A COMPOSITIONAL APPROACH TO OCCLUSION IN PANOPTIC SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper concerns image segmentation, with emphasis on correctly classifying objects that are partially occluded. We present a novel approach based on compositional modeling that has proven to be effective at classifying separate instances of foreground objects. We demonstrate the efficacy of the approach by replacing the object detection pipeline in UPSNet with a compositional element that utilizes a mixture of distributions to model parts of objects. We also show extensive experimental results for the COCO and Cityscapes datasets. The results show an improvement of 2.6 points in panoptic quality for the top "thing" classes of COCO, and a 3.43% increase in overall recall, using standard UPSNet as a baseline. Moreover, we present qualitative results to demonstrate that improved metrics and datasets are needed for proper characterization of panoptic segmentation systems.

1 INTRODUCTION

Panoptic image segmentation has emerged in recent years as an important visual recognition task (Kirillov et al., 2019b). The goal is to assign a label to each pixel of an image, so that some labels represent amorphous background regions and other labels indicate separate instances of foreground objects. Panoptic segmentation therefore balances the need to identify semantic background portions of an image while simultaneously identifying countable instances of individual objects in the foreground.

Occlusion presents a problem that must be addressed by panoptic segmentation systems, especially for crowded scenes. Although occlusion has been studied extensively (e.g., Koller et al. (1994); Elgammal & Davis (2001); Sun et al. (2005); Hoiem et al. (2007); Wang et al. (2009); Enzweiler et al. (2010); Kortylewski et al. (2020b)), the problem continues to represent a difficult hurdle for many visual analysis tasks. Example images are provided in Figure 1 to illustrate the complexity of the problem. In these examples, some of the foreground objects are significantly occluded by other foreground objects. The segmentation problem is exacerbated when an object is occluded

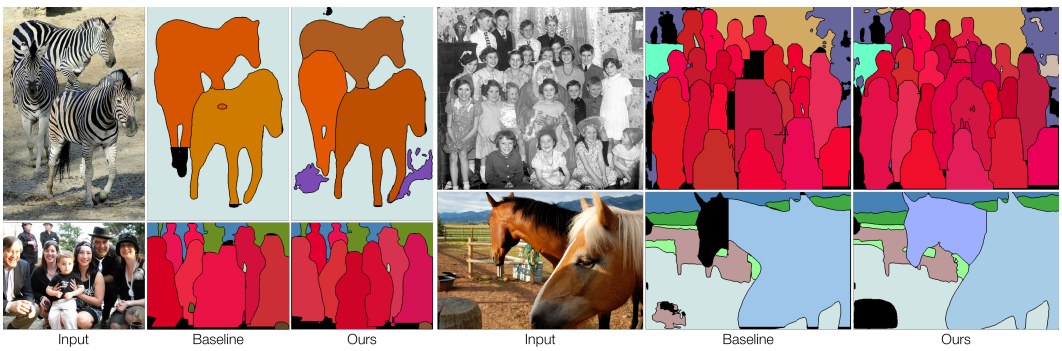

Figure 1: Example panoptic segmentation results on images from the COCO dataset (Lin et al., 2014). The baseline system has incorrectly merged several of the foreground objects (center). In contrast, our system was able to distinguish those cases through its compositional approach (right).

by another instance of the same category, which makes it difficult to generate accurate masks and recognize each case as a separate instance. For the examples shown here, the baseline system was not able to distinguish all of the foreground objects, with the result that some of the detected regions have been merged incorrectly. However, the new approach that is presented in this paper was able to detect the separate foreground regions correctly.

In order to improve performance when occlusion is present, one might argue that strategies involving larger datasets or improved data-augmentation techniques should be adequate. However, experiments by Kortylewski et al. (2020a) reveal that these techniques alone are not sufficient to improve performance for occlusion-related problems. As an alternative, Kortylewski et al. (2020b) have argued that *compositional models* may offer a suitable approach to addressing difficulties related to occlusion. Our work demonstrates that such a compositional approach can indeed improve detection of occluded objects.

Compositionality (Bienenstock et al., 1997; Geman et al., 2002) refers to an ability to represent entities as hierarchies of constituent parts, with the parts themselves being meaningful, reusable entities. The compositionality property has been shown to be beneficial in tackling several high-level tasks. For example, neuroscientific research shows that human cognition uses compositionality for object classification so that severely occluded objects can be detected and recognized in many situations (Bienenstock et al., 1997; Fodor & Pylyshyn, 1988). For computational systems, compositional models can recursively represent object parts and facilitate decisions that contribute to the final result. Several researchers, for example, have successfully employed compositional models for image classification tasks (e.g., George et al. (2017); Kortylewski (2017); Wang et al. (2017); Zhang et al. (2018)).

In the work presented here, we demonstrate that a compositional approach leads to improved image segmentation for foreground objects that may be partially occluded. To this end, we have integrated a compositional encoding component into UPSNet (Xiong et al., 2019), which is an adaptation of the Mask R-CNN (He et al., 2017) framework for the panoptic segmentation task. The system consists of a feature extraction backbone that is made up of the popular ResNet (He et al., 2016) architecture, followed by a Feature Pyramid Network (FPN) module that computes features at multiple scales. The resulting multiscale feature maps serve as input to two task-specific heads that separately perform instance segmentation and semantic segmentation. The semantic head is a series of deformable convolution layers that produce class predictions for each pixel in the input. The instance head consists of parallel branches for classification, bounding box regression and segmentation mask prediction. The outputs of the semantic and instance heads are then combined within a panoptic head to produce a single pixel-wise segmentation output.

We have tested the performance of our approach on two popular datasets for panoptic segmentation: COCO (Lin et al., 2014) and Cityscapes (Cordts et al., 2016). We show both qualitative and quantitative improvements on the COCO panoptic dataset relative to the UPSNet baseline. In summary, the major contributions of this paper are as follows:

- To our knowledge, this work is the first to apply a compositional approach to the task of panoptic image segmentation.
- We describe a modular implementation of compositional models using a generalized mixture model that can be plugged in any two-stage detection network. The approach has been tested using the UPSNet design for panoptic segmentation, but is broadly applicable. For example, our modular approach is applicable to the problems of object detection and classification directly.
- We propose a new training strategy for the instance segmentation head of panoptic segmentation systems, using available ground-truth mask information to learn object representations.

