| Ours - c_i, v_i, l_{ij}, d_i | 87.43 | 72.28 | 96.54 | 90.47 |

To provide an ablation study, we report our relation prediction results in several settings: (1) When only class labels from images are available \((\text{Ours-c}_i)\). These results are comparable to ComplEx. (2) When RGB features are also utilized \((\text{Ours-c}_i, \text{v}_i, \text{l}_{ij})\). These are comparable to VTransE and Yu’s-S. (3) When depth maps are also available \((\text{Ours-c}_i, \text{v}_i, \text{l}_{ij}, d_i)\). To see the influence of using depth maps, one can compare the last two settings. (4) \(\text{Ours-d}_i\) is the results when using only the depth values and no image or label information.

### 4.5 Experiments

In this section, we describe the experimental settings. Each experiment is accompanied by a discussion to analyze the results.

**Predicate Prediction** Our main goal is to show the relation detection performance with depth maps. Therefore we do not focus on improving the object detection accuracy and report predicate prediction results. In this setting, the relation detection performance is isolated from the object detector’s error by using ground truth bounding boxes of the entities and their class labels to predict the most likely predicates.

Some triples e.g. \((\text{Man}, \text{rides}, \text{Horse})\) might appear in both test and training sets (with a differently appearing man and horse) while some triples in test set might have never been observed during training at all. To evaluate the ability of our model in generalizing to such unseen triples, we also report *zero-shot predicate prediction* where we evaluate the prediction accuracy of relations that are never seen during the training.

**Discussion:** The results are shown in Table 1. The upper part of the table demonstrates the results directly reported from those works while the lower two parts present the results of our own implementations. We can see that our full model achieves the highest accuracy in comparison to the others in all settings. It is also interesting to note that when using only depth
Figure 3: Using depth maps improves the understanding of *perspective* and increases the detection rate of relations such as (Tower, taller, Trees) even under noisy conditions such as in the left image. On the other hand, detecting the relation (Person, stand behind, Person) in the right image is not improved. One should note that here the athletes have similar depth values, which is more commonly observed in other relations such as *next to*. It might be difficult for the model to learn this case, unless supportive samples are presented during training.

maps we can already achieve a significant accuracy in predicate prediction, emphasizing on the value of relational information stored within a depth map. As will be shown later, gaining this improvement would not have been possible without cautious employment of the feature extractor.

To get a better intuition of the improvements that using different types of information bring, we plotted the top 10% absolute changes in prediction accuracy for each predicate in Figure 5. The top plot shows the performance changes from Ours-\(c_i\) to Ours-\(c_i, v_i, l_{ij}\). The lower plot shows the changes from Ours-\(c_i, v_i, l_{ij}\) to Ours-\(c_i, v_i, l_{ij}, d_i\). When depth maps are included, predicting some of the predicates such as *across*, *sleep on* and *taller than* gains a large improvement while some of the predicates such as *lying on*, *look* and *stand behind* either get worse or stayed the same. Specifically, we expected to have improved accuracy in detecting relations such as stand behind, when using depth maps while we see no improvements. To study the possible reasons for this failure, we examined the test images. One of the test images containing the predicate *stand behind* is shown in Figure 3, on the right. Here, the annotations indicate that (Person, stand behind, Person) while based on the corresponding depth map, they are in the same depth. This effect might appear in many instances within the dataset, leading to worse performances when dealing with such predicates. It is important to note that humans describe a predicate from their point of view which is not necessarily associated with the camera’s point of view. A simple way to overcome such problems is by having a richer set of data. On the other hand, Figure 3 shows the relation (tower, taller than, trees) on the left, which was correctly detected after including the depth maps. In general, as shown in Figure 4, the accuracy of relations including the predicate of taller than, has been improved. As sometimes the large distant objects might appear smaller than the nearby smaller objects, understanding the relations with this predicate requires a good understanding of perspective, provided by the depth maps. The imprecisions within the generated depth map, e.g. sky and reflective objects such as glasses, are inevitable as there are no ground truth depth maps available.
Figure 4: This plot shows the top 10 percent absolute changes in prediction performance per predicate, comparing Ours-$c_i$ to Ours-$c_i$, $v_i$, $l_{ij}$ and Ours-$c_i$, $v_i$, $l_{ij}$ to Ours-$c_i$, $v_i$, $l_{ij}$, $d_i$. The aim is to understand the effect of using RGB and depth maps on detection rate of each predicate. The bars are sorted according to the larger improvements. The darker shades within each blue and red area indicate larger frequency of those predicates appearing in the test set. An improvement in predicates with more frequency has a larger effect on the total accuracy presented in Table 1.

