CVPR 2018

Relation Networks for Object Detection

Han Hu1* Jiayuan Gu2*† Zheng Zhang1* Jifeng Dai1 Yichen Wei1
1 Microsoft Research Asia
2 Department of Machine Intelligence, School of EECS, Peking University
https://github.com/msracver/Relation-Networks-for-Object-Detection

主讲人：贾馥溪

2018.7.8

1. Itroduction

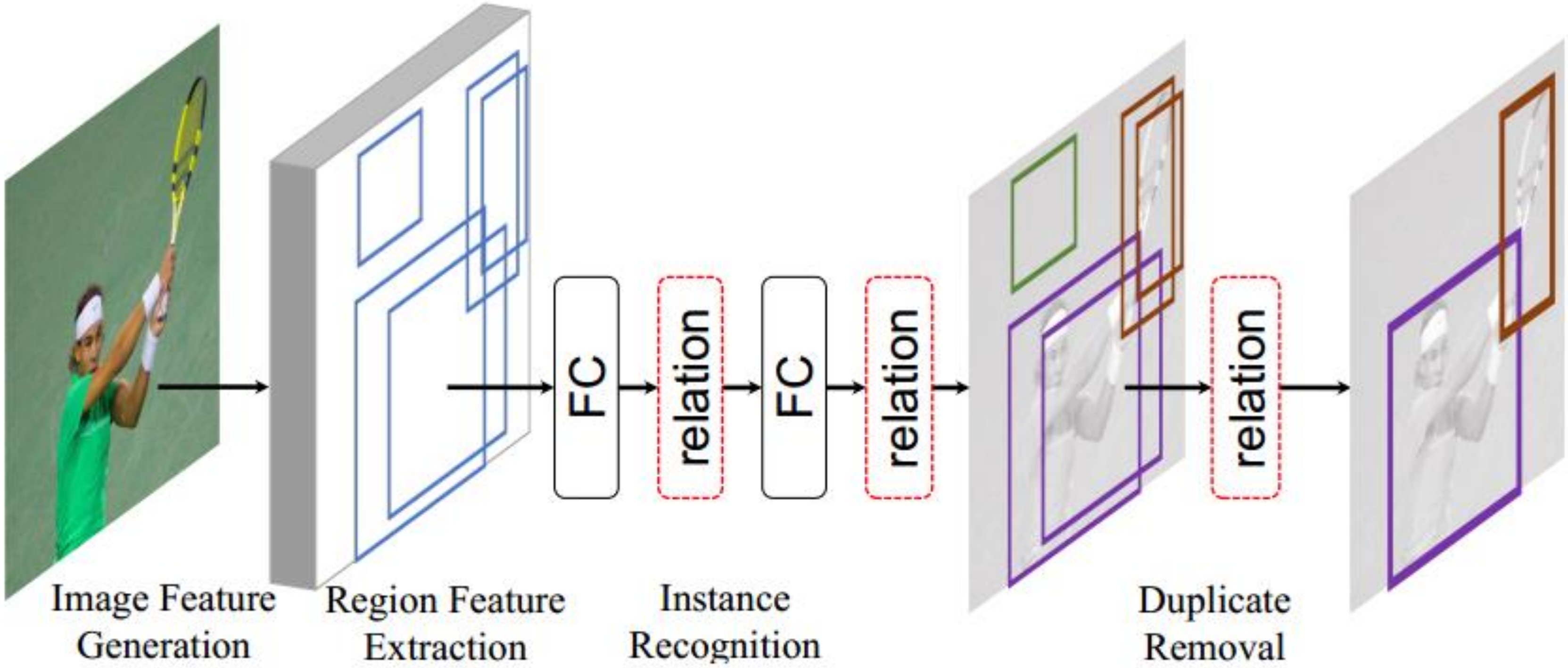


Figure 1. Current state-of-the-art object detectors are based on a four-step pipeline. Our object relation module (illustrated as red dashed boxes) can be conveniently adopted to improve both instance recognition and duplicate removal steps, *resulting in an end-to-end object detector*.

