VRED: A Deep Detector of Visual Relationships Informed about Their Semantic Similarity

Khoi Nguyen and Sinisa Todorovic
School of Electrical Engineering and Computer Science
Oregon State University, Corvallis, OR 97330
Email: nguyenkh@oregonstate.edu, sinisa@eecs.oregonstate.edu

***Abstract*—This work addresses the problem of detecting visual relationships in images. A visual relationship is a triplet $\langle subject, predicate, object\rangle$ (e.g. $\langle man, ride, bike\rangle$), where the predicates include a range of spatial relationships and interactions between the subjects and objects. From bounding boxes of detected subject-object pairs, we first extract deep visual and spatial features, and then project them onto a visual-relationship semantic space. This projection allows our two key contributions: a) Regularization of learning such that deep features of semantically similar relationships are closer to one another than to features of other types of relationships; and b) Taking into account domain constraints among the predicted classes of subjects, predicates, and objects through "message passing". Our results show that we outperform the state-of-the-art on both the VRD and Visual Genome datasets. We also present insightful visualizations of the embedding of visual relationships.**

I. Introduction

Given an image, our goal is to detect and localize visual relationships present. A visual relationships is defined as a triplet $\langle subject, predicate, object\rangle$, where the predicate is specified as one of the following types: positional (spatial), comparative, and interactive relationships. A positional relationship indicates a relative spatial position between the subject and object, e.g., $\langle bike, in\ \ front\ \ of, bus\rangle$. A comparative relationship compares a particular attribute of the subject and object, e.g., $\langle bus, taller\ \ than, man\rangle$. An interactive relationship defines an action that the subject performs on the object, e.g., $\langle man, ride, bike\rangle$. Fig. 1 illustrates examples of the three relationship types. Note that an object may serve as both subject and object in different subject-object pairs in the image, and that a subject-object pair may be characterized by more than one visual relationship.

Recently, visual relationship detection has received significant traction in the literature [1]–[10]. This problem is important, because detecting $\langle subject, predicate, object\rangle$ triplets can be used to efficiently build higher-order image representations, including scene graphs [11], [12], or tree-like scene structures [13].

The key challenge in visual relationship detection is learning from relatively small datasets the large semantic space of $\langle subject, predicate, object\rangle$ triplets. For N object categories and K predicate categories, the total number of distinct triplet categories is N^2K. This number is typically very large in comparison with the number of training examples in current benchmark datasets, such as Visual Relationship Dataset [1] with $N = 100$ and $K = 70$, and (only) $37,993$ examples.

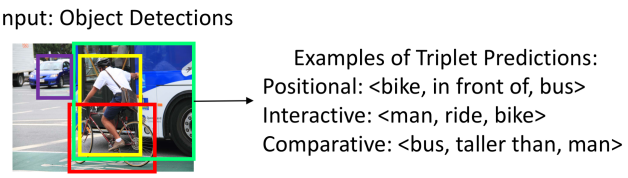

Fig. 1: Visual relationship detection: for all candidate subject-object pairs we predict their positional, interactive and comparative relationships, as well as localize bounding boxes of the subjects (yellow) and objects (red). As can be seen, an object in the image may be both a subject and object depending on the relationship considered, and a subject-object pair may have more than one visual relationship.

Recent work typically addresses this challenge in two steps [1]–[10]. They first detect objects in the image with an object detector, and then apply a a convolutional neural network (CNN) on every pair of detected bounding boxes for predicting a relationship class. These methods have some limitations: (a) Poor generalization capabilities to previously unseen relationships; and (b) Not explicitly taking into account domain constraints in terms of common dependencies among subject, predicate, and object classes.

To address these limitations, we specify a new end-to-end deep architecture, illustrated in Fig. 2. Input to our network are bounding boxes of object detections in the image, and output consists of labeled bounding boxes of subjects and objects as well as predictions of their visual relationships. As shown in Fig. 2, our approach has four steps: (1) Selection of candidate subject-object pairs based on input object detections; (2) Feature extraction from bounding boxes around the selected subject-object candidates; (3) Feature embedding in the semantic space of visual relationships; and (4) Iterative "message passing" for taking into account domain constraints among the predicted classes of subjects, predicates, and objects. In this paper, domain constraints represent common dependencies among subject, predicate, and object classes, which we learn directly from training images.