Table 2: Comparing predicate prediction performance given different feature extractors. When a model pre-trained on RGB images is employed for depth map feature extraction, the accuracy drops to even less than RGB & Label setting with no depth maps (compare with Table 1). Highest accuracy is achieved when the feature extractor is trained from scratch and specifically for the relation detection task.

| Task                  | Zero-shot Phrase Det. | Predicate Pred. |
|-----------------------|-----------------------|------------------|
|                       | R@100     | R@50    | R@100     | R@50    |
| Eitel - Raw           | 87.43     | 72.28   | 96.54     | 90.47   |
| Eitel - Pre-trained   | 83.59     | 67.20   | 60.29     | 45.77   |
| VGG16 - Pre-trained   | 82.74     | 66.44   | 71.17     | 48.93   |

Depth Feature Extraction  In this experiment, we elaborate on the discussion presented in Section 3.2 and evaluate the effect of feature extraction on predicate prediction accuracy. Table 2 presents the results for this experiment. We compare the results when (1) The Eitel [5] architecture is pre-trained for object detection using the RGB images of the 100 object categories (in VRD dataset) and then fine-tuned to depth maps of VRD for relation detection (Eitel - Pre-trained). (2) The Eitel [5] architecture is trained from scratch using depth maps of VRD for relation detection (Eitel - Raw). (3) A deeper network (VGG16 [19]) is pre-trained on a larger dataset (ImageNet [17]) and fine-tuned on depth maps of VRD for relation detection (VGG16 - Pre-trained). Note that VRD dataset does not contain sufficient data to also train the VGG16 from scratch in a comparable manner to ImageNet data.

Discussion: As the results show, a small network trained specifically on depth maps for the relation detection task (Eitel - Raw) is superior to a network using weights pre-trained on RGB images for object detection (Eitel - Pre-trained) even if the network is deeper and pre-trained on a much larger corpus such as ImageNet (VGG16 - Pre-trained). In fact, when we used pre-trained networks there was no improvement in performance at all.
5 Conclusion

We identified 3D information as an important attribute for visual relation detection. Since this information is typically not provided in visual data sets, we employed an RGB-to-Depth network, which was pre-trained on a large corpus of data, to generate depth information. We showed that for relation detection, one gets significant improvements by training a CNN feature extractor for the depth images from scratch, rather than using a CNN optimized for RGB data, as it is common practice. In empirical evaluations, we demonstrate that by using depth information, one achieves significantly better performance compared to other state-of-the-art methods.

References

[1] Stephan Baier, Yunpu Ma, and Volker Tresp. Improving visual relationship detection using semantic modeling of scene descriptions. In International Semantic Web Conference, pages 53–68. Springer, 2017.

[2] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Unsupervised feature learning for rgb-d based object recognition. In Experimental Robotics, pages 387–402. Springer, 2013.

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 26, pages 2787–2795. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data.pdf.

[4] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 601–610. ACM, 2014.

[5] Andreas Eitel, Jost Tobias Springenberg, Luciano Spinello, Martin Riedmiller, and Wolfram Burgard. Multimodal deep learning for robust rgb-d object recognition. In Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, pages 681–687. IEEE, 2015.

[6] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, pages 249–256, 2010.

[7] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In European Conference on Computer Vision, pages 345–360. Springer, 2014.

[8] Caner Hazirbas, Lingni Ma, Csaba Domokos, and Daniel Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In Asian Conference on Computer Vision, pages 213–228. Springer, 2016.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
[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. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[11] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 239–248. IEEE, 2016.

[12] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *European Conference on Computer Vision*, pages 852–869. Springer, 2016.

[13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[14] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[15] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. 2011.

[16] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2016.

[17] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[18] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Learning 3-d scene structure from a single still image. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

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

[20] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pages 926–934, 2013.

[21] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pages 2071–2080, 2016.

[22] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

[23] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 670–685, 2018.

[24] Ruichi Yu, Ang Li, Vlad I Morariu, and Larry S Davis. Visual relationship detection with internal and external linguistic knowledge distillation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.
[25] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5831–5840, 2018.

[26] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, volume 2, page 4, 2017.