2. Object Relation Module

An object relation module aggregates in total N_r relation features and augments the input object's appearance feature via addition,

$$\mathbf{f}_A^n = \mathbf{f}_A^n + \text{Concat}[\mathbf{f}_R^1(n), ..., \mathbf{f}_R^{N_r}(n)], \text{ for all } n. \quad (6)$$

fA : *appearance feature*

—— is up to the task

fG : *geometric feature*

—— a 4-dimensional object bounding box

Given input set of N objects $\{(\mathbf{f}_A^n, \mathbf{f}_G^n)\}_{n=1}^N$, the *relation feature* $\mathbf{f}_R(n)$ of the whole object set with respect to the n^{th} object, is computed as

$$\mathbf{f}_R(n) = \sum_m \omega^{mn} \cdot (W_V \cdot \mathbf{f}_A^m). \quad (2)$$

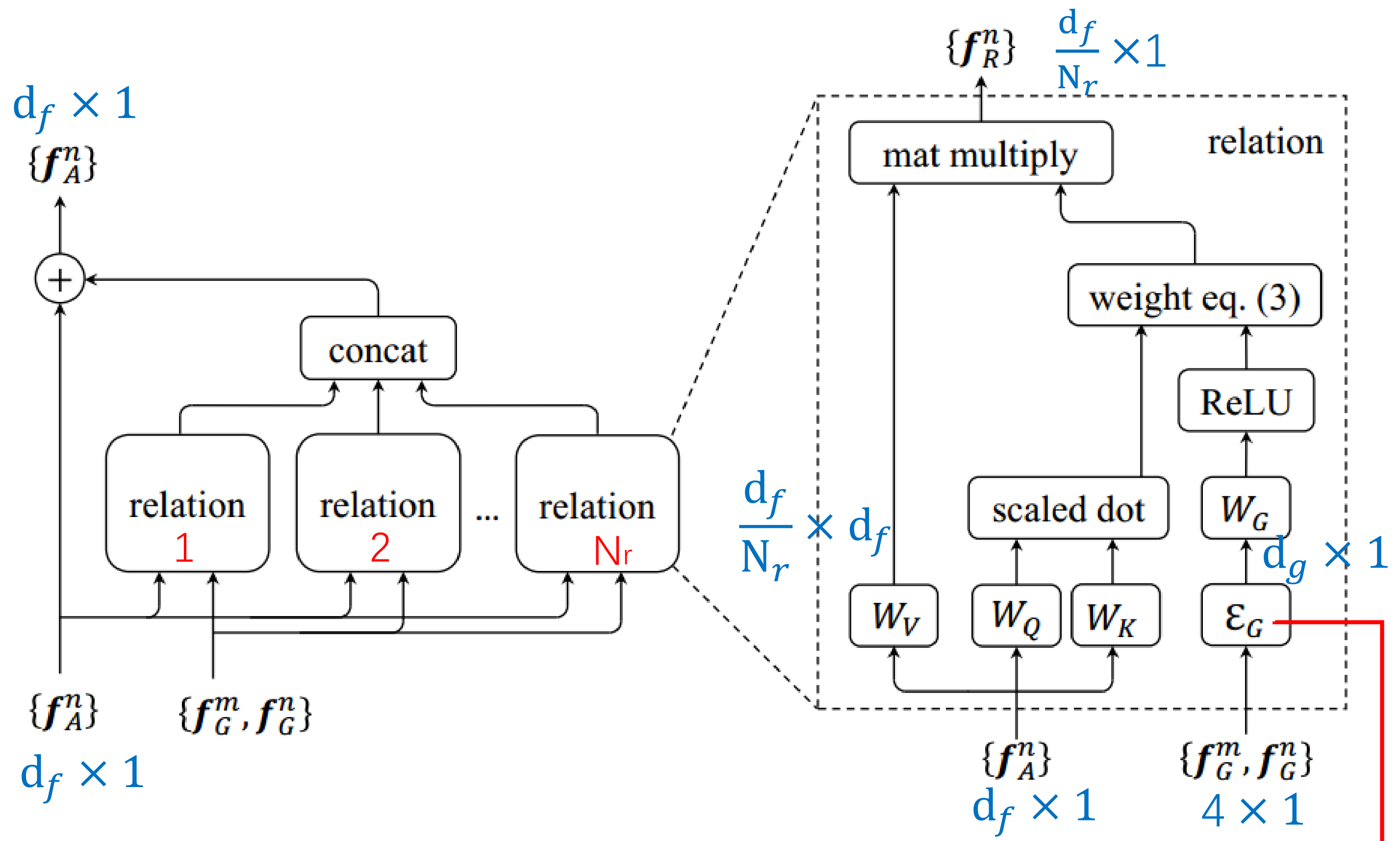


Figure 2. **Left**: object relation module as Eq. (6); **Right**: relation feature computation as Eq. (2).

这里是通过另一篇论文 (Attention Is All You Need) 中提到的方法将低维数据映射到了高维，映射后的维数为dg。

