Independent Relationship Detection for Real-Time Scene Graph Generation

Tianlei Jin(✉), Wen Wang, Shiqiang Zhu, Xiangming Xi, Qiwei Meng, Zonghao Mu, and Wei Song(✉)

Zhejiang Laboratory, Intelligent Robot Research Center, Hangzhou, China
{jtl,wangwen,zhusq,xxm21,mengqw,muzonghao,weisong}@zhejianglab.com

Abstract. The current scene graph generation (SGG) task still follows the method of first detecting objects-pairs and then predicting relationships between objects-pairs. This paper introduces a parallel SGG thought that decouples relationship detection and object detection. In detail, we propose an independent visual relationship detection method, 'Relationship You Only Look Once' (RYOLO), which calculates relationships directly from the input image. For SGG, we present Similar Relationship Suppression and Object Matching Rules to match relationships and detected objects. In this way, the relationship detection and object detection can be calculated in parallel, and detected relationships can easily cooperate with detected objects to generate diversified scene graphs. Finally, our thought has verified the feasibility on the public Visual Genome dataset, and our method may be the first to attain real-time SGG.

Keywords: Relationship Detection · Scene Graph Generation · Relationship You Only Look Once

1 Introduction

Recently, computer vision has achieved great success in visual perceptual tasks, such as object detection. However, generating cognitive relationships from perceptual objects is still challenging. Scene graph generation (SGG) is an essential method for building relationship graphs between individual objects in the scene. In fact, the scene graph is often used as an introductory module to help high-level visual understanding tasks, such as image captioning [1], visual question answering [2], and visual grounding [3].

In the SGG task, one relationship between two objects can be represented as a triple: $<subject, relation, object>$, such as $<man, wear, shirt>$. Traditional SGG works [4–8] always relies on a series structure, as shown in Fig. 1(A). In the first stage, an image is fed into an object detection model to get object proposals, and in the second stage, a relationship prediction model is used to predict relationships based on these object proposals. In this case, some SGG works [5,8] rely on the two-stage object detection [9,10] to obtain intermediate

M. Tanveer et al. (Eds.): ICONIP 2022, CCIS 1791, pp. 106–118, 2023.
https://doi.org/10.1007/978-981-99-1639-9_9

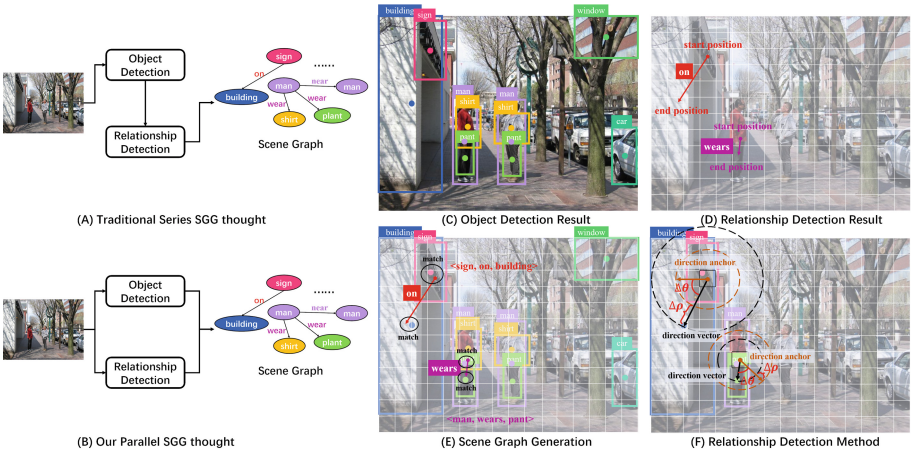

Fig. 1. (A) Traditional series SGG thought. (B) Our parallel SGG thought. (C) Independent object detection results calculate the center of the bounding box. (D) Our relationship detection results on the grids. (E) Object matching for scene graph generation. (F) The direction anchor for our relationship detection method.

features of object proposals through RoIAlign. Other SGG works [7,11] take the bounding box and category from the object detection as input to the predicted relationships. In general, the series structure makes the relationship prediction need wait for object detection processing, which restricts the real-time scene graph generation. In addition, each object-pair (subject and object) must perform one operation to predict the relationship, which is a quadratic number time complexity [4]. A large number of objects detected will seriously slow down the inference speed of SGG. However, for agents such as robots, the surrounding scene is changing in all the time, so a real-time SGG method is of great significance for rapid response of agents.