In addition, we present qualitative results that highlight some limitations of current datasets. We propose a refinement to the panoptic segmentation metric that accounts for noisy labelling of datasets. With this refinement, we demonstrate significant gains in performance over the baseline for all panoptic segmentation metrics.

2 RELATED WORK

Panoptic segmentation, as introduced by Kirillov et al. (2019b), integrates the problems of instance segmentation and semantic segmentation to produce a single output that contains class labels and an instance identifier for each pixel. By convention, "thing" classes refer to foreground objects, and "stuff" classes indicate background portions of an image. Virtually all panoptic segmentation systems that have been proposed can be categorized broadly as either *top-down* or *bottom-up*. The work being presented here falls into the top-down category.

The top-down approach first predicts bounding boxes around objects and then generates a segmentation mask to associate each pixel that lies within the mask with its corresponding class score. Many approaches of this type, such as (Li et al., 2019; Porzi et al., 2019; Yang et al., 2020; Lazarow et al., 2020) use Mask R-CNN (He et al., 2017) as a foundation and propose novel additions to improve segmentation performance. Kirillov et al. also propose a strong baseline called Panoptic FPN (Kirillov et al., 2019a) that is formed by adding a semantic segmentation branch to Mask R-CNN. AUNet (Li et al., 2019) joins the two heads by adding attention mechanisms to produce coherent outputs and to minimize conflict between foreground instance predictions and background predictions. Petrovai & Nedevschi (2019) propose a panoptic segmentation network for automated driving that combines the benefits of PSPNet (Zhao et al., 2017) for semantic segmentation and Mask R-CNN for instance segmentation. Liu et al. (2019) propose a Spatial Ranking module to resolve conflicts between overlapping instance masks. Some transformer based approaches include (Dai et al., 2021; Wang et al., 2021).

Bottom-up methods make class predictions for each pixel before using grouping strategies to detect and localize instances. AdaptIS (Sofiiuk et al., 2019) first performs semantic segmentation by generating pixel-wise predictions, and then uses point proposals to generate instance masks. Li et al. (2018b) use weak supervision in the form of bounding boxes and image-level tags to produce non-overlapping instance masks while performing semantic segmentation. FPSNet (de Geus et al., 2020) prioritizes runtime and attempts to perform both instance and semantic segmentation as a fully convolutional model by predicting a class score and an instance ID for each pixel. Hou et al. (2020) present a single-shot fully convolutional panoptic segmentation strategy. Li et al. (2021); Hong et al. (2021) take a fully convolutional approach to panoptic segmentation.

There have been several efforts to develop segmentation algorithms with emphasis on addressing the problem of occlusion. Tighe et al. (2014) first obtain pixelwise labels and a set of candidate object instances for hundreds of object classes. Overlapping regions are then used to obtain an occlusion ordering using a graph-theoretical approach to achieve an output that "explains" the image contents. Chen et al. (2015) collect segmentation masks and class scores for objects that are possibly occluded and formulate them into an energy minimization framework. Wang et al. (2018) introduce "repulsion loss," which tries to force bounding box regressors to move toward the correct target while also being repelled by other nearby target proposals. The panoptic segmentation model proposed by Lazarow et al. (2020) has a design identical to Kirillov et al. (2019a), with a ResNet FPN backbone followed by two task-specific heads, Mask R-CNN for instance tasks and FCNs for semantic tasks. Other approaches that emphasize occlusion have been presented by Zhu et al. (2017), Liu et al. (2019), Zhan et al. (2020), and Yang et al. (2020).

3 APPROACH

This section introduces an instance detection strategy that is based on a compositional model, and explains how this model has been integrated into the Mask R-CNN pipeline for improved instance segmentation. The proposed architecture is shown in Figure 2.

3.1 SHARED BACKBONE AND SEMANTIC HEAD

The backbone network of our architecture is identical to Mask R-CNN (He et al., 2017), which is a combination of ResNet (He et al., 2016) and FPN (Lin et al., 2017). The spatial dimensions of the input image are $H \times W$ pixels. The semantic head is fully convolutional, based on deformable convolution (Dai et al., 2017) as adopted from Xiong et al. (2019). It consists of one $3 \times 3 \times 256$ convolution layer followed by two $3 \times 3 \times 128$ convolution layers.

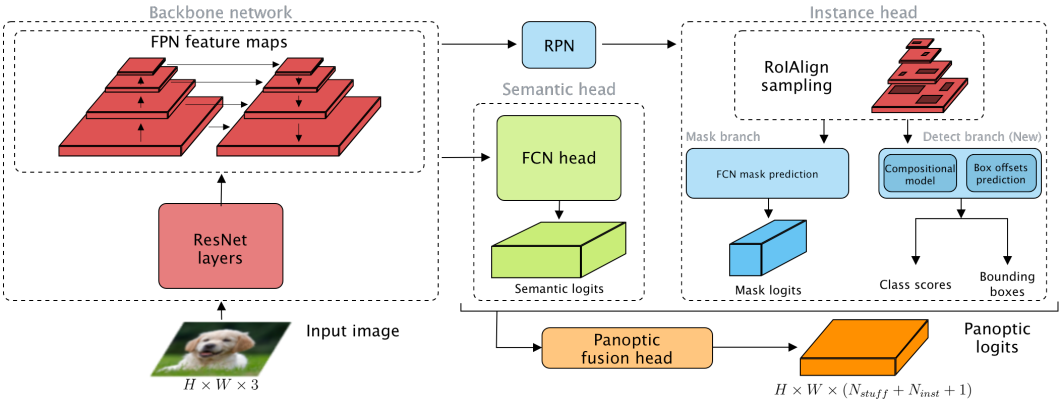

Figure 2: The high-level architecture of our panoptic segmentation system, which is adapted from UPSNet (Xiong et al., 2019). Within the instance head at the right, our design replaces the instance detection pipeline from Mask R-CNN (He et al., 2017) with a new element based on a compositional model (indicated by *New* in the diagram).