While prior work typically fixes detected object classes after step (1), we allow the classes to change in the "message passing", and thus ensure that they respect domain constraints. Our key novelty is the embedding of deep features in the semantic space of visual relationships, illustrated in Fig. 2,

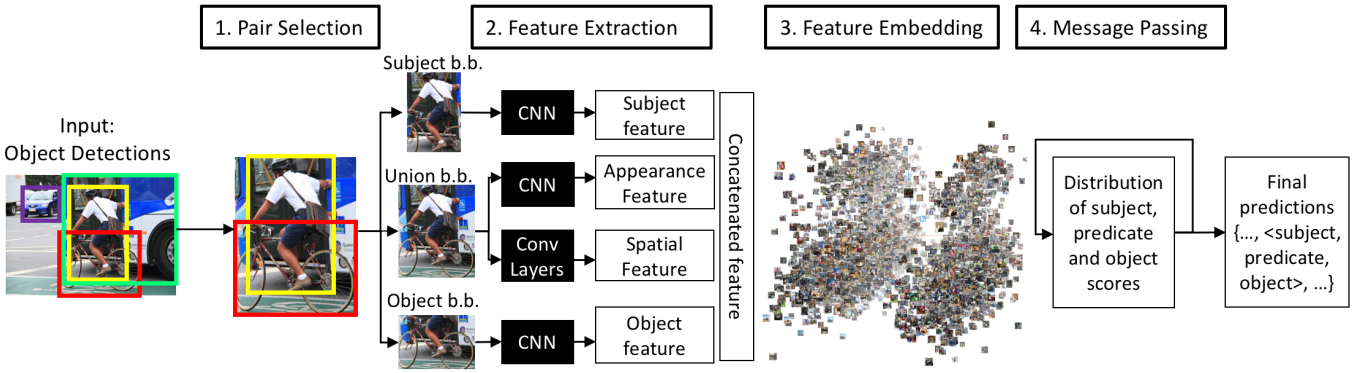

Fig. 2: Our approach consists of four steps: subject-object pair selection from input object detections, feature extraction from bounding boxes (BB), feature embedding in a semantic space of visual relationships, and message passing.

such that features of semantically similar relationships are closer to one another than to features of other types of relationships. Hence, we call our deep architecture visual relationship embedding detector (VRED).

Our evaluation on the benchmark Visual Relationship Dataset (VRD) [1] and Visual Genome (VG) [14] datasets shows that VRED generalizes well to previously unseen relationships, unlike existing work. The paper also presents insightful visualizations of our feature embedding in the high-dimensional visual-relationship space.

Our main contributions include:

- A new deep architecture, VRED. VRED leverages dependencies among subjects, predicates and objects to improve recognition of each.
- We are not aware of any related work that specifies a visual-relationship feature embedding. VRED uses this embedding as a regularizer in learning. In this way, VRED addresses the aforementioned key challenge, that of learning from few examples.
- VRED significantly outperforms the state-of-the-art on the VRD and Visual Genome datasets.

In the following, Sec. II specifies VRED, Sec. III describes our implementation details, Sec. IV reports our experimental results, and Sec. V presents our concluding remarks.

II. Our Deep Architecture

Fig. 2 illustrates the four main steps of VRED: selection of subject-object pairs from input object detections, feature extraction, feature embedding, and message passing.

Given an image, we apply Faster-RCNN [15] for identifying candidate objects. Then, in the first step, VRED couples these object detections into all possible (legal) subject-object pairs which can have a visual relationship. For every selected pair, we use a CNN to compute four deep features: two are extracted from bounding boxes of the subject and object, and other two visual and spatial features are extracted from the tightest union bounding box around the subject and object. These four features are concatenated, and then projected onto a visual-relationship semantic space. The resulting embedded feature is then passed to an iterative fully connected layer aimed at performing "message passing" toward the final prediction of the subject, object and predicate classes that are consistent with domain constraints. In the following, we describe the above four steps in greater detail.

A. Subject-Object Pair Selection

The first step is aimed at selecting pairs of input object detections that can be in a subject-object relationship. To this end, we use the pair-filter network, introduced in [5], which usually reduces a total of n^2 pairs from n object detections[1] to 2–3 times smaller number of valid subject-object pairs, depending on the image. The pair-filter network of [5] is shown in Fig. 3. Note that we re-use the spatial feature for the subsequent steps in VRED.