In this paper, we propose a parallel SGG thought with decoupling object detection and relationship detection for the real time SGG, shown in Fig. 1(B). In our thought, the object detection and the relationship detection are performed in parallel, and the scene graph is generated by combining the two results. For an image, we use a normal object detection method such as YOLOv5 [12] to get the position and category of objects as shown in Fig. 1(C). Meanwhile, we use another independent visual relationship detection method to predict the relationships existing in the image. Each predicted relationship contains a *start_position*, an *end_position* and a *relation_type*, as shown in Fig. 1(D). Afterward, we use the start and end positions to match the nearest object based on the object center positions from object detection results. Once the start and end positions are both matched by different objects, the triple relationship in the scene graph is established, as shown in Fig. 1(E). The object matched by the start position is the subject-object in the triple relationship, while the object matched by the end position is the object-object in the triple relationship, and the relation type is

the predicate. In this way, matching the nearest object is a batch minimum operation that replaces previous objects-pairs predictions and improves operational efficiency. In addition, because the visual relationship detection is independent, it can easily cooperate with any object detection model to generate scene graphs.

The key technology of our SGG thought is the visual relationship detection method. Inspired by the YOLO [13], we propose a novel independent visual relationship detection method RYOLO that can predict relationships from a whole image without intermediate steps such as object proposals [5,14] or knowledge embedding [15,16]. In detail, same as common visual tasks, an image is input into a backbone network and get a feature map. The feature map is composed of $W \times H$ grid cells. As shown in Fig. 1(F), for each grid cell, we preset some direction anchors in the polar coordinate. Each direction anchor contains an initial length ρ^{anchor} and an initial radian direction θ^{anchor}. One object can be assigned to a unique grid cell based on its center position. Then, the relationship of this object can be expressed as a direction vector from the grid cell, pointing to the center position of another object. Therefore, the neural network will predict direction offsets of the direction anchor $\Delta\rho$, $\Delta\theta$ and make direction anchors close to direction vectors after regulating by direction offsets. In addition, in order to get the relation type, the neural network will output a confidence score and relation type scores for each direction anchor while predicting direction offsets. Overall, our contribution can be summarized as:

- We propose a new parallel scene graph generation thought that decouples object detection and relationship detection. We verify the feasibility of this thought on the public dataset and achieve real-time scene graph generation.
- We propose an independent visual relation detection method RYOLO with preset direction anchors in the polar coordinate. RYOLO can cooperate with any object detection model to generate scene graph.

2 Related Work

From the perspective of the inputting information, we categorize recent SGG researches into three: First, using external knowledge. VRD [17] and UVTransE [15] introduce an external language model to embed word features of objects and relations. GB-NET [6] introduces an external commonsense knowledge graph into SGG. Second, using statistical context information from the dataset, VCTree [18] constructs tree structure with statistical information, while KERN [19] constructs knowledge graph. Third, only using the visual image, IMP [20], Graph R-CNN [4] and FCSGG [21] input the whole image and output the scene graph without additional operation. Our method also falls into this category.

From the perspective of the relationship prediction method, MOTIFS [5], VCTree [18], CogTree [22] construct grouping ordered objects structures and use RNN or LSTM to predict relationships. Graph R-CNN [4], GPS-Net [14] and KERN [19] construct graph structures and use graph neural networks to predict relationships. TDE [23] and PUM [24] put forward new modules, and

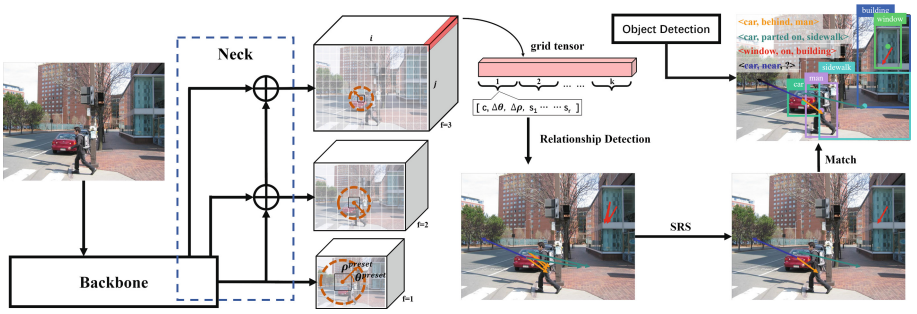

Fig. 2. The example of our relationship detection method RYOLO and scene graph generation method. We use the yolov5 backbone for feature extraction and generate multi-scale feature maps through the neck structure. Different direction anchors are preset on feature maps. Each grid tensor contains some clusters and each cluster contains a confidence, a direction offset and some relation type scores. The detected relationships generates a scene graph through similarity relationship suppression (SRS) and object matching.