3.2 Detection using a Compositional Model

To perform instance segmentation, the second stage of Mask R-CNN consists of two branches. One branch performs detection, producing class scores and bounding box offsets for each region of interest (RoI). The second branch generates segmentation masks for each of the highest-scoring RoIs. Let us refer to these as the *detect branch* and the *mask branch*, respectively. The usual detect model within Mask R-CNN is a standard multilayer perceptron (MLP). A major innovation in our design is the insertion of a generative compositional model before these dense layers. The goal is to incorporate a higher degree of spatial awareness into the classification process.

Our modified detect branch is shown in Figure 3. The input to this branch is a feature tensor for a particular RoI, as represented by the orange block in the figure. (Each feature tensor is extracted using RoIAlign (He et al., 2017), for regions of interest selected by the Region Proposal Network, or RPN (Ren et al., 2015), within the shared backbone.) We denote this tensor as $f \in \mathbb{R}^{k \times k \times D}$, where $k \times k$ represents the spatial dimensions of the RoI lattice, and D is the number of channels of the feature map. For a fair comparison, we use the same values as previous designs for k and D.

To determine a class prediction, our strategy is to compute a tensor of posterior probability values from the given feature map f, scale those values using an attention map, and finally produce a vector containing a distribution of class scores for the given RoI. Let $f_{i,j}$ represent a particular D-dimensional feature vector at position (i, j) within the $k \times k$ lattice. We implement the compositional approach by assuming that any given $f_{i,j}$ can be represented using a weighted set of M components. To simplify the model, we consider each lattice position (i, j) to be independent of the rest. Let $v \in \mathbb{R}^M$ represent a vector of contribution levels by constituent components. The contribution of a particular component is written as v_m, for $m = 1, 2, ..., M$.

For a given object instance, assume that the probability of occurrence of $f_{i,j}$ can be expressed as follows:

$$p(f_{i,j}) \;=\; \sum_{m=1}^{M} p(f_{i,j}, v_m) \;=\; \sum_{m=1}^{M} p(v_m)\, p(f_{i,j} |\, v_m) \tag{1}$$

If we further adopt the assumption that each likelihood $p(f_{i,j} |\, v_m)$ is normally distributed, then we have

$$p(f_{i,j}) = \sum_{m=1}^{M} \pi_m \, \mathcal{N}(f_{i,j} |\, \mu_m, \Sigma_m), \tag{2}$$

where $\pi_m \equiv p(v_m)$, μ_m is a D-dimensional mean vector, and Σ_m is a covariance matrix of size $D \times D$. This is the familiar form of a Gaussian mixture model, for which the different terms π_m, μ_m,

and Σ_m can be estimated from training samples through clustering procedures. After training, this representation has the advantage that each μ_m can be interpreted as a reusable, frequently occurring component within $\mathbb{R}^D$ that is characteristic to particular *thing* classes in the training set.

This approach allows computation of soft assignments for each position (i, j) on the RoI lattice, and distributes responsibility of classification across the RoI. These assignments are then combined to arrive at the final class prediction for each RoI. Such an approach allows for accumulation of evidence from multiple components of an object, even when some of the components are absent because of occlusion.

As shown in the figure, a $k \times k$ attention mask is also generated and is used to scale the tensor of posterior values. This attention mechanism is implemented as a fully connected layer followed by a sigmoid activation function. The resulting object-presence scores (in the range $[0, 1]$) cause the system to concentrate on foreground objects within the RoI. In the final step of the detection pipeline, the system passes all elements of the attention-scaled tensor of posterior values to a series of dense layers to make final class predictions for the current RoI. Unlike Mask R-CNN, the bounding box offset prediction for the detect branch is separated from classification and is performed in parallel, using a couple of convolutional layers for channel adjustment followed by dense layers for prediction of final offsets of each RoI. The mask prediction branch of the instance head is identical to the one introduced by He et al. (2017) that generates class-agnostic binary masks for each RoI.

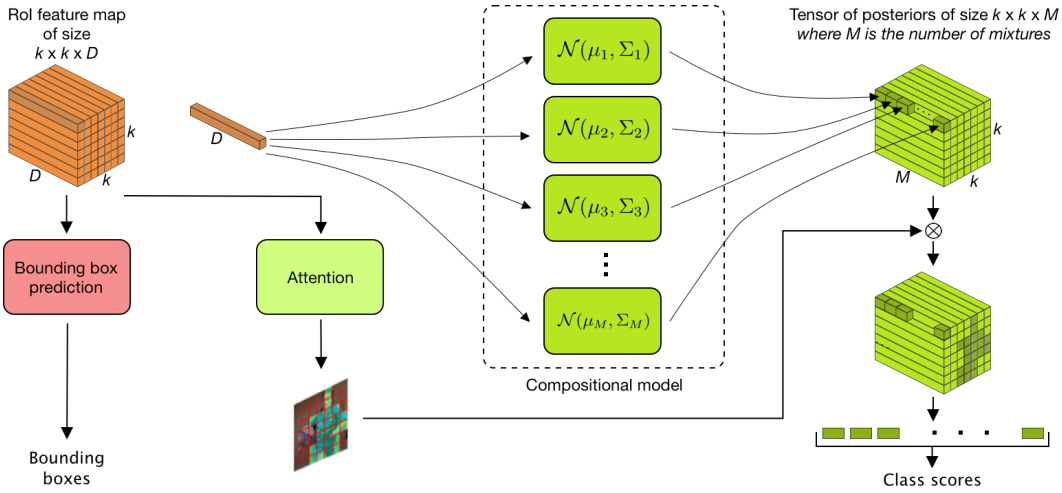

Figure 3: The new detect branch of the instance head. Feature vectors at each position in the $k \times k$ sized RoI lattice (shown in orange) are processed using a compositional model to compute a map of posterior values. The posterior values are then used to produce final class score predictions.