B. Feature Extraction

For every subject-object pair, selected in the previous step, we apply a CNN to the union and individual bounding boxes of the subject and object, and use the CNN's top-layer activations as visual features for each bounding box. Implementation details about the CNN are described in Sec. III. The three visual features are then concatenated with the spatial feature, specified in Sec. II-A, to form a unified feature representation of the subject-object pair.

C. Feature Embedding

It is straightforward to form meaningful sentences in English from triplets $\langle subject, predicate, object \rangle$. These English sentences, in turn, can be readily mapped to a semantic feature space using an off-the-shelf sentence-to-vec network. In this paper, we use InferSent [16] for its many advantages. InferSent ensures that English sentences with close semantic meanings are represented by feature vectors whose Euclidean distances are smaller than distances to other features

[1]Note that the subject-object ordering in a pair matters.

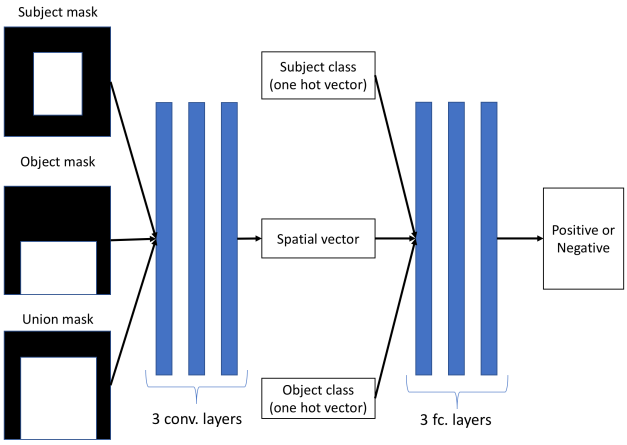

Fig. 3: The pair-filter introduced in [5] is used for our subject-object pair selection. Given bounding boxes of two object detections, their corresponding binary masks are passed through three convolutional layers for extracting the spatial feature. This feature is then concatenated with one-hot vector representing the subject and object classes, and further processed by fully connected layers for predicting whether the two object detections represent a valid subject-object pair (positive class) which can have a visual relationship, or illegal combination (negative class).

representing semantically unrelated sentences. Given ground-truth $\langle subject, predicate, object\rangle$ triplets in training, we use InferSent to generate their semantic feature space, called visual-relationship space.

The third step of VRED projects features of detected subject-object pairs onto the visual-relationship space using a fully connected layer. This layer is trained with a loss, defined as a Euclidean distance between the feature projection and InferSent's embedding of the English sentence corresponding to the ground-truth $\langle subject, predicate, object\rangle$ triplet.

There are many advantages of using this feature embedding. First, it serves as a strong regularizer in our learning, and thus facilitates the prediction of subject, predicate, and object classes under uncertainty (e.g., partial occlusion). For example, chances of misclassifying $\langle man, ride, bicycle\rangle$ as $\langle man, carry, bicycle\rangle$ get reduced in the visual-relationship space, since they are semantically dissimilar and thus mapped to far away features. Second, the embedding helps us overcome the problem of recognizing previously unseen relationships, and hence improve our generalization capabilities. Finally, the embedding can be used for further extensions of our work (not explored in this paper), including a robust image retrieval, and generation of rich, additional interpretations of a subject-object relationship by analyzing "neighbors" of the predicted triplet in the visual-relationship space.

After obtaining the projected features, we use another fully connected layer to produce the distribution of prediction scores for subject, predicate, and object classes.

D. Message Passing

To ensure that our predictions are consistent with domain constraints, the aforementioned distributions of prediction scores for subject, predicate, and object are passed to the "message passing" module, introduced in [5]. This module represents an iterative fully connected layer, which repeatedly updates the distributions of subject, predicate and object, denoted as q_s, q_r, q_o, as

$$\begin{aligned} q_s &\leftarrow \sigma(W_{ss}q_s + W_{rs}q_r + W_{os}q_o), \\ q_r &\leftarrow \sigma(W_{rr}q_r + W_{sr}q_s + W_{or}q_o), \\ q_o &\leftarrow \sigma(W_{oo}q_o + W_{so}q_s + W_{ro}q_r), \end{aligned} \tag{1}$$

where σ is the ReLU activation function, and W_{ij} is a learnable weight matrix that captures dependences between two class distributions i and j. Note that W_{ij} is the transpose of W_{ji}.