improve the performance based on previous methods. However, these methods are all based on the series SGG structure: using Faster-RCNN to get object proposals or features, then objects-pairs feature finetunes iteratively for relationship prediction. This structure causes the relationship prediction heavily dependent on the object detection, and the inference speed is slow. Pixels2Graphs [25] and FCSGG [21] predict objects and relationships in one model which tends to the parallel structure. This paper, we propose a parallel SGG thought and introduce RYOLO to detect potential relationships with direction directly from images.

3 Method

3.1 Independent Relationship Detection

We redesign the output of an excellent object detection work yolov5 [12] and make it can be used for visual relationship detection. We call it 'Relationship You Only Look Once', RYOLO. As shown in Fig. 2, the whole image is input into the yolov5's backbone to extract features. The neck structure can generate multi-scale feature maps, and we finally get F different scale feature maps. We preset a set of direction anchors for each feature map, including ρ^{anchor} and θ^{anchor}, and direction anchors apply to all grid cells. On each grid cell of the feature map, the grid tensor outputs several clusters, which contains direction offsets $\Delta\rho$, $\Delta\theta$, confidence c of the relationship, and scores s of each relation type. We will introduce how to get the start position, the end position, and the relation type based on the direction anchor in detail.

Relationship Calculation. On the whole, we can get F feature maps, and we set $F = 3$ means 3 different scale feature maps. For each feature map f with

$f \in F$, it has $i \times j$ grid cells, while $i, j \in W_f, H_f$ and W, H are the scale of the feature map. The scale between the input image and the feature map f can be expressed as S_f. Each grid cell on the feature map can be the starting position:

$$start\_position_{fij} = i \times S_f, j \times S_f. \tag{1}$$

On each feature map f, we preset K direction anchors in the polar coordinate system, and we set $K = 6$ means 6 different directions. For each direction anchor d_{fk}^{anchor}, we can represent as $d_{fk}^{anchor} = [\rho_{fk}^{anchor}, \theta_{fk}^{anchor}], f \in F$ and $k \in K$ with $\rho_{fk}^{anchor} \in [0, 1]$ and $\theta_{fk}^{anchor} \in [0, 2\pi]$. ρ_{fk}^{anchor} is the normalized length in the polar coordinate system, while θ_{fk}^{anchor} is the radian direction. In our work, we divide the radian direction θ^{anchor} into K evenly for diverse directions. We hope to predict the short-distance relationship on the large-scale feature map while the long-distance relationship on the small-scale feature map. The larger the feature map, the smaller ρ^{anchor}. On the feature map f, each grid tensor also contains K predicted clusters. Each cluster contains the direction offset $\Delta d_{fkij} = [\Delta\rho_{fkij}, \Delta\theta_{fkij}]$ for the direction anchor $[\rho_{fk}^{anchor}, \theta_{fk}^{anchor}]$ on the grid cell i, j, which makes the direction anchor can point to the end position. Therefore, we can calculate the end position by the following formula:

$$\begin{aligned} d\_x_{fkij} &= \rho_{fk}^{anchor} \Delta\rho_{fkij} cos(\theta_{fk}^{anchor} + \Delta\theta_{fkij}) \\ d\_y_{fkij} &= \rho_{fk}^{anchor} \Delta\rho_{fkij} sin(\theta_{fk}^{anchor} + \Delta\theta_{fkij}), \end{aligned} \tag{2}$$

$$\begin{aligned} end\_position_{fkij} &= start\_position_{fij} + d_{fkij} \\ &= i \times S_f + d\_x_{fkij}, j \times S_f + d\_y_{fkij}, \end{aligned} \tag{3}$$

In fact, although we named $\Delta\rho$ direction offset, it is actually a scale factor to adjust the length ρ^{anchor}. $\Delta\theta$ add on θ^{anchor} to adjust the radian direction. In this way, each cluster can generate a predicted direction for connection between a start position and an end position. The confidence c in each cluster is used to judge whether the relationship is established. Each cluster also contains scores s independently of the R relation type, which is used to predict the relation type between the start position and end position.