3.3 Final Prediction

The panoptic fusion head produces the final output of the system by aggregating the outputs of the semantic and instance heads. We use the panoptic head introduced in UPSNet (Xiong et al., 2019). It operates by creating a logit tensor of size $H \times W \times (N_{stuff} + N_{inst} + 1)$. The first N_{stuff} channels are taken from the logits produced by the semantic head, and N_{inst} channels are taken from the logits that were produced by the instance head. The final panoptic output is obtained by applying a softmax operation along the channel dimension. If the maximum across channels indicates one of the N_{stuff} channels, then the pixel belongs to a *stuff* class, otherwise it belongs to one of the instances of a *thing* class. The class assignment for each instance is calculated by the strategy explained in Xiong et al. (2019). To reduce the risk of making incorrect predictions, logits for an additional "unknown" category are also predicted.

3.4 Implementation Details

Model training. We implement our system in PyTorch (Paszke et al., 2019) and perform training with up to 16 GPUs. The novel instance head is trained with 4 loss terms for box classification, foreground

attention, box regression and mask segmentation. Because the architecture is based on (Xiong et al., 2019; He et al., 2017), we follow many of the settings used in those works. Unless otherwise specified, experimental results are obtained using the ResNet-50 FPN (R-50 FPN) backbone. In some cases, we also present results using the larger ResNet-101 FPN (R-101 FPN) backbone. Results for all datasets are reported on the validation split. We use the pretrained backbone and RPN weights from the baseline, and retrain all other task-specific heads. Since our instance head performs spatially aware object classification, we also need mask information for training. Therefore, we move away from the training strategy of Mask R-CNN and generate training data for object classification that includes mask information. All results are reported on setups with $M = 320$ mixtures ($4 \times$ number of classes in COCO, as used in Kortylewski et al. (2020a)) with a feature vector size $D = 256$ and lattice size $k = 7$ as used in He et al. (2017). Additional details related to the training process are provided in the appendix.

Learning compositional model parameters. We use the pretrained backbone network to collect feature vectors in $\mathbb{R}^M$, as needed to obtain parameters of our compositional model. For all foreground classes, ground-truth bounding boxes and masks are used to select feature vectors. First, we bring the ground-truth mask of each RoI to size $k \times k$ using bilinear interpolation. Then we collect feature vectors $f_{i,j}$ for all locations (i, j) having a mask value of 1 (after applying a threshold of 0.5). Similar to Girshick et al. (2015), we also add an extra "background" class to improve learning within the classification pipeline. A subset of these background RoIs are sampled from regions that correspond to *stuff* classes in the training images. Using these features, we apply the standard Expectation-Maximization algorithm to obtain $\{\pi_m, \mu_m, \Sigma_m\}$ for $m = 1, ..., M$. The parameters of the compositional model are learned before training the remaining portions of the task heads. More details are given in the appendix.

4 EXPERIMENTS

4.1 DATASETS AND EVALUATION METRICS

We perform most of our experiments on the Microsoft COCO dataset (Lin et al., 2014), which consists of 118k training images, 5k validation images and 20k images for testing. For panoptic segmentation, a total of 133 categories are specified, consisting of 80 *thing* categories and 53 *stuff* categories. We also show results on the Cityscapes dataset (Cordts et al., 2016), which contains pixel level annotations for a total of 19 classes of which 11 belong to *stuff* and 8 to *thing* categories. Images are divided into group sizes of 2975, 500, and 1525 for training, validation, and testing, respectively. The performance of our approach is evaluated using the Panoptic Quality (PQ) metric from Kirillov et al. (2019b):

$$PQ = \underbrace{\frac{\sum_{p,g \in TP} IoU(p,g)}{|TP|}}_{\text{Segmentation Quality (SQ)}} \times \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\text{Recognition Quality (RQ)}} \tag{3}$$

The term *IoU*$(p,\ g)$ is the Intersection over Union of a predicted segment p and a ground-truth segment g; *TP* is the number of True Positive segments (i.e., *IoU*$(p,\ g) > 0.5$); *FP* is the number of False Positives; and *FN* is number of False Negatives. We see that SQ is the average of Intersection over Union for all *TP* segments. We also compute PQ^{th} (PQ over all *thing* classes) and PQ^{st} (PQ over all *stuff* classes) to gain insights into instance and semantic segmentation results, respectively. We do not include the "unknown" category (section 3.3) when calculating Panoptic Quality.

4.2 QUALITATIVE COMPARISON

Some examples of qualitative results from our approach are shown in Figure 4. The first and second columns show the input data and the ground truth, respectively. The results of the baseline (Xiong et al., 2019) have also been included for comparison. These baseline results have been generated using a retrained model which performs slightly better than reported in Xiong et al. (2019).

All examples shown in Figure 4 contain some instances that appear in close proximity to others and suffer from varying amounts of occlusion. As seen in the figure, our approach makes an