VRED is trained end-to-end using the cross-entropy loss of distributions q_s, q_r, q_o estimated by the "message passing" module, and regularized by the $L2$-norm in the visual-relationship space.

E. Ranking the Results

For all input object detections identified as valid subject-object pairs, VRED outputs distributions q_s, q_r, q_o, which in turn enables multiple distinct interpretations of every pair as triplets $\langle subject, predicate, object\rangle$. For selecting top-$K$ results, we rank all triplet interpretations based on a product of subject, predicate, and object scores: $s_{triplet} = q_s(subject) \cdot q_r(predicate) \cdot q_o(object)$.

III. Implementation Details

This section presents our implementation details. Below, we will use K and N to denote the number of predicate and object classes in a dataset, respectively.

Architecture details. Our approach is implemented in PyTorch Deep Learning framework [17]. The CNN used for extracting visual features is Densenet [18] pre-trained with Image-net [19]. We use the 2208-dimensional feature output right before the softmax layer. The spatial feature is computed with 3 convolutional layers with ReLU activations between them. The filter sizes are $5\times5, 5\times5, 8\times8$, and the number of filters are 95, 128, 64, respectively. In the "message passing" module, we use weight matrices with dimensions equal to the number of classes considered, e.g., W_{sr}, W_{or}, W_{rr} have sizes $N\times K, N\times K, K\times K$, respectively.

Training details. We use Faster-RCNN pre-trained on the VRD and VG datasets for object detection. Details of this training can be found in [8]. The rest of VRED's components are trained in an end-to-end manner. We use the batch size 8; Adam optimizer [20] with the initial learning rate 10^{-4}; and Reduce-on-Plateau scheduler for monitoring the learning rate with weight decay 10. In the "message passing", the activation σ is ReLU [21], and the number of iterations is 10.

IV. EXPERIMENTS

Datasets: We evaluate our approach on two datasets: Visual Relationship Dataset (VRD) [1] and Visual Genome (VG) [14]. VRD consists of $5,000$ images showing $37,993$ visual relationship instances that belong to $6,672$ triplet types. The numbers of object and predicated classes are 100 and 70. We follow the train/test split setting presented in [1]. VG is much larger dataset aimed of multiple tasks including Visual Relationship Detection. We follow [8] and use their subset of VG. In particular, the VG subset we consider contains $99,658$ images with 200 object categories and 100 predicate categories, resulting in $1,174,692$ relationship annotations with $19,237$ unique relationships and 57 predicates per object category. We split the data into $73,801$ images for training and $25,857$ images for testing.

Tasks: Following [1], we evaluate our approach in three tasks: predicate prediction, phrase detection and relationship detection. Below, we defined each task.

- **Predicate prediction**: Given subject and object classes and bounding boxes, predict the class of predicate.
- **Phrase detection**: Given an image, predict the three classes of the triplet, and localize its union bounding box. A phrase detection is judged as correct if it exists in the image, and the detected union bounding box has $IoU \geq 0.5$ with its respective ground-truth.
- **Relationship detection**: Same as Phrase detection but need to localize subject and object bounding boxes instead of union bounding box.

Metrics: Following [1] and other related work, we use $Recall@K$ as evaluation metric. $Recall@K$ (or $R@K$) is the fraction of ground-truth instances that are correctly recalled in top K predictions. We report $Recall@100$ and $Recall@50$ in our experiments. The reason for using *recall* instead of *precision* is that the annotations are incomplete, where some visual relationships appearing in images are not annotated. Using $Recall@K$ for evaluation has become standard practice in related work.

A. The Ablation Study on the VRD dataset

In our ablation studies, we test several variants of VRED all of which use as input ground-truth bounding boxes of objects present in images. In this way, we conduct more focused tests of modules in VRED, reducing the uncertainty arising from noise in input object detections. The VRED variants that we study gradually add modules of VRED to the simplest baseline, and thus test the effect of each module, as explained below. **Visual + Spatial (V+S)** is a variant of VRED that uses only the first two modules: subject-object pair selection and feature extraction. **Visual + Spatial + Message Passing (V+S+MP), Visual + Spatial + Project (V+S+P), and Visual + Spatial + Project + Message Passing (V+S+P+MP or VRED)** extends V+S by additionally applying "message passing", "projection" and both respectively.