3. Relation for Instance Recognition

Given the RoI pooled features for n^{th} proposal, two fc layers with dimension 1024 are applied. The instance classification and bounding box regression are then performed via linear layers. This process is summarized as

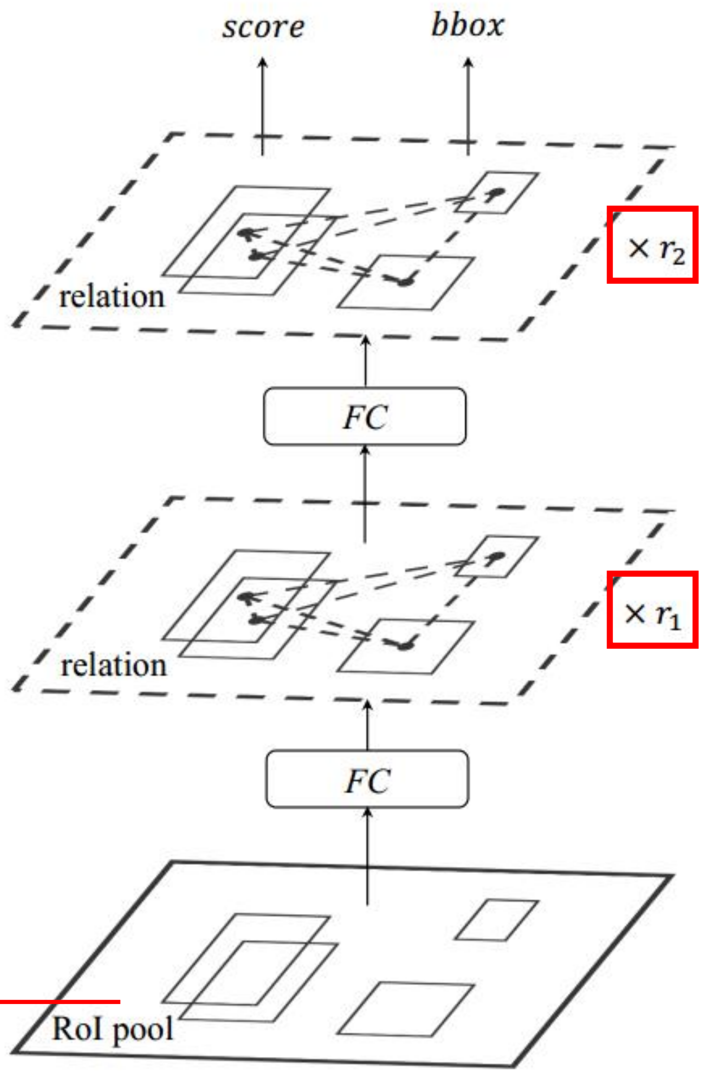
$$\begin{aligned} RoI\_Feat_n &\xrightarrow{FC} 1024 \\ &\xrightarrow{FC} 1024 \\ &\xrightarrow{LINEAR} (score_n, bbox_n) \end{aligned} \tag{9}$$

Such enhanced $\boxed{2fc+RM}$ (RM for relation module) head is illustrated in Figure 3 (a) and summarized as

$$\begin{aligned} \{RoI\_Feat_n\}_{n=1}^{N} &\xrightarrow{FC} 1024 \cdot N \xrightarrow{\{RM\}^{r_1}} 1024 \cdot \boxed{N} \\ &\xrightarrow{FC} 1024 \cdot N \xrightarrow{\{RM\}^{r_2}} 1024 \cdot N \\ &\xrightarrow{LINEAR} \{(score_n, bbox_n)\}_{n=1}^{N} \end{aligned} \tag{10}$$