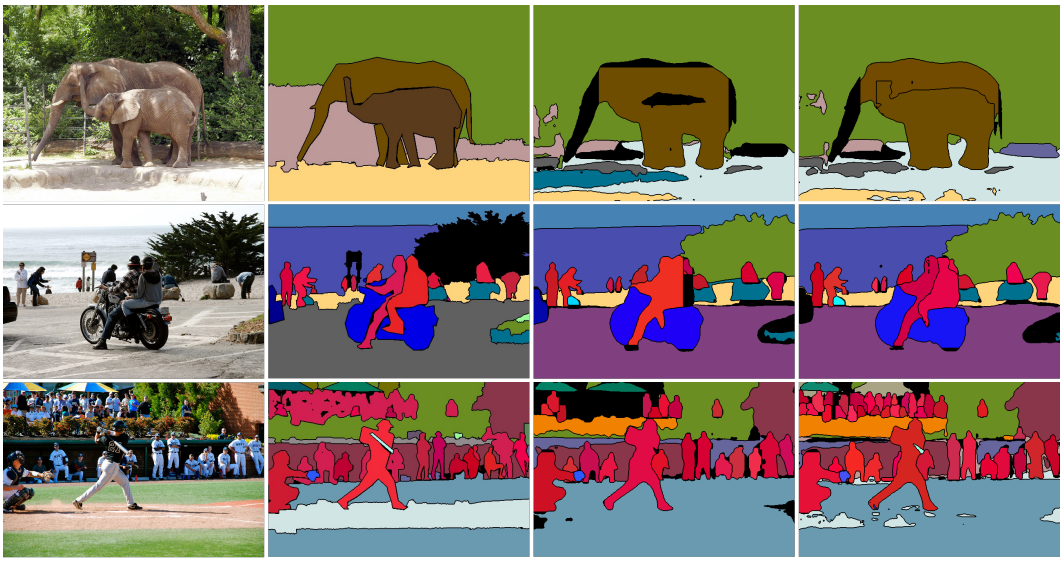

Figure 4: Qualitative results on examples from the COCO *Val* dataset.

Table 1: Panoptic segmentation results on the COCO 2018 *Val* dataset. Superscripts 'Th' and 'St' denote numbers for *thing* and *stuff* classes respectively. (*: computed using noisy annotations.)

Method	Backbone	PQ	SQ	RQ	PQ^{Th}	SQ^{Th}	RQ^{Th}	PQ^{St}
PCV (Wang et al., 2020)	R-50 FPN	37.5	77.7	47.2	40.0	78.4	50.0	33.7
AUNet (Li et al., 2019)	R-50 FPN	38.6	76.4	47.5	46.2	80.2	56.2	27.1
OANet (Liu et al., 2019)	R-50 FPN	39.0	77.1	47.8	48.3	81.4	58.0	24.9
UPSNet (Xiong et al., 2019)	R-50 FPN	42.5	78.5	52.5	48.1	79.2	59.2	33.9
*Ours**	R-50 FPN	40.3	78.2	50.0	44.6	78.8	55.0	33.9
UPSNet (Xiong et al., 2019)	R-101 FPN	46.7	80.5	56.9	53.2	81.2	64.6	36.9
*Ours**	R-101 FPN	44.9	80.5	54.7	50.3	81.6	60.9	36.8

improvement over the baseline and is able to detect most of the occluded instances. For example, the mother occluded by a baby elephant in the first row is a difficult case as the two instances are very similar and without clearly discernible edges, and as a result, is not handled well by the baseline, but they are distinguished as separate instances by our approach. In the second row, two people on a motorcycle have been reliably detected by our approach, but were merged by the baseline. In the third row, our system has detected a person who is occluded by the batter, just to the left. Also in the third row, our system has correctly detected many individuals in the stands, but they were not detected by the baseline. We note that there remains room for improvement in the masks produced by our system, but we have observed many cases for which the baseline system incorrectly merges two neighboring instances while our system distinguishes them correctly.

4.3 Quantitative Comparison

We show quantitative results on the validation split of both COCO (Table 1) and Cityscapes datasets (Appendix A). For most methods, we include architectures that share the same backbone as our method for a fair comparison. Despite dataset limitations (Section 4.4), our approach still performs well in comparison to many state-of-the-art approaches to panoptic segmentation and ranks second in terms of overall metrics as shown in Table 1. It is interesting to note that all methods except one in the table use the same base design of Mask R-CNN. The performance scores using the ResNet-101 backbone are also shown in the table. Making use of the larger ResNet-101 backbone improves overall PQ performance by 4.8 points and improves RQ by 6.3 points, which are both better than

gains made by the baseline. By introducing the compositional model in the instance head, our approach improves PQ^{Th} by 2.6 points and RQ^{Th} by 2.6 points, as reported in Table 2a.

To exclude the incorrect false positive predictions, we also report detection performance using *TP* counts, *FN* counts, and recall. In terms of detecting ground truth instances, our approach shows an improvement over the baseline. We note that an increase in recall could possibly result in lower precision. However, our method performs exceedingly well on crowded scenes; a large portion of *FP*s are contributed by such segments, which, given perfect annotations, would turn into TPs and improve both precision and recall even further. Table 2b shows a comparison of our approach with the baseline. We see an improvement of 1.45 points on the overall recall score, accompanied by a 2.53% increase in *TP* counts and a 3.36% decrease in *FN* counts. Using the larger ResNet-101 backbone increases the overall recall improvement even further to 2.14 points.

4.4 Panoptic Quality Computation

As the panoptic segmentation task is still fairly recent, the datasets and metrics continue to evolve. A particular problem for the work reported here is the noise level for small and partially occluded objects within the COCO dataset. Here we discuss some of these issues and propose alternatives to aid in minimizing the discrepancy between qualitative performance and quantitative metrics for panoptic segmentation. Similar discussions and suggestions can also be found in (FiftyOne, 2020; Porzi et al., 2019).

Computing panoptic quality can be defined as a two-step procedure: First, segment matching of detections is performed. Each ground truth instance is matched with a prediction if the IoU of the two is greater than 0.5. After matching, each prediction falls into one of three groups: *TP* (matched pairs), *FP* (unmatched predictions), and *FN* (unmatched ground truth annotations). In the second step, these groups are then used to compute the final PQ using (3). If an image in the COCO dataset contains more than 10 instances of the same category, often they have been labelled collectively as an "iscrowd" category. Each annotation has a parameter that indicates if the annotation is an "iscrowd". During usual quantitative metric calculation, the predictions made over regions that belong to "iscrowd" are ignored. However, the "iscrowd" flag has been left unset for many annotations. This causes successful detection of instances present in complex scenes to be evaluated as FPs (despite being TPs), contributing negatively to the quantitative performance. Moreover, it was found that many smaller sized instances in the COCO dataset have not been annotated at all, which caused smaller sized predictions to be considered as false positives. A detailed analysis is provided in the appendix.

