CVML TEST CASE
Visual Relationship Detection

July 13, 2020

1 Task Overview

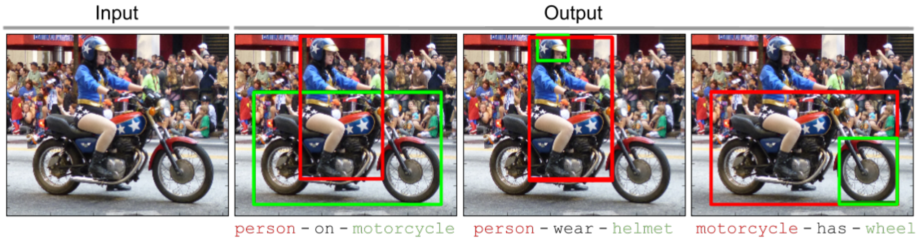

Figure 1: An example of the visual relationship detection. Taking an image as input, the designed model detects multiple relationship phrases in the form of $< object_1 - relationship(predicate) - object_2 >$. Both the objects are localized in the image as bounding boxes.

In this test case, your task is to perform the visual relationship detection on Visual Relationship Dataset [3] (termed VR-dataset). Figure 1 shows an example of the visual relationship detection, where the possible relationship phrases are detected in the image based on the bounding boxes of the objects. But different from the previous work [3], where the relationship phrases of all the relationship types are expected to be detected (see Figure 2), you are asked to detect the relationship phrases about the types *Action* and *Verb*. We name this subset of VR-dataset as VR-selected-dataset. To start visual relationship detection, we provide three subtasks with different difficulty levels:

1. **Subtask 1 (basic) - Classification based visual relationship detection**. First, train the first VGG-16 net [5] in VR-dataset to classify the bounding boxes of the objects in VR-selected-dataset. Next, train the second VGG-16 net in VR-selected-dataset to classify the bounding boxes of the relationships (predicates) in VR-selected-dataset. The bounding box of each relationship is the union of the bounding boxes

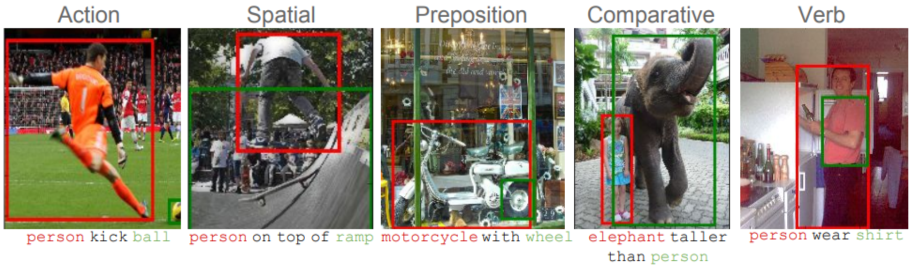

Figure 2: Types of relationships in Visual Relationship Dataset.

of the two participating objects in the ground truth relationship phrase. Then, implement and optimize a ranking loss function to maximize the rank of the ground truth relationship phrase with the bounding boxes of objects. You can refer to the Equation 1 and Equation 6 in the paper [3]. But note that, you don't have to consider the zero-shot problem and language module in the paper [3]. Finally, obtain the candidate bounding boxes in the test set of VR-selected-dataset via Faster-RCNN [4] (trained on MSCOCO [2]), and conduct the evaluation by referring to the Equation 8 in the paper [3] and using the codes "result_visualization.m", "top_recall_Phrase.m" and "top_recall_Relationship.m" from Evaluation Tool. For the visualization, two examples in VR-selected-dataset are required: "3665708190_b99175077d_o.jpg" and "4418514401_cd86bc8e53_b.jpg". Beyond that, you are encouraged to show and analyze more.

Tip: You can use the VGG-16 model pretrained on ImageNet [1] to improve the classification.

2. **Subtask 2 (basic) - Object detection based visual relationship detection**. First, train Faster-RCNN [4] (using VGG-16 as the inner net) in VR-dataset, to detect the the bounding boxes of objects in VR-selected-dataset. Next, train a VGG-16 net in VR-selected-dataset to classify the bounding boxes of the relationships (predicates) in VR-selected-dataset. The bounding box of each relationship is the union of the bounding boxes of the two participating objects in the ground truth relationship phrase. Then, implement and optimize a ranking loss function to maximize the rank of the ground truth relationship phrase with the *detected* bounding boxes of objects. You need to set an IoU floor for the correct bounding boxes and modify the Equation 1 and Equation 6 to take the detected bounding boxes into consideration. Finally, obtain the candidate bounding boxes in the test set of VR-selected-dataset via the trained Faster-RCNN model, and conduct the evaluation as with **Subtask 1**.