Loss and Training. The loss consists of direction loss, confidence loss, relation loss. We first pick out the $start\_position_{fij}$ with the relationship, which can generate a direction vector d_{fij}^{vector}. For direction loss L_{dir}, the label direction offset $\Delta d_{fkij}^{label} = [\Delta\rho_{fkij}^{label}, \Delta\theta_{fkij}^{label}]$ from the direction anchor d_{fkij}^{anchor} to the direction vector d_{fij}^{vector} can be simply expressed as:

$$\Delta\rho_{fkij}^{label} = \rho_{fij}^{vector} / \rho_{fkij}^{anchor}, \Delta\theta_{fkij}^{label} = \theta_{fij}^{vector} - \theta_{fkij}^{anchor}, \tag{4}$$

We calculate the loss between the predicted direction Δd_{fkij}^{pred} and the label direction Δd_{fkij}^{label} through L2 loss, which hopes the predicted direction vector is equal to the label direction vector as much as possible.

For confidence loss L_{conf}, when the $start\_position_{fij}$ cannot generate direction vectors, we set c_{fij}^{label} to 0. But when there is a direction vector on the

$start\_position_{fij}$, we compare the distance between the direction vector and direction anchors. We retain direction anchors that close to the direction vector, and set the c_{fkij}^{label} to 1. Other direction anchors set c_{fkij}^{label} to 0.

For relation loss L_{rel}, relation types between subjects and objects are not unique in labels. For an object-pair, $<man, wears, shirt>$ and $<man, has, shirt>$ may both exist in the label. We set each relation type score s_{fkijr}^{label} to 1 if the relation type exists in the label for the corresponding cluster. We also use BCELoss to calculate relation loss L_{rel} and confidence loss L_{conf}.

In summary, the toal loss can be expressed as:

$$\begin{aligned} Loss &= L_{conf} + L_{rel} + L_{dir} \\ &\frac{1}{N_{conf}} \sum_{f=1}^{F} \sum_{k=1}^{K} \sum_{i=1}^{W} \sum_{j=1}^{H} BCE(c_{fkij}^{pred}, c_{fkij}^{label}) + \\ &\frac{1}{N_{rel}} c_{fkij}^{label} \sum_{f=1}^{F} \sum_{k=1}^{K} \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{r=1}^{R} BCE(s_{fkijr}^{pred}, s_{fkijr}^{label}) + \\ &\frac{1}{N_{dir}} c_{fkij}^{label} \sum_{f=1}^{F} \sum_{k=1}^{K} \sum_{i=1}^{W} \sum_{j=1}^{H} ||\Delta d_{fkij}^{pred}, \Delta d_{fkij}^{label}||_2, \end{aligned} \tag{5}$$

In the Eq. 5, N_{conf}, N_{rel} and N_{dir} are used to calculate the average of L_{conf}, L_{rel} and L_{dir}. Only when $c_{fkij}^{label} = 1$, the corresponding L_{rel} and L_{dir} are counted into loss.

During our following training, the size of input images is variable multi-scale similiar with yolov5, and the initial learning rate is 0.04. After 100k interations, the learning rate decay to 0.001. The model is trained end-to-end using SGD optimizer with the batch size of 32.

3.2 Scene Graph Generation

Similar Relationship Suppression. We can get many relationships through our relationship detection RYOLO, and each relationship contains a *start_position*, an *end_position*, and a *relation_type*. We design multiple direction anchors in multi-scale feature maps and multiple directions, but different direction anchors may predict the same relationships. We propose a similar relationship suppression (SRS) method during SGG. In detail, we filter out all relationships by threshold t^c, that $c_{fkij}^{pred} > t^c$. Then, we compare the relationship $<start\_position, end\_position, relation\_type>$ sorted by c^{pred} with rest relationships. During the comparison, SRS will suppress similar relationships. The suppression condition can be expressed as:

$$\begin{cases} start\_position_{fkij} - start\_position_{f'k'i'j'} < t^d \\ end\_position_{fkij} - end\_position_{f'k'i'j'} < t^d \\ relation\_type_{fkij} \neq relation\_type_{f'k'i'j'}, \end{cases} \tag{6}$$

Table 1. R@K and ng-R@K evaluation results on VG-150 dataset. G1, G2, G3 stands for group 1, 2, 3. The SGG works in group 1 draw on external knowledge and in group 2 draw on statistical context information. The works in group 3 only use visual images for SGG. Ours work falls into group 3.