To accommodate noisy labelling in the COCO dataset, we propose an alteration to the segment-matching algorithm which reduces the negative effect of such incorrect false positives. First, to account for the unset "iscrowd" flag, we compute the overlap area of a prediction with the corresponding ground-truth segment. If the prediction has a significant overlap (> 0.5) with a ground-truth segment of the same category, we consider it as a True Positive (represented in Table

Table 2: Results on COCO *Val*. (B = Baseline (Xiong et al., 2019); *O* = Ours.)

(a) Results for top 20 *thing* classes (by instance count) using ResNet-101 FPN backbone and updated segment matching. (WS: without smaller FPs; MI: detections on unset "iscrowd" instances; CR: detections on "iscrowd" instances.)

Method	PQ	RQ	PQ^{Th}	RQ^{Th}
B WS	41.4	51.1	53.3	66.5
O WS	**41.6**	**51.5**	**54.5**	**68.0**
B WS+MI	45.4	55.2	67.7	81.4
O WS+MI	**45.9**	**55.9**	**70.0**	**83.9**
B WS+MI+CR	45.5	55.4	68.4	82.0
O WS+MI+CR	**46.1**	**56.1**	**71.0**	**84.6**

(b) Comparison of instance detection performance for our approach against the baseline. (R-50 and R-101 are ResNet-50 and ResNet-101 with FPN backbones, respectively.) (↑ = Higher is better; ↓ = Lower is better.)

Method	*TP* ↑	*FN* ↓	Recall ↑
B (R-50)	20,599	15,504	0.5705
O (R-50)	**21,121**	**14,982**	**0.5850**
Increase			+0.0145
B (R-101)	22,535	13,568	0.6241
O (R-101)	**23,308**	**12,795**	**0.6455**
Increase			+0.0214

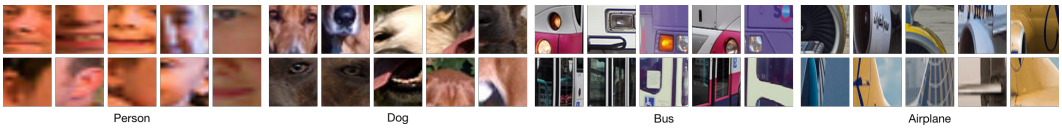

Figure 5: Patterns that map to the same component while detecting instances of four categories are shown. As seen in the figure, similar regions across instances map to the same component of the compositional model.

Figure 6: Parts that map to two components that are active for the *Vehicle* supercategory. The first five columns show parts of wheels taken from instances that were classified as *Car*, *Motorcycle*, *Bus* and *Truck*, with all parts having the same most active component. Similarly, the last five columns show parts of *Bicycle* and *Motorcycle* instances.

2a by MI, for Missing Iscrowd). The segments predicted over "iscrowd" having the same class are also deemed True Positives to gauge the detection performance over the complex "iscrowd" region (represented by CR in Table 2a). Secondly, false positives that have a small area (less than 64×64) are also not considered while calculating Panoptic Quality as they might lack the annotation in ground truth (represented in Table 2a by WS, for Without Smaller FPs).

4.5 Analysis of Compositional Model

This section discusses intermediate results for the compositional model, as generated during the inference phase. For each RoI, we retain the posterior values generated by the compositional model and consider the *argmax* at each position on the 2D lattice of the feature map after applying a threshold of 0.8. Then for each input image, RoIs predicted by the model are extracted and divided into $k \times k$ parts. Figure 5 shows parts of the RoIs where the same component is the most active for a particular category.

For each category, the COCO dataset also specifies a "supercategory" that indicates the broad group to which a category belongs. For instance, *thing* classes such as *Car*, *Bicycle*, *Motorcycle*, *Truck* and *Bus* altogether form the *Vehicle* supercategory. Interestingly, classes in some of the supercategories also share some common features or patterns that are unique to the supercategory. Figure 6 shows parts that map to two components that are active while detecting classes of the *Vehicle* supercategory. This shows that the compositional model is able to learn feature vectors that encode the common patterns observed on instances of the classes present in the dataset.

5 Conclusion

We have introduced a novel object classification approach based on compositional modelling that has proven to be effective at classifying separate instances of foreground objects. We demonstrated the efficacy of the approach by replacing the object detection pipeline in UPSNet with a compositional element that utilizes a mixture of distributions to model parts of objects. We presented extensive experimental results for the MS COCO dataset, and showed significant gains in performance in detecting foreground (*thing*) classes. Finally, we presented qualitative results to demonstrate that improved metrics and datasets are needed for proper characterization of panoptic segmentation systems.

REFERENCES

Anurag Arnab and Philip Torr. Pixelwise instance segmentation with a dynamically instantiated network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 441–450, 2017.

Harkirat Singh Behl, Mohammad Naja, Anurag Arnab, and Philip Torr. Meta-learning deep visual words for fast video object segmentation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8484–8491, 2020.

Elie Bienenstock, Stuart Geman, and Daniel Potter. Compositionality, MDL priors, and object recognition. In *Advances in Neural Information Processing Systems*, pp. 838–844, 1997.

Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. YOLACT: Real-time instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9157–9166, 2019.

Qiang Chen, Anda Cheng, Xiangyu He, Peisong Wang, and Jian Cheng. Spatialflow: Bridging all tasks for panoptic segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

Yi-Ting Chen, Xiaokai Liu, and Ming-Hsuan Yang. Multi-instance object segmentation with occlusion handling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3470–3478, 2015.

Bowen Cheng, Maxwell D. Collins, Yukun Zhu, Ting Liu, Thomas S. Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, 2016.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 764–773, 2017.

Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1601–1610, 2021.

Daan de Geus, Panagiotis Meletis, and Gijs Dubbelman. Panoptic segmentation with a joint semantic and instance segmentation network. *arXiv preprint arXiv:1809.02110*, 2018.

Daan de Geus, Panagiotis Meletis, and Gijs Dubbelman. Fast panoptic segmentation network. *IEEE Robotics and Automation Letters*, 5(2):1742–1749, 2020.