Tab. I shows our ablation results on the VRD dataset. From the table, we can see that when detections of subject and object are perfect (i.e., provided by ground truth), the variants of VRED that do not have "message passing" perform by 5% better than the other variants with "message passing". While "message passing" seems redundant for ideal inputs, this is not the case for noisy inputs, as we show in the following experiments. Also, the ablation results in Tab. I suggest that feature embedding in the visual-relationship space leads to by 3% performance improvement. Therefore, we choose our two variants – V+S+P and V+S+P+MP, where the latter is our full VRED – for the following experiments with noisy object detections as input.

TABLE I: The ablation study on VRD

Method	V+S	V+S+MP	V+S+P	V+S+P+MP
R@50	76.28	76.7	**82.31**	78.33
R@100	87.37	82.27	**90.38**	85.98

TABLE II: Review of recent results for phrase and relationship detection on VRD

Method	Phrase Detection		Rel. Detection	
	R@50	R@100	R@50	R@100
Lang. Prior [1] (k =?)	16.17	17.03	13.86	14.7
VTransE [8] (k =?)	19.42	22.42	14.07	15.2
DRNet [5] (k =?)	19.02	22.85	16.94	20.22
Vip-CNN [9] (k =?)	22.78	27.91	17.32	20.01
VRL [4] (k =?)	21.37	22.6	18.19	20.79
Weakly-sup. [2] (k =?)	17.9	19.5	15.8	17.1
PPR-FCN [3] (k =?)	19.62	23.15	14.41	15.72
Towards [7] (k =?)	24.04	25.56	20.35	23.52

B. Evaluation on VRD

In this section, we evaluate on two tasks – phrase detection and relationship detection – using as input bounding boxes of object detections produced by Faster-RCNN [15] on the VRD dataset. In our experiments, the number of objects detected by Faster-RCNN is around 22 per image. Thus, number of subject-object pairs is more than 400.

Tab. II reviews results of recent approaches for the tasks of phrase and relationship detection on the VRD dataset, and Tab. III compares our results with the state of the art for the same tasks on the same dataset. We split these results in two tables, because the approaches listed in Tab. II did not report the parameter K used for evaluating their $Recall@K$. Consequently, a fair comparison with the approaches in Tab. II may not be possible. Fortunately, two recent approaches Distill [6] and CCA [10] report their $Recall@K$ for $K = 10$. We also choose $K = 10$, and compare our results with these two approaches in Tab. III.

From Tab. III, our V+S+P+MP (i.e., VRED) outperforms V+S+P, and hence in the following we just evaluate VRED. This demonstrates that the "message passing" module is critical for dealing with noisy object detections in input, since it accounts for domain constraints and thus helps correctly resolve prediction ambiguities. Also, Tab. III shows that our VRED outperforms Distill [6] and CCA [10].

TABLE III: Comparison with the state of the art for phrase and relationship detection on VRD

Method	Phrase Detection		Rel. Detection	
	R@50	R@100	R@50	R@100
Distill [6] ($k = 10$)	26.47	29.76	22.56	29.89
CCA [10] ($k = 10$)	-	-	15.08	18.37
V+S+P ($k = 10$)	17.58	28.87	14.02	23.33
VRED ($k = 10$)	**26.7**	**35.13**	**23.15**	**30.23**

TABLE IV: Zero-shot learning results on VRD

Method	Phrase Detection		Rel. Detection	
	R@50	R@100	R@50	R@100
Lang. Prior [1] (k =?)	3.36	3.75	3.13	3.52
VRL [4] (k =?)	9.17	10.31	7.94	8.52
Towards [7] (k =?)	**10.78**	11.3	9.54	10.26
Weakly sup. [2] (k =?)	7.4	8.7	7.1	8.2

C. Generalization Using a Zero-Shot Split of VRD

In this section, we follow [1] and other related work to report results on a "zero-shot split" of the VRD test set. This test set consists of test triplets that are not seen in training. As input, we use bounding boxes from Faster-RCNN detections in the same 'zero-shot split" test set as in [1]. As before, we split results in Tab. IV and Tab. V, where Tab. IV reviews resent results on Zero-Shot Split, and Tab. V compares our results with those of Distill [6] and CCA [10]. We cannot directly compare with approaches in Tab. IV, since they did not specify K. Tab. V shows that our feature embedding module enables VRED to generalize well to unseen examples. Also, we achieve a comparable performance to the state-of-the-art the art, and some improvement on $R@100$ for phrase prediction.