	R@K Method		mAP @50	FPS	PredCls	SGCls	SGDet
					R@20/50/100	R@20/50/100	R@20/50/100
G1	GB-Net-β		–	1.92	–/66.6/68.2	–/37.3/38.0	–/26.3/29.9
	UVTransE		23.8	–	–/65.3/67.3	–/35.9/36.6	–/30.1/33.6
G2	KERN		–	1.27	–/65.8/67.6	–/36.7/37.4	–/27.1/29.8
	GPS-Net		–	–	67.6/69.7/69.7	41.8/42.3/42.3	22.3/28.9/33.2
	MOTIFS-TDE		–	1.15	33.6/46.2/51.4	21.7/27.7/29.9	12.4/16.9/20.3
	VCTree		–	0.75	60.1/66.4/68.1	35.2/38.1/38.8	22.0/27.9/31.3
G3	IMP		20.0	–	58.5/65.2/67.1	31.7/34.6/35.4	14.6/20.7/24.5
	Graph RCNN		23.0	5.26	–/54.2/59.1	–/29.6/31.6	–/11.4/13.7
	FCSGG		25.0	12.50	24.2/31.0/34.6	13.6/17.1/18.8	11.5/15.5/18.4
Ours	RYOLOs	+yolov5s	20.2	**35.71**	22.9/30.9/32.6	8.8/13.6/15.8	7.6/11.9/14.7
		+yolov5l	26.2	25.82		10.8/16.4/19.0	8.7/13.6/16.9
	RYOLOl	+yolov5s	20.2	22.42	**23.4/32.1/33.9**	8.9/13.8/16.1	7.6/12.0/15.0
		+yolov5l	26.2	17.59		**11.0/16.9/19.7**	**8.7/13.7/17.2**
	R@K Method		mAP @50	FPS	PredCls	SGCls	SGDet
					ng-R@20/50/100	ng-R@20/50/100	ng-R@20/50/100
G1	GB-Net-β		–	1.92	–/83.5/90.3	–/46.9/50.3	–/29.3/35.0
G2	KERN		–	1.27	–/81.9/88.9	–/45.9/49.0	–/30.9/35.8
	LSBR		–	–	77.9/82.5/90.2	43.6/46.2/50.2	26.9/31.4/36.5
G3	Pixels2Graphs		–	0.28	–/68.0/75.2	–/26.5/30.0	–/9.7/11.3
	FCSGG		25.0	12.50	28.1/40.3/50.0	14.2/19.6/24.0	12.7/18.3/23.0
Ours	RYOLOs	+yolov5s	20.2	**35.71**	29.1/42.1/50.8	9.3/13.6/17.2	10.0/14.7/18.4
		+yolov5l	26.2	25.82		11.7/17.4/21.7	12.0/17.7/21.9
	RYOLOl	+yolov5s	20.2	22.42	**29.9/43.3/52.1**	9.5/13.9/17.4	10.2/15.0/18.7
		+yolov5l	26.2	17.59		**12.0/17.8/22.0**	**12.2/18.0/22.3**

In suppression condition Eq. 6, $f'k'i'j' \neq fkij$, and t^d is the preset distance threshold. If relationships satisfy all three suppression conditions at the same time, low-confidence relationships will be suppressed, and only the highest-confidence relationship will be retained.

Object Matching. To generate a scene graph, we need to employ the results of object detection. In other word, we need to match relationships and objects on the image. The *start_position* or *end_position* will query all center coordinates of detected objects and find the nearest object for matching as the subject-object or object-object, shown in Fig. 1(C). During Object Matching, we use the simple distance threshold to eliminate failed matching between the *start_position* or *end_position* and the center point of objects. The *start_position* and the *end_position* cannot be matched by the same object.

Table 2. mR@K, zsR@K and zsR@K, ng-zsR@K evaluation results on VG-150 dataset.