A. E. Elgammal and Larry S. Davis. Probabilistic framework for segmenting people under occlusion. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, volume 2, pp. 145–152, 2001.

Markus Enzweiler, Angela Eigenstetter, Bernt Schiele, and Dariu M. Gavrila. Multi-cue pedestrian classification with partial occlusion handling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 990–997. IEEE, 2010.

Voxel FiftyOne. Detection on COCO. `https://voxel51.com/docs/fiftyone/tutorials/evaluate_detections.html`, 2020.

Jerry A. Fodor and Zenon W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71, 1988.

Naiyu Gao, Yanhu Shan, Xin Zhao, and Kaiqi Huang. Learning category- and instance-aware pixel embedding for fast panoptic segmentation. *IEEE Transactions on Image Processing*, 30:6013–6023, 2021. doi: 10.1109/TIP.2021.3090522.

Stuart Geman, Daniel F. Potter, and Zhiyi Chi. Composition systems. *Quarterly of Applied Mathematics*, 60(4):707–736, 2002.

Dileep George, Wolfgang Lehrach, Ken Kansky, Miguel Lázaro-Gredilla, Christopher Laan, Bhaskara Marthi, Xinghua Lou, Zhaoshi Meng, Yi Liu, Huayan Wang, A. Lavin, and D. S. Phoenix. A generative vision model that trains with high data efficiency and breaks text-based CAPTCHAs. *Science*, 358(6368), 2017.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):142–158, 2015.

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*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969, 2017.

Derek Hoiem, Andrew N. Stein, Alexei A. Efros, and Martial Hebert. Recovering occlusion boundaries from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1–8, 2007.

Weixiang Hong, Qingpei Guo, Wei Zhang, Jingdong Chen, and Wei Chu. LPSNet: A lightweight solution for fast panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16746–16754, June 2021.

Rui Hou, Jie Li, Arjun Bhargava, Allan Raventos, Vitor Guizilini, Chao Fang, Jerome Lynch, and Adrien Gaidon. Real-time panoptic segmentation from dense detections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8523–8532, 2020.

Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6399–6408, 2019a.

Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9404–9413, 2019b.

Dieter Koller, Joseph Weber, and Jitendra Malik. Robust multiple car tracking with occlusion reasoning. In *Proceedings of the European Conference on Computer Vision*, pp. 189–196. Springer, 1994.

Adam Kortylewski. *Model-based image analysis for forensic shoe print recognition*. PhD thesis, University of Basel, 2017.

Adam Kortylewski, Ju He, Qing Liu, and Alan L. Yuille. Compositional convolutional neural networks: A deep architecture with innate robustness to partial occlusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8940–8949, 2020a.

Adam Kortylewski, Qing Liu, Huiyu Wang, Zhishuai Zhang, and Alan Yuille. Combining compositional models and deep networks for robust object classification under occlusion. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pp. 1333–1341, 2020b.

Justin Lazarow, Kwonjoon Lee, Kunyu Shi, and Zhuowen Tu. Learning instance occlusion for panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10720–10729, 2020.

Jie Li, Allan Raventos, Arjun Bhargava, Takaaki Tagawa, and Adrien Gaidon. Learning to fuse things and stuff. *arXiv preprint arXiv:1812.01192*, 2018a.

Qizhu Li, Anurag Arnab, and Philip Torr. Weakly-and semi-supervised panoptic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 102–118, 2018b.

Yanwei Li, Xinze Chen, Zheng Zhu, Lingxi Xie, Guan Huang, Dalong Du, and Xingang Wang. Attention-guided unified network for panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7026–7035, 2019.

Yanwei Li, Hengshuang Zhao, Xiaojuan Qi, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. Fully convolutional networks for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 214–223, 2021.

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 *Proceedings of the European Conference on Computer Vision*, pp. 740–755. Springer, 2014.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, 2017.

Huanyu Liu, Chao Peng, Changqian Yu, Jingbo Wang, Xu Liu, Gang Yu, and Wei Jiang. An end-to-end network for panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6172–6181, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Andra Petrovai and Sergiu Nedevschi. Multi-task network for panoptic segmentation in automated driving. In *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 2394–2401, 2019.

Lorenzo Porzi, Samuel Rota Bulo, Aleksander Colovic, and Peter Kontschieder. Seamless scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8277–8286, 2019.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

Konstantin Sofiiuk, Olga Barinova, and Anton Konushin. AdaptIS: Adaptive instance selection network. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 7355–7363, 2019.

Jian Sun, Yin Li, Sing Bing Kang, and Heung-Yeung Shum. Symmetric stereo matching for occlusion handling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pp. 399–406, 2005.

Joseph Tighe, Marc Niethammer, and Svetlana Lazebnik. Scene parsing with object instances and occlusion ordering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3748–3755, 2014.

Haochen Wang, Ruotian Luo, Michael Maire, and Greg Shakhnarovich. Pixel consensus voting for panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9464–9473, 2020.

Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. MaX-DeepLab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5463–5474, 2021.

Jianyu Wang, Zhishuai Zhang, Cihang Xie, Jun Zhu, Lingxi Xie, and Alan Yuille Yuille. Detecting semantic parts on partially occluded objects. In *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 73.1–73.13, September 2017.

Xiaoyu Wang, Tony X. Han, and Shuicheng Yan. An HOG-LBP human detector with partial occlusion handling. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 32–39, 2009.

Xinlong Wang, Tete Xiao, Yuning Jiang, Shuai Shao, Jian Sun, and Chunhua Shen. Repulsion loss: Detecting pedestrians in a crowd. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7774–7783, 2018.

Mark Weber, Jonathon Luiten, and Bastian Leibe. Single-shot panoptic segmentation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8476–8483. IEEE, 2020.

Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. UPSNet: A unified panoptic segmentation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8818–8826, 2019.