In Eq. (10), $\boxed{r_1}$ and $\boxed{r_2}$ indicate how many times a relation module is repeated. Note that a relation module also

total **N** proposals
Faster RCNN[38]: N=300
DCN[10]: N=300
FPN[32]: N=1000
思考：对于SSD那种proposals特别多的检测框架不知道会不会太慢?



(a) enhanced *2fc* head

4. Relation for Duplicate Removal

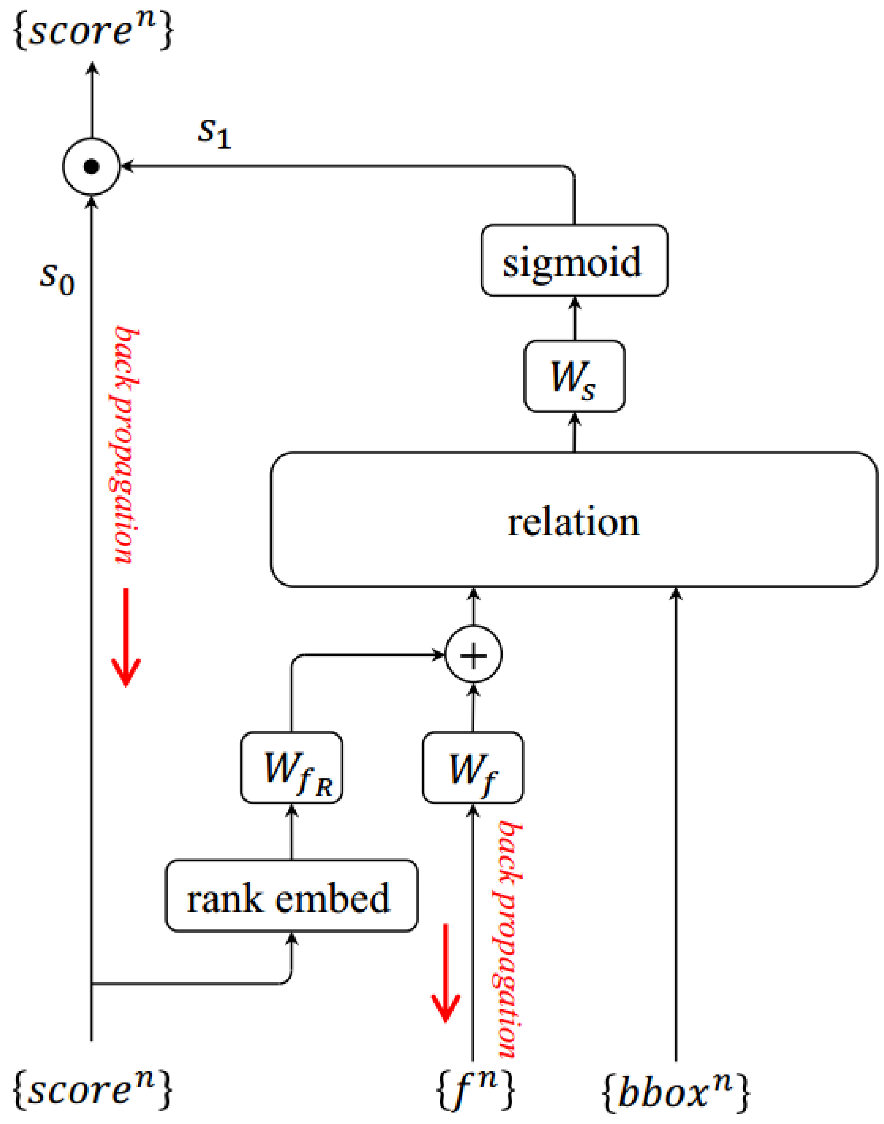
作者把duplicate removal当成一个二分类问题:
对于每个ground truth, **只有一个detected object被归为correct类，而其余的都是duplicate。**

模块的输入是instance recognition模块的output，也就是是一系列的detected objects，它们每个都有1024-d的特征，分类分数s0还有bbox。
模块的输出则是s0*s1得到的最终分类分数。

如何判断哪个detected object是correct，而哪些是duplicate?

predefined threshold η for the IoU between detection box and ground truth box, all detection boxes with IoU $\geq \eta$ are firstly matched to the same ground truth. The detection box with highest score is *correct* and others are *duplicate*.