mR@K zsR@K	PredCls		SGCls		SGDet	
Method	mR@	zsR@	mR@	zsR@	mR@	zsR@
	50/100	50/100	50/100	50/100	50/100	50/100
GB-NET-β	22.1/24.0	–	12.7/13.4	–	7.1/8.5	–
KERN	17.7/19.4	–	9.4/10.0	–	6.4/7.3	–
VCTree	17.9/19.4	–	12.7/13.4	–	7.1/8.5	–
CogTree	28.4/31.0	–	15.7/16.7	–	11.1/12.7	–
MOTIFS-TDE	25.5/29.1	14.4/18.2	13.1/14.9	3.4/4.5	8.2/9.8	2.3/2.9
VCTree-TDE	25.4/28.4	14.3/17.6	12.2/14.0	3.2/4.0	9.3/11.1	2.6/3.2
FCSSG	5.2/6.1	8.6/10.9	2.9/3.4	1.7/2.1	2.6/3.1	1.0/1.4
RYOLOs+yolov5s	5.3/5.9	6.9/7.4	2.3/2.9	1.2/1.5	2.0/2.6	0.8/0.9
RYOLOs+yolov5l			2.8/3.5	1.7/2.1	2.3/2.9	1.1/1.4
RYOLOl+yolov5s	5.7/6.3	7.3/7.9	2.4/3.1	1.2/1.4	2.1/2.7	0.7/0.9
RYOLOl+yolov5l			2.9/3.7	1.6/2.0	2.4/3.1	1.2/1.5
ng-mR@K ng-zsR@K	PredCls		SGCls		SGDet	
Method	ng-mR@	ng-zsR@	ng-mR@	ng-zsR@	ng-mR@	ng-zsR@
	50/100	50/100	50/100	50/100	50/100	50/100
FCSSG	9.5/14.7	12.8/19.6	6.3/9.4	2.9/4.4	4.7/6.9	1.8/2.7
RYOLOs+yolov5s	9.7/15.4	12.2/19.2	3.9/6.0	1.6/2.7	4.5/6.8	1.6/2.8
RYOLOs+yolov5l			5.2/7.8	**2.6/4.8**	5.4/8.2	**2.7/4.3**
RYOLOl+yolov5s	**10.2/16.1**	**12.5/19.6**	4.1/6.2	1.5/2.5	4.6/7.1	1.6/2.6
RYOLOl+yolov5l			**5.3/8.0**	2.4/4.1	**5.5/8.4**	2.5/4.1

4 Experiment

4.1 Dataset, Model and Metrics

Dataset. We train and evaluate our models on the public VG-150 [23]. VG-150 contains the most frequent 150 object categories and 50 predicate categories from the Visual Genome dataset [26].

Model. We decouple the relationship detection and object detection for SGG. For the object detection, we train independent model yolov5s and yolov5l with different backbone [12]. Similarly, for the relationship detection, we introduce our RYOLO method but use two backbone networks named RYOLOs and RYOLOl. Both Yolov5 and RYOLO are trained in the VG-150 dataset. We will show the impact of different performances object detection models and relationship detection models on SGG.

Metrics. We analyze our method on three standard SGG evaluation tasks: Predicate Classification (PredCls), Scene Graph Classification (SGCls), and Scene Graph Detection (SGDet). The PredCls task only needs to perform our

relationship detection and object information can be obtained from label. The SGCls and SGDet tasks need to employ the results of object detection. The conventional metric of SGG is Recall@K (R@K) [17]. Since predicates are not exclusive, Pixels2Graphs [25] proposes No Graph Constraint Recall@K (ng-R@K). Mean Recall@K (mR@K) [18] and No Graph Constraint Mean Recall@K (ng-mR@K) optimize the influence of high-frequency predicates. For verify generalization of SGG, Zero Shot Recall@K (zR@K) [17] and No Graph Constraint Zero Shot Recall@K (ng-zR@K) [23] count triplet relationships that not occurred in the training. In addition, the object detection metric mAP@50 [28] is also displayed as a reference.

4.2 Results and Discussion

Results Analysis. The results of our work are shown in Table 1 and Table 2. We decouple the dependence of relationship detection and object detection in traditional series SGG, so our relationship detection method RYOLO can deal with the PredCls task independently. The traditional SGG methods extract the features of specific objects based on the ground truth bounding box and category, and the predicted relationship is more accurate. RYOLO predicts the relationship from the whole image whether the object bounding box is known or not. The ground truth bounding box and category are only used in the object matching process.

We introduce an additional independent object detection method yolov5 to help RYOLO complete the SGCls and SGDet tasks. From the results, our method cannot bring accuracy improvements in SGG tasks. For a fair comparison with previous works, we divide previous works into three groups: using external knowledge, using statistical context information, and only using visual images. Our method is competitive in SGDet tasks compared with methods in G3. Compared to previous methods in R@K and mR@K metrics, our method performs similarly on SGCls and SGDet tasks. The main reason is that previous methods highly depend on object detection results, and biased detected objects drop performance from SGCls to SGDet. But our method is independent and detects relationships from the whole image. Biased detected object only affect object matching slightly. In the No Graph Constraint condition, each objects-pair can predict multiple possible relation types. Similar to other methods, RYOLO can recall more relationships in this condition.

