# Joint COCO and Mapillary Workshop at ICCV 2019: COCO Panoptic Segmentation Challenge Track

Technical Report: Generator evaluator selector net for image segmentation

Sagi Eppel
Vector Institute and Deparment of Chemistry
University of Toronto, Canada
sagieppel@gmail.com

Alan Aspuru-Guzik
Departments of Chemistry and Computer Science
University of Toronto, Canada
alan@aspuru.com

## Abstract

*In machine learning and other fields, suggesting a good solution to a problem is usually a harder task than evaluating the quality of such a solution. This asymmetry is the basis for a large number of selection oriented methods that use a generator system to guess a set of solutions and an evaluator system to rank and select the best solutions. This work examines the use of this approach to the problem of image segmentation. The generator/evaluator approach for this case consists of two independent convolutional neural nets: a generator net that suggests variety segments corresponding to objects and stuff regions in the image, and an evaluator net that chooses the best segments to be merged into the segmentation map. The result is a trial and error evolutionary approach in which a generator that guesses segments with low average accuracy, but with wide variability, can still produce good results when coupled with an accurate evaluator. The generator consists of Pointer net that given an image and a point in the image predicts the region of the segment containing the point. Generating and evaluating each segment separately is essential in this case since it demands exponentially fewer guesses compared to a system that guesses and evaluates the full segmentation map in each try. The classification of the selected segments is done by a separate net. This allows the segmentation to be class agnostic and hence capable of segmenting unfamiliar categories that were not part of the training set. Code has been made available at: this https url*

## 1. Introduction

Systems that consist of a generator process that generates a variety of random products, and a selector process that filters the products according to some property, appear in a wide range of fields [7, 2, 10]. This work will examine the use of a generator selector approach [5] for the task of panoptic image segmentation [8]. This will be done by combining a generator net that guesses various segments corresponding to objects (things [9]) and none object regions (stuff [1]), and an independent evaluator net that ranks and selects the best segments to be used in the final segmentation map (Figure 1). The generator consists of Pointer net [4], that given an image and a point in the image, finds the segment that contains the input point (Figure 2).

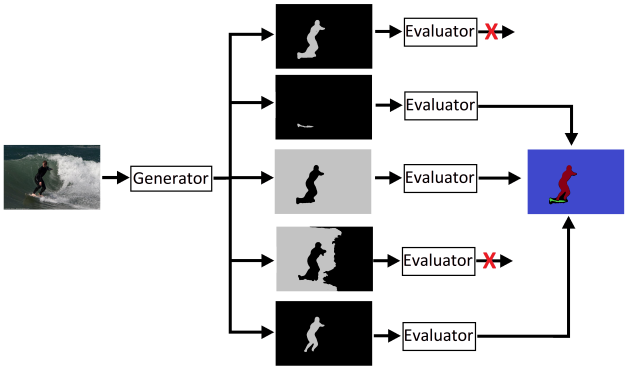


Figure 1: The generator suggests segments while the evaluator rank and select segments for the segmentation map.

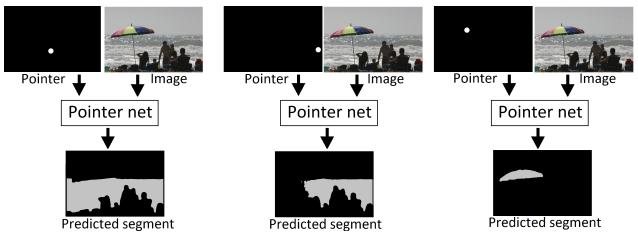


Figure 2: Pointer net as a segments generator. Given an image and a pointer point, the net predicts the segment containing the point.

The variability in the predicted segments emerges from the location of the input pointer point. Pointer net will produce different predictions even for different points located in the same segment (Figure 2). The evaluator system consists of a simple CNN that receives an image and a segment mask and returns the segment grade. Both the generator and evaluator are category agnostic [4]. To classify each segment, a region-specific classification net is used [3, 11]. This net receives the segment mask and the image, and returns the segment category.

## 2. Method details

A schematic for the full modular system is shown in Figure 3. The method is comprised of four independent neural nets combined into one modular structure. The first step is generating several different segments using the generator (pointer net). The segments generated by this net, are restricted to a given region of interest (ROI) which covers the unsegmented image region. The generated segments are then ranked by the evaluator net. This net assigned each segment a score that estimates how well it corresponds to a real segment in the image. The segments which receive the highest scores and are consistent with each other are selected, while low-ranking segments are filtered out. The selected segments are then polished using the refinement net. Each of the selected segments is then classified using the classifier net. Finally, the selected segments are stitched into the segmentation map (Figure 3). The segmentation map is passed to the next cycle which repeats the process in the remaining unsegmented image regions. The process is repeated until either the full image has been segmented or the quality assigned to all of the predicted segments by the evaluator drop below some threshold.

### 2.1. Pointer net

Pointer net [4] act as the segment generator, which creates proposals for different segments in the image (Figure 4.b). Pointer net receives an image and a point within this image. The net predicts the mask of the segment that contains the input point. In this work, the pointer point location is chosen randomly within the unsegmented region of the image. The net will predict different segments for different input points, even if the points are located within the same segment. This allows the pointer net to act as a random segment generator. Another input of the pointer net is a region of interest (ROI) mask which restricts the region of the predicted segments (Figure 4.b). The generated output segment region will be confined to the ROI mask. This property prevents newly generated segments from overlapping previously generated segments. The ROI mask is simply the unsegmented region of the image.

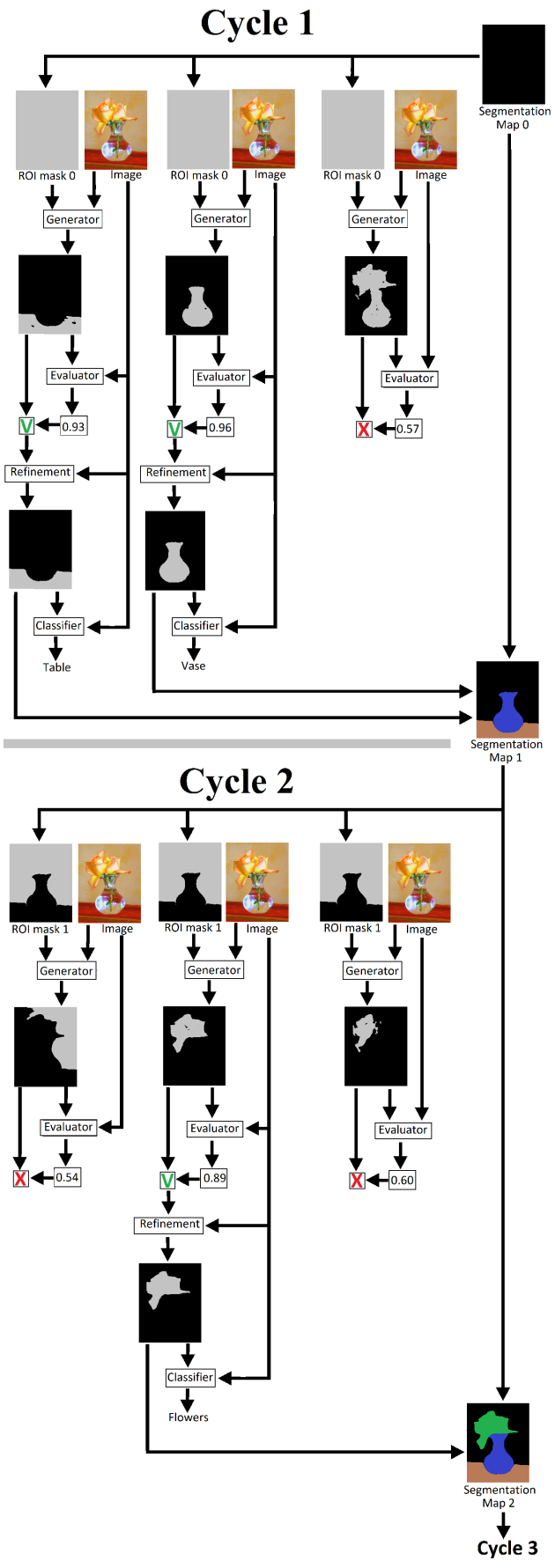


Figure 3: Schematic of the modular generator evaluator selector segmentation system