与NMS相比的优势:
NMS需要一个预设置的参数；而duplicate removal模块是自适应学习参数的。
作者发现阈值的设置和最后的指标有某种联系。例如mAP0.5在**η** 为0.5时的效果最好，mAP0.75在**η** 为0.75时的效果最好



(b) duplicate removal network

5. Experiments

作者主要使用了ResNet 50和101，用两个fc层作为baseline进行了很多对比实验。

效果的提升是来自参数和层数的增加吗？ 表2对比了不同深度，宽度的网络结构。

head	mAP	mAP_{50}	mAP_{75}	# params	# FLOPS
(a) 2fc (1024)	29.6	50.9	30.1	38.0M	80.2B
(b) 2fc (1432)	29.7	50.3	30.2	44.1M	82.0B
(c) 3fc (1024)	29.0	49.4	29.6	39.0M	80.5B
(d) 2fc+res $\{r_1, r_2\}$=$\{1, 1\}$	29.9	50.6	30.5	44.0M	82.1B
(e) 2fc (1024) + global	29.6	50.3	30.8	38.2M	82.2B
(f) 2fc+RM $\{r_1, r_2\}$=$\{1, 1\}$	**31.9**	53.7	33.1	44.0M	82.6B
(g) 2fc+res $\{r_1, r_2\}$=$\{2, 2\}$	29.8	50.5	30.5	50.0M	84.0B
(h) 2fc+RM $\{r_1, r_2\}$=$\{2, 2\}$	**32.5**	54.0	33.8	50.0M	84.9B

Table 2. Comparison of various heads with similar complexity.

5. Experiments

本文提出的检测框去重复算法的优势体现在哪些方面？表4对比了本文方法与NMS和softNMS[4]的性能。

method	parameters	mAP	mAP_{50}	mAP_{75}
NMS	$N_t = 0.3$	29.0	51.4	29.4
NMS	$N_t = 0.4$	29.4	**52.1**	29.5
NMS	$N_t = 0.5$	29.6	51.9	29.7
NMS	$N_t = 0.6$	**29.6**	50.9	30.1
NMS	$N_t = 0.7$	28.4	46.6	**30.7**
SoftNMS	$\sigma = 0.2$	30.0	**52.3**	30.5
SoftNMS	$\sigma = 0.4$	30.2	51.7	31.3
SoftNMS	$\sigma = 0.6$	**30.2**	50.9	31.6
SoftNMS	$\sigma = 0.8$	29.9	49.9	**31.6**
SoftNMS	$\sigma = 1.0$	29.7	49.7	31.6
ours	$\eta = 0.5$	30.3	**51.9**	31.5
ours	$\eta = 0.75$	30.1	49.0	**32.7**
ours	$\eta \in [0.5, 0.9]$	**30.5**	50.2	32.4
ours (e2e)	$\eta \in [0.5, 0.9]$	**31.0**	51.4	32.8

Table 4. Comparison of NMS methods and our approach (Section 4.3). Last row uses end-to-end training (Section 4.4).

5. Experiments

端到端的目标识别

作者使用不同的检测框架，进行对比。

backbone	test set	mAP	mAP_{50}	mAP_{75}	#. params	FLOPS
faster RCNN [38]	*minival*	32.2→34.7→**35.2**	52.9→55.3→**55.8**	34.2→37.2→**38.2**	58.3M→64.3M→64.6M	122.2B→124.6B→124.9B
	test-dev	32.7→35.2→**35.4**	53.6→**56.2**→56.1	34.7→37.8→**38.5**		
FPN [32]	*minival*	36.8→38.1→**38.8**	57.8→59.5→**60.3**	40.7→41.8→**42.9**	56.4M→62.4M→62.8M	145.8B→157.8B→158.2B
	test-dev	37.2→38.3→**38.9**	58.2→59.9→**60.5**	41.4→42.3→**43.3**		
DCN [10]	*minival*	37.5→38.1→**38.5**	57.3→57.8→**57.8**	41.0→41.3→**42.0**	60.5M→66.5M→66.8M	125.0B→127.4B→127.7B
	test-dev	38.1→38.8→**39.0**	58.1→**58.7**→58.6	41.6→42.4→**42.9**		

Table 5. Improvement (2fc head+SoftNMS [4], 2fc+RM head+SoftNMS and 2fc+RM head+e2e from left to right connected by →) in state-of-the-art systems on COCO *minival* and *test-dev*. Online hard example mining (OHEM) [40] is adopted. Also note that the strong SoftNMS method ($\sigma = 0.6$) is used for duplicate removal in non-e2e approaches.

6. Summary

Relation module究竟学习到了什么?

作者提出的Relation module是一个很好的研究点，遗憾的是文中没有很好的解释Relation module学到了什么，作者说这个不在文章的讨论范围。为了对文章所提出的模型给出一个直观的解释，作者分析了Relation module中最后一个fc之后的RM中的关系权重，如下图所示，蓝色代表检测到的物体，橙色框和数值代表对该次检测有帮助的关联信息。

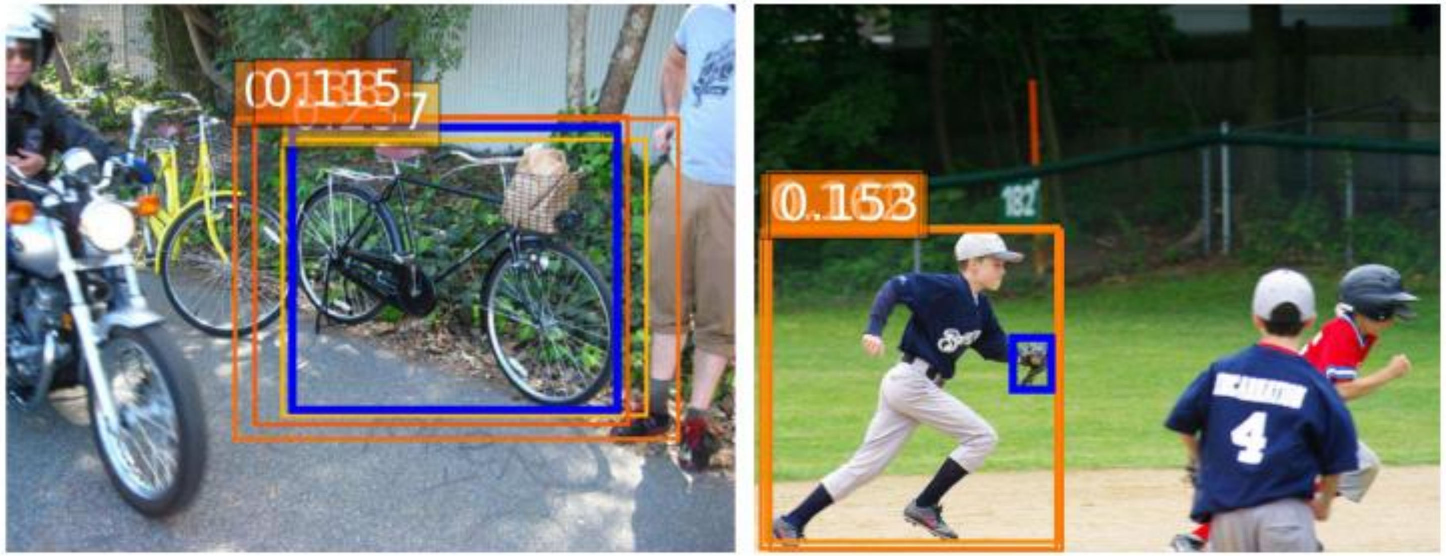


Figure 4. Representative examples with high relation weights in Eq. (3). The reference object n is blue. The other objects contributing a high weight (shown on the top-left) are yellow.