As for our results of zsR@K and ng-zsR@K, these two metrics are used to judge whether the SGG method can predict the unseen triple relationship in training. They are not common in previous SGG evaluations, and there are limited references. Since our SGG method is independent, it is not restricted by the objects-pairs. From the ng-zsR@K results, our method can predict unseen triple relationships more than the latest fully convolutional scene graph generation method FCSGG. In addition, it seems that SGDet performs better than SGCls when no graph constraint. The reason is that object matching uses detected object positions before Non-Maximum Suppression in SGDet, rather than using ground truth object bounding box in SGCls.

Fig. 3. Visualization results of scene graph combined with OWOD [27] and our RYOLOs. In these examples, the color of the bounding box is the same as the color of the text label and each objects-pair only shows the highest-confidence relation.

Advantages. The advantages of our SGG method with parallel thought lie in its inference speed and flexibility. In terms of inference speed, our method has an obvious advantage. To compare the speed with the previous works under the same GPU condition [21], we also perform our method on an NVIDIA GeForce GTX 1080 Ti GPU with batch size 1. Using the combination of RYOLOs and yolov5s, our method realizes real-time scene graph generation (FPS > 30). Compare the fast SGG method FCSGG (HRNetW48-1s) [21], our method improves inference speed by nearly three times with less loss of precision, in Table 1. This is because we adopt a light full convolutional backbone network from yolov5 with less computation and our method supports the parallel structure and can simultaneously detect objects and relationships through multiple processes. In addition, traditional methods with objects-pairs relationship detection have a quadratic number time complexity, but RYOLO detects all relationships at the same time. Object matching is a batch operation to find the minimum position in the matrix operation, and it can maintain high-speed calculation. Furthermore, in the case of a single GPU, the parallel structure can not well reflect the advantages. Object detection yolov5 and relationship detection RYOLO, run in parallel only about 5% faster than running in series, as we show the inference time in Table 1. But nearly 30% faster with multiple GPUs in our experiment.

In terms of flexibility, thanks to the decoupling of object detection and relationship detection, RYOLO can easily cooperate with any object detection model to generate scene graphs. A better object detection model can reduce false detections and missed detections, and improve the accuracy of SGG. In Table 1, we can easily replace the different yolov5 models without retraining the relationship detection model. We believe that this independent method has a more comprehensive and wider practical application value. In addition, we try to replace yolov5 with open-world object detection OWOD [27] for the open-world scene graph generation. As shown in Fig. 3, OWOD can detect unknown objects and mark them as *unknown*. Similar to human cognition, humans may not recognize a new object but can analyze the relationship between this object and other objects. Combining OWOD and RYOLO in Fig. 3, we can generate some novel relationships, such as $<unknow, near, tv>$ and $<unknow, on, diningtable>$. Based on these relationships, the computer can infer unknown attributes through knowledge, such as the *unknown* object on the diningtable may be tableware.

Limitations. Our method sacrifices accuracy for speed and flexibility. As shown in Fig. 3, there are still some error relationships such as the relationship $<person, sitting\ on, car>$. Due to our parallel thought, relationships are detected directly from the whole image, the details of the objects themselves will be ignored. Although our relationship detection and object detection are independent, we rely on the consistency of the object center positions detected in the SGG. The current object matching only considers location information from object detection and relationship detection without contextual content. It cannot avoid some false matches. For example, a man and a shirt may fall on the same grid cell, and the nearest object matching may produce a wrong triple $<shirt, hold, cup>$ rather than $<man, hold, cup>$. In the long-distance direction prediction in the polar coordinate, a slight shift in the radian causes a huge error in the *end_position*.

5 Conclusion

In this paper, we rethink the methods of SGG and introduce a parallel SGG thought with an independent visual relationship detection method RYOLO. In RYOLO, we design direction anchors to directly predict relationships from the image without relying on object detection results. As for SGG, object detection and relationship detection results are correlated through object matching rules to generate triples and the scene graph. This way, we decouple object detection and relationship detection and realize real-time SGG. We expect our method can become a new baseline for the real-time scene graph generation. In the future, we will consider incorporating knowledge and statistical context information to improve the performance of real-time SGG.

Acknowledgement. The research was supported by the National Natural Science Foundation of China (Grant No. U21A20488) and the '10000 Talents Plan' of Zhejiang Province (Grant, No. 2019R51010). The research was supported by Lab-initiated Research Project of Zhejiang Lab (No. G2021NB0AL03).

References

1. Gu, J., Joty, S., Cai, J., Zhao, H., Yang, X., Wang, G.: Unpaired image captioning via scene graph alignments. In: CVPR (2019)
2. Hudson, D.A., Manning, C.D.: Learning by abstraction: the neural state machine. In: NIPS (2019)
3. Wan, H., Luo, Y., Peng, B., Zheng, W.: Representation learning for scene graph completion via jointly structural and visual embedding. In: IJCAI (2018)
4. Yang, J., Lu, J., Lee, S., Batra, D., Parikh, D.: Graph R-CNN for scene graph generation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11205, pp. 690–706. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01246-5_41

5. Zellers, R., Yatskar, M., Thomson, S., Choi, Y.: Neural motifs: scene graph parsing with global context. In: CVPR (2018)
6. Zareian, A., Karaman, S., Chang, S.-F.: Bridging knowledge graphs to generate scene graphs. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12368, pp. 606–623. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58592-1_36
7. Zhang, J., Kalantidis, Y., Rohrbach, M., Paluri, M., Elgammal, A., Elhoseiny, M.: Large-scale visual relationship understanding. In: AAAI (2019)
8. Lin, X., Li, Y., Liu, C., Ji, Y., Yang, J.: Scene graph generation based on node-relation context module. In: Cheng, L., Leung, A.C.S., Ozawa, S. (eds.) ICONIP 2018. LNCS, vol. 11302, pp. 134–145. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-04179-3_12
9. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS (2015)
10. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: ICCV (2017)
11. Gkanatsios, N., Pitsikalis, V., Koutras, P., Maragos, P.: Attention-translation-relation network for scalable scene graph generation. In: ICCV Workshops (2019)
12. Glenn-Jocher, et al.: yolov5 (2021). https://github.com/ultralytics/yolov5
13. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: CVPR (2016)
14. Lin, X., Ding, C., Zeng, J., Tao, D.: GPS-Net: graph property sensing network for scene graph generation. In: CVPR (2020)
15. Hung, Z., Mallya, A., Lazebnik, S.: Contextual translation embedding for visual relationship detection and scene graph generation. IEEE Trans. Pattern Anal. Mach. Intell. **43**(11), 3820–3832 (2020)
16. Gu, J., Zhao, H., Lin, Z., Li, S., Cai, J., Ling, M.: Scene graph generation with external knowledge and image reconstruction. In: CVPR (2019)
17. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 852–869. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_51
18. Tang, K., Zhang, H., Wu, B., Luo, W., Liu, W.: Learning to compose dynamic tree structures for visual contexts. In: CVPR (2019)
19. Chen, T., Yu, W., Chen, R., Lin, L.: Knowledge-embedded routing network for scene graph generation. In: CVPR (2019)
20. Xu, D., Zhu, Y., Choy, C.B., Fei-Fei, L.: Scene graph generation by iterative message passing. In: CVPR (2017)
21. Liu, H., Yan, N., Mortazavi, M., Bhanu, B.: Fully convolutional scene graph generation. In: CVPR (2021)
22. Yu, J., Chai, Y., Wang, Y., Hu, Y., Wu, Q.: CogTree: cognition tree loss for unbiased scene graph generation. In: IJCAI (2021)
23. Tang, K., Niu, Y., Huang, J., Shi, J., Zhang, H.: Unbiased scene graph generation from biased training. In: CVPR (2020)
24. Yang, G., Zhang, J., Zhang, Y., Wu, B., Yang, Y.: Probabilistic modeling of semantic ambiguity for scene graph generation. In: CVPR (2021)
25. Newell, A., Deng, J.: Pixels to graphs by associative embedding. In: NIPS (2017)
26. Krishna, R., et al.: Visual genome: connecting language and vision using crowdsourced dense image annotations. Int. J. Comput. Vis. **123**(1), 32–73 (2017). https://doi.org/10.1007/s11263-016-0981-7

27. Joseph, K.J., Khan, S., Khan, F., Balasubramanian, V.: Towards open world object detection. In: CVPR (2021)
28. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal visual object classes challenge: a retrospective. Int. J. Comput. Vis. **111**(1), 98–136 (2014). https://doi.org/10.1007/s11263-014-0733-5