D. Evaluation on VG

Tab. VI reports results for predicate prediction, phrase detection and relationship detection on the VG dataset. For the phrase and relationship detection tasks, we evaluate VRED on input object detections from Faster-RCNN. The table shows that VRED outperforms the state of the art. Specifically, it is by 4.88% better in $R@100$ for phrase detection.

TABLE V: Zero-shot learning results on VRD

Method	Phrase Detection		Rel. Detection	
	R@50	R@100	R@50	R@100
Distill [6] ($k = 10$)	10.44	10.89	8.89	9.14
CCA [10] ($k = 10$)	-	-	**9.67**	**13.43**
VRED ($k = 10$)	9.76	**12.38**	9.53	13.09

TABLE VI: Predicate prediction (PP), phrase detection (PD), and relationship detection (PD) on VG

Method	PP		PD		RD	
	R@50	R@100	R@50	R@100	R@50	R@100
VTrans [8]	44.76	44.76	9.46	10.45	5.52	6.04
PPR-FCN [3]	47.43	47.43	10.62	11.08	6.02	6.91
VRED ($k = 10$)	**64.71**	**65.99**	**13.32**	**15.96**	**6.99**	**8.13**

Fig. 4: A visualization of our visual-relationship embedding, see more at https://goo.gl/X6Wj4b

E. Visualizing Visual Relationship Semantic Embedding on VRD dataset

Fig. 4 shows a visualization of the visual-relationship space with 4096-dimensional features which we extracted from VRD. We provide full visualizations at https://goo.gl/X6Wj4b. As can be seen, visual relationships form distinct clusters according to their semantic similarity. These groupings are likely to facilitate prediction of triplets, especially in the face of noisy input object detections.

F. Qualitative Results on VRD

Fig. 5 shows examples of our results on VRD. As can be seen, VRED usually predicts well the predicate class. However, VRED typically makes mistakes for semantically similar predicates, e.g., the ground truth predicate "hold" in $\langle person, hold, umbrella \rangle$ is misclassified as "under" in $\langle person, under, umbrella \rangle$.

V. Conclusion

We have developed a new deep architecture, VRED, for detecting $\langle subject, predicate, object \rangle$ triplets, called visual relationships, in images. VRED and its variants have been evaluated on the benchmark VRD and VG datasets in three tasks: predicate prediction, phrase detection and relationship detection. The ablation results show that feature embedding in VRED leads to performance improvement. Also, the "message passing" module helps deal with noise in input object detections. On both datasets, VRED outperforms the state of the art in all of three tasks, and demonstrates competitive generalization capabilities to previously unseen examples. We have also presented insightful visualizations of our feature embedding.

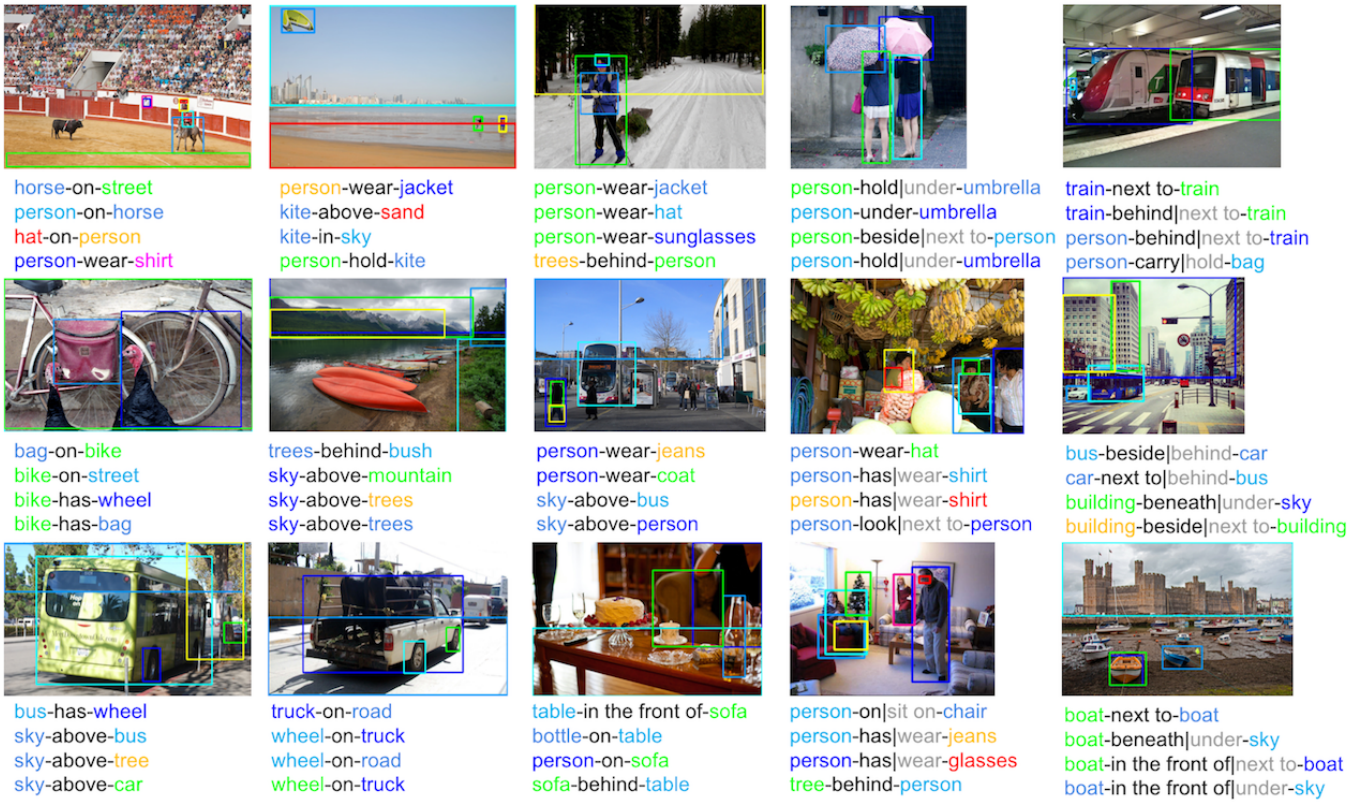

Fig. 5: Examples of our results on the VRD dataset. The first three columns show correct predictions of VRED and the last two columns show failure examples. The ground truth predicates are marked black, and our predictions are marked grey.

REFERENCES

[1] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," in *ECCV*. Springer, 2016, pp. 852–869.

[2] J. Peyre, I. Laptev, C. Schmid, and J. Sivic, "Weakly-supervised learning of visual relations," *ICCV*, 2017.

[3] H. Zhang, Z. Kyaw, J. Yu, and S.-F. Chang, "Ppr-fcn: Weakly supervised visual relation detection via parallel pairwise r-fcn," *IEEE ICCV*, 2017.

[4] X. Liang, L. Lee, and E. P. Xing, "Deep variation-structured reinforcement learning for visual relationship and attribute detection," in *CVPR*, July 2017.

[5] B. Dai, Y. Zhang, and D. Lin, "Detecting visual relationships with deep relational networks," in *CVPR*, July 2017.

[6] R. Yu, A. Li, V. I. Morariu, and L. S. Davis, "Visual relationship detection with internal and external linguistic knowledge distillation." *ICCV*, 2017.

[7] B. Zhuang, L. Liu, C. Shen, and I. Reid, "Towards context-aware interaction recognition," *ICCV*, 2017.

[8] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua, "Visual translation embedding network for visual relation detection," in *CVPR*, July 2017.

[9] Y. Li, W. Ouyang, X. Wang, and X. Tang, "Vip-cnn: Visual phrase guided convolutional neural network," in *CVPR*, July 2017.

[10] B. A. Plummer, A. Mallya, C. M. Cervantes, J. Hockenmaier, and S. Lazebnik, "Phrase localization and visual relationship detection with comprehensive image-language cues," in *CVPR*, 2017, pp. 1928–1937.

[11] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in *CVPR*, July 2017.

[12] A. Newell and J. Deng, "Pixels to graphs by associative embedding," *NIPS*, 2017.

[13] L. Lin, G. Wang, R. Zhang, R. Zhang, X. Liang, and W. Zuo, "Deep structured scene parsing by learning with image descriptions," in *CVPR*, 2016, pp. 2276–2284.

[14] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," 2016. [Online]. Available: https://arxiv.org/abs/1602.07332

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015, pp. 91–99.

[16] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," *EMNLP*, 2017.

[17] A. Paszke, S. Gross, and S. Chintala. Tensors and dynamic neural networks in python with strong gpu acceleration. [Online]. Available: http://pytorch.org/

[18] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, July 2017.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, 2009.

[20] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR*, 2015.

[21] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010, pp. 807–814.