   Tip: You can use the Faster-RCNN model pretrained on MSCOCO [2] to improve the detection.

3. **Subtask 3 (advanced and optional) - Visual relationship detec-**

Table 1: Setting of relationship types

Relationship Type	Relationships (Predicates)
Action & Verb	walk to, walk, walk past, walk beside, hold, ride, touch, drive, drive on, eat, pull, talk, fly, play with, follow, hit, feed, kick, skate on, wear, has, sleep next to, sit next to, stand next to, park next, walk next to, stand behind, sit behind, park behind, stand under, sit under, sit on, carry, look, stand on, use, attach to, cover, watch, contain, park on, lying on, lean on, face, sleep on, rest on
Spatial & Preposition	next to, above, behind, in the front of, under, near, below, beside, beneath, on the top of, on the left of, on the right of, inside, adjacent to, outside of, on, in, over, by, with, at, against, across
Comparative	taller than

tion with the condition of the relationship type. In an image of VR-dataset, the detected relationship phrases may cover several types. For example, Figure 1 shows two types of relationships, *i.e.*, *Spatial* and *Verb*. And the relationship phrases with the same participating objects may be different due to the different relationship types, *e.g.*, $< person - on - motorcycle >$ and $< person - ride - motorcycle >$. In this subtask, you are asked to take the types of relationships as the condition/input and detect the specific relationships and relationship phrases. For example, the designed model takes the image of Figure 1 along with *Verb* type as input and outputs $< person - ride - motorcycle >$. The key problem is to explore the correlations among the relationship types and distinguish the relationship phrases with the condition of relationship type. Nevertheless, it's still an open problem and you are free to explore the problem and design your algorithm based on **Subtask 1** by using VR-dataset. For the evaluation, you should take each image-type pair as an instance and conduct the evaluation under all the types and the specific type, respectively. You can use the evaluation codes in **Subtask 1** and probably add R@25 (Recall @ 25) metric for the better evaluation. Note that, a potential solution is to transfer the model directly from **Subtask 1** to **Subtask 3** and select the type-specific relationships during evaluation. So you need to make a comparison between this nonparametric and your parametric-learning methods.

2 Introduction to The Dataset

Visual Relationship Dataset [3] (VR-dataset) contains 5,000 images with 100 object categories and 70 relationships (predicates). There are totally 37,993 relationship phrases among all the images with 6,672 relationship phrase categories. Each object category has 24.25 relationships on average. VR-dataset is split into the training set and test set with 4,000 images and 1,000 images, respectively. Each image is annotated with several ground truth relationship phrases and their corresponding object bounding boxes. For the relationship type, the paper [3] declares that VR-dataset contains *Action*, *Spatial*, *Preposition*, *Comparative*, and *Verb*. But because some relationships are hard to distinguish among the relationship types, *e.g.*, "on" can belong to *Spatial* and *Preposition*, we reset the relationship types showed in Table 1. For **Subtask 3**, two relationship types, *i.e.*, *Action-Verb* and *Spatial-Preposition*, are used due to the number of their relationships. You can assign the images with relationship types and filter the images and the ground truth relationship phrases in VR-dataset according to those two relationship types in Table 1.

VR-selected-dataset derives from VR-dataset. To construct VR-selected-dataset, you can filter the images and the ground truth relationship phrases according to the relationship type *Action-Verb* in Table 1.

3 Formalized Description of the Task

During the model training, let $\mathcal{S}_{<i,k,j>} =< \mathcal{O}_i, \mathcal{R}_k, \mathcal{O}_j >$ denote a ground truth relationship phrases with the i-th object category, the j-th object category, and the k-th relationship category in an image I. For all N ($N = 100$) object categories, $\mathcal{O}_i$ and $\mathcal{O}_j$ are respectively indexed by $i, j = 1, \ldots, N$. For all K ($K = 70$) relationship categories, $\mathcal{R}_k$ is indexed by $k = 1, \ldots, K$. Let B_1 and B_2 denote the bounding boxes of $\mathcal{O}_i$ and $\mathcal{O}_j$ respectively in $\mathcal{S}_{<i,k,j>}$. And let $U(B_1, B_2)$ denote the union of the bounding boxes B_1 and B_2. For the **Subtask 1**, you should use the first CNN to compute the probabilities $P(\mathcal{O}_i|B_1)$ and $P(\mathcal{O}_j|B_2)$ of classifying B_1 and B_2 as $\mathcal{O}_i$ and $\mathcal{O}_j$ respectively. You can compute the probability of $\mathcal{S}_{<i,k,j>}$ conditioned on the pair $< B_1, B_2 >$ of bounding boxes via:

$$P(\mathcal{S}_{<i,k,j>}| < B_1, B_2 >) = P(\mathcal{O}_i|B_1)\big[\mathbf{w}_k^T F\big(U(B_1, B_2)\big) + b_k\big]P(\mathcal{O}_j|B_2), \quad (1)$$

where $F()$ denotes the operation of the feature extraction in the final fully-connected layer of the second CNN. $F\big(U(B_1, B_2)\big)$ outputs a feature vector for $U(B_1, B_2)$. $\mathbf{w}_k$ and b_k are the parameters about mapping vector and bias respectively, which convert the CNN feature to the probability of the k-th relationship. We set $\theta = \{\mathbf{w}_k, b_k|k = 1, \ldots, K\}$, which is the parameter set that need to be learned. Compared to the Equation 1 in the paper [3], Equation 1 is probably clearer. We provide a ranking loss to score the wrong probability estimation against the correct one:

$$\begin{aligned} L(\theta) = \sum_{<B_1,B_2>,\mathcal{S}_{<i,k,j>}} & \max\{1 - P(\mathcal{S}_{<i,k,j>}| < B_1, B_2 >) \\ & + \max_{\substack{<B'_1,B'_2>\notin C(\mathcal{S}_{<i',k',j'>}) \\ i',k',j' \neq i,k,j}} P(\mathcal{S}_{<i',k',j'>}| < B'_1, B'_2 >), 0\} \end{aligned} \tag{2}$$

where $C()$ denotes the operation that obtains the pair set of the bounding boxes according to the ground truth relationship phrase. If $\mathcal{S}_{<i',k',j'>}$ doesn't occur in the image, then $C(\mathcal{S}_{<i',k',j'>}) = \emptyset$. Compared to the Equation 6 in the paper [3], Equation 2 removes the language module and fixes some imperfect expressions, which is probably clearer. You should maximize the rank of the ground truth relationship phrase with the bounding boxes of objects by minimizing the ranking loss. During testing/evaluation, you should let your model to choose the correct relationship phrases given two bounding boxes as the Equation 8 in the paper [3].

For **Subtask 2**, you should take B_1 and B_2 as two candidate bounding boxes from Faster-RCNN [4] and recompute the $P(\mathcal{O}_i|B_1)$ and $P(\mathcal{O}_j|B_2)$ in Equation 1 by using the probabilities of the candidate bounding boxes from Faster-RCNN [4]. Besides, you should set an IoU floor to match the candidate bounding boxes and the ground truth bounding boxes to judge whether the candidate bounding boxes corresponds to the ground truth relationship phrases. Typically, IoU > 0.5. You also need to consider a trick to deal with the missing candidate bounding boxes which should have corresponded to the ground truth relationship phrases.

For **Subtask 3**, you are expected to rethink the Equation 1 to estimate $P(\mathcal{S}_{<i,k,j>}| < B_1, B_2 >, t)$, where t is the relationship type. In addition, you are expected to rethink the ranking losses among the relationship types and within the relationship types in Equation 2.

4 Preliminaries for The Experiment

It will be easier for you to complete the task if you have the following foundations:

1. Be familiar with some basic operations or knowledge in term of Deep Learning, such as Softmax, cross entropy loss, the algorithm of stochastic gradient descent, the structure of CNN, the optimization of the ranking loss, *etc*.

2. Be familiar with some image processing techniques and Python/C/C++/Matlab coding.

3. Know about the framework of object detection and can handle with the training/inference of some common-used object detection networks like RCNN, Fast-RCNN, Faster-RCNN, SSD or Yolo.

5 Requirements

There are some requirements you should note:

1. **Subtask 1** and **Subtask 2** are two compulsory items while **Subtask 3** is an optional item.

2. You should write a report of at most 2 page about your experiments and present your results and analysis. There are no restrictions regarding the framework/language that you can use for developing your models.

3. You should complete the experiments and the report before the required deadline.

References

[1] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Li, F.-F. Imagenet: A large-scale hierarchical image database. In *CVPR* (2009), pp. 248–255.

[2] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *ECCV* (2014), pp. 740–755.

[3] Lu, C., Krishna, R., Bernstein, M., and Li, F.-F. Visual relationship detection with language priors. In *ECCV* (2016), pp. 852–869.

[4] Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS* (2015), pp. 91–99.

[5] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).