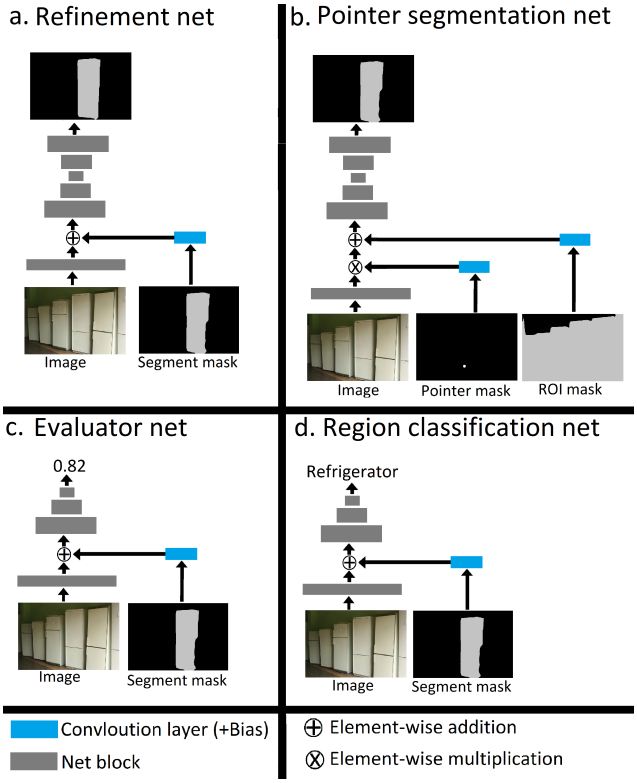


Figure 4: The nets used in the system

### 2.2. Evaluator net

The evaluator net is used to check and rank the generated segments. The ranking is done according to how well the input segment fits the best matching real segments in the image. The evaluator net is a simple convolutional net that receives two inputs: an image and a generated segment mask (Figure 4.c). The evaluator net predicts the intersection over union (IOU) between the input segment and the closest real segment in the image.

### 2.3. Refinement net

Refinement net is used to polish the boundaries of the generated segment. The net receives the image and an imperfect segment mask. The net output is a refined version of the input segment (Figure 4.a).

### 2.4. Classifier net

Determining the segment category is done using a region-specific classification net. The net receives the image and a segment mask. The net predicts the category of the input segment (Figure 4.d). This approach has been explored in previous research [3, 11]].

## 3. Architecture

All the nets used in this work are based on a standard CNN with ResNet architecture [6]. The pointer and refinement net are both based on the pyramid scene parsing (PSP) net [12]. The evaluator and classification nets (Figure 4.c-d) are based on the standard ResNet 101 model. For the evaluator net, the final layer was changed into a single channel prediction, which corresponds to the predicted IOU value (Figure 4.c). Adding the segment mask (or ROI mask) as an additional input to the nets was done by taking this mask and processing it using a single convolutional layer (Figure 4). The output of this process was merged with the feature map of the ResNet first layer using an element-wise addition (Figure 4). Similarly, the pointer point input for the pointer net [4] was introduced by representing the point as a binary mask, where the value of the cell in the pointer point location is one and the rest of the cells are zero (Figure 4.b). This pointer mask was processed using a single convolutional layer, and the output feature map was merged with the feature map of the ResNet first layer using an element-wise multiplication (Figure 4.b).

## 4. Training

Each of the above nets was trained separately using standard training methods. The training data for the pointer net was created by picking random segments from the annotation of the COCO panoptic training dataset. Segments of things (people, cars) were taken as the full object instance mask. Segments of stuff (sky, grass) were taken as the connected component region of pixels with the same class. Pointer points input for the pointer net were picked by selecting random points within the selected segments. ROI mask input for the pointer net was generated by picking a random segment from the annotation map and using their combined region as the ROI mask. For half of the pointer net training iterations, an ROI mask that covers the full image area was used. Training data for the refinement, evaluation and classification nets were generated by running various versions of pointer nets on the COCO panoptic training set [5]. The training loss for the pointer, classification and refinement nets was the standard cross-entropy. The loss for the evaluator net was the square difference between the predicted and the real IOU of the input segment and closest match segment in the ground truth annotation. Each of the nets was trained on a single TITAN XP GPU for about 1–2 million interactions. The full system composed of the four networks run in half-precision on a single TITAN XP.

## 5. Results

The results of the full generator-evaluator-selector system are given in Figures 5 and Table 1. To examine the effect of the evaluator, the system was run with no evalu-

ator (all segments were approved). The result is a significant decrease of 9 points in the PQ score (Table 1). This confirms the importance of the evaluator/selector module. To examine the maximum effect of the evaluator, the system was run with a perfect evaluator. The perfect evaluator was simulated by replacing the IOU score predicted by the evaluator with the real IOU extracted from the ground truth annotation. This increased the PQ score by 6 points (Table 1). To examine the contribution of the refinement net, the system was run without the refinement stage. This results in a drop of 1.5 points in the PQ score (Table 1). In order to examine the effect of misclassification, the category generated by the classification net was replaced by the segment real class. The result is a significant increase of 12.7 points in the PQ score (Table 1). This implies that misclassification is a major source of errors in the system. This PQ increase is particularly large for stuff categories implying misclassification is a bigger problem for none-object classes.


Figure 5: Sampled results of the system versus ground truth annotation

**Acknowledgments:** We thank Natural Resources Canada for their generous support for this project. We also thank Anders G. Froseth for his support of our work on artificial intelligence. We thank the Vector Institute for computational resources. We also thank the Office of Naval Research for support.

| Method | Set | PQ | RQ | SQ | $PQ^{St}$ | $RQ^{St}$ | $SQ^{St}$ | $PQ^{Th}$ | $RQ^{Th}$ | $SQ^{Th}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| **Full system** | **Test** | **33.7** | **41.4** | **79.6** | **31.5** | **39.3** | **78.4** | **35.1** | **42.9** | **80.4** |
| Full system | Eval | 33.2 | 40.9 | 79.2 | 30.5 | 38.0 | 78.1 | 34.9 | 42.8 | 80.0 |
| No evaluator | Eval | 24.5 | 30.8 | 76.7 | 27.0 | 34.0 | 76.6 | 22.8 | 28.7 | 76.7 |
| Perfect Evaluator | Eval | 39.3 | 48.8 | 78.3 | 35.3 | 44.8 | 77.2 | 41.9 | 51.5 | 79.0 |
| No refinement | Eval | 31.7 | 39.7 | 78.2 | 28.9 | 36.4 | 77.6 | 33.6 | 41.8 | 78.6 |
| Perfect classification | Eval | 45.9 | 56.6 | 79.8 | 54.1 | 66.9 | 80.0 | 40.4 | 49.9 | 79.7 |

Table 1: Results on the COCO panoptic dataset. Th and St stand for things and stuff

## References

[1] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and stuff classes in context. In *CVPR*, 2018. 1

[2] Donald T Campbell. Blind variation and selective retentions in creative thought as in other knowledge processes. *Psychological review*, 67(6):380, 1960. 1

[3] Sagi Eppel. Classifying a specific image region using convolutional nets with an roi mask as input. *arXiv preprint arXiv:1812.00291*, 2018. 2, 3

[4] Sagi Eppel. Class-independent sequential full image segmentation, using a convolutional net that finds a segment within an attention region, given a pointer pixel within this segment. *arXiv preprint arXiv:1902.07810*, 2019. 1, 2, 3

[5] Sagi Eppel and Alan Aspuru-Guzik. Generator evaluator-selector net: a modular approach for panoptic segmentation. *arXiv preprint arXiv:1908.09108*, 2019. 1, 3

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[7] David L Hull, Rodney E Langman, and Sigrid S Glenn. A general account of selection: Biology, immunology, and behavior. *Behavioral and brain sciences*, 24(3):511–528, 2001. 1

[8] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019. 1

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1

[10] Chris J Maddison, Aja Huang, Ilya Sutskever, and David Silver. Move evaluation in go using deep convolutional neural networks. *arXiv preprint arXiv:1412.6564*, 2014. 1

[11] S Nowee. Directing attention of convolutional neural networks. 2, 3

[12] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 3