Tien-Ju Yang, Maxwell D. Collins, Yukun Zhu, Jyh-Jing Hwang, Ting Liu, Xiao Zhang, Vivienne Sze, George Papandreou, and Liang-Chieh Chen. DeeperLab: Single-shot image parser. *arXiv preprint arXiv:1902.05093*, 2019.

Yibo Yang, Hongyang Li, Xia Li, Qijie Zhao, Jianlong Wu, and Zhouchen Lin. SOGNet: Scene overlap graph network for panoptic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 12637–12644, 2020.

Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised scene de-occlusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3784–3792, 2020.

Zhishuai Zhang, Cihang Xie, Jianyu Wang, Lingxi Xie, and Alan L. Yuille. Deepvoting: A robust and explainable deep network for semantic part detection under partial occlusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1372–1380, 2018.

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*, pp. 2881–2890, 2017.

Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1464–1472, 2017.

A Additional Results

A.1 Comparison with State-of-the-art Methods

We show additional results using both ResNet-50 FPN and ResNet-101 FPN backbones. First, qualitative results are shown in Figures 7, 8, and 9 for the COCO dataset, and Figure 10 shows additional results on the Cityscapes dataset. As seen in the figures, our approach makes significant improvements at detecting occluded instances, as compared to the baseline.

A quantitative comparison with several state-of-the-art panoptic segmentation methods on the COCO and Cityscapes datasets is shown in Tables 3 and 4, respectively. We emphasize that many of the qualitative improvements from Figures 1-4 are *not* reflected in the quantitative results. The reason for the discrepancy is discussed in detail in Section 4.3. Table 3 reports the overall PQ score and PQ averaged over *thing* classes and *stuff* classes as well. The methods are grouped into two broad categories called Single stage and Two stage. It is interesting to note that six of the eleven two stage methods in the table use the same base design of Mask R-CNN He et al. (2017). For most methods, we include architecture versions that share the same backbone as our method for

uniform comparison. The performance of our approach is comparable to UPSNet as we use many modules from the UPSNet architecture. Results using the ResNet-101 backbone are also shown in the table. Making use of the larger ResNet-101 backbone improves overall performance 4.8 points and improves RQ by 6.3 points, which are both better than improvements made by the baseline. The results on the Cityscapes dataset are shown in Table 4. In contrast to COCO, the Cityscapes dataset is much smaller in terms of both, dataset size and category count. The dataset also contains a fairly large ratio of crowds, with very little visibility of occluded instances. This makes it tougher to detect instances well. As seen in the table, the performance of the proposed approach is almost identical to the baseline.

A.2 Ablation Experiments on Compositional Model

Number of clusters. The compositional model is implemented here as a mixture of multivariate Gaussian distributions. Determining the optimum number of components to model an entire class in the dataset is not trivial. Table 5 shows results of experiments on the COCO *Val* dataset with varying sizes of the compositional model. Clusters represent the count of mixture components. As seen in the table, the RQ and Recall metrics improve as number of clusters are increased. To maintain speed of training, we show all results using a compositional model of 320 components ($C \times 4$ where C is the number of classes). However, performance improvements can be expected if size of the compositional model is increased.

Ground truth data. The ground truth data consists of all boxes and masks of ground truth instances. The ratio of *foreground* to *background* instances for training is set to 1:4, as inferred from the training data used for Faster R-CNN Ren et al. (2015) (as both methods use RPN to generate proposals). It is important to learn this separation function well, as assigning high confidence to RoIs with low overlap is not desirable. To include a high number of *background* proposals, multiple folds of the *foreground* samples are included each batch of the training data. This strategy results in a larger count of *background* RoIs as $N \times 4$ *background* proposals are sampled (N is the count of *foreground* proposals). Table 6 shows the results of including multiple folds of *foreground* samples in the training data. As seen in the table, showing more *background* RoIs to the detect branch during training improves performance of the detection branch.

Compositional model training. While learning parameters of the compositional model, feature vectors used for training are sampled using the procedure explained in Appendix C.

Following Girshick et al. (2015), we also predict an extra *background* class (along with *thing* classes) to recognize RoIs that have low overlap with a foreground instance. Therefore, features that map to *background* regions are also included in the training data to maintain some components that can recognize *background* regions in the RoI. We explored this design choice and performed A/B testing to determine if inclusion of *background* features improves recognition performance. Inclusion of *background* features helps the compositional model determine if some position on the *RoI lattice* corresponds to the background. The results of A/B testing that justify the inclusion of *background* features are shown in Table 7. As seen from the metrics, including *background* features improves detection performance.

A.3 Classwise Performance

Here we discuss the class-wise performance of our approach on the COCO dataset. For the panoptic segmentation task in COCO, there are a total of 133 classes, consisting of 80 *thing* or object categories and 53 *stuff* categories.

Table 8 shows the *TP* and *FN* counts and recall for some classes in the COCO dataset. The *Person* and *Car* classes have the highest frequency in the dataset and have an improved recall of 2.59 and 2.99 percentage points respectively. Smaller sized instances (by area) of categories such as *Traffic Light*, *Bird*, *Book* and fruits see the largest gains in recall, which indicates that our approach is able to detect instances with varied scales and is able to discern instances even though they are closely packed together (and often partially occluded, as well). A situation where our approach struggles is when objects with very similar textures are closely packed together. For example, in crowded scenes in which animals of the same type (elephant, zebra, sheep) overlap one another, the boundaries of

instances often become ambiguous. In some cases, our model and the baseline may incorrectly detect them as a single instance. However, we have found several such cases for which our method separates individual instances well and generates better segmentation masks than the baseline.

The features used for learning the parameters of the compositional model are extracted using ground truth bounding boxes and masks. During our experiments, the input images were used without the application of any data augmentation techniques for feature extraction to maintain fast convergence times. Further improvements to the performance of the compositional model can be expected by including augmented data for learning the parameters of the compositional model.

B PANOPTIC QUALITY COMPUTATION

In this section, we discuss some of the issues with panoptic segmentation metrics and noisy labeling that create a discrepancy between qualitative performance and quantitative metrics for panoptic segmentation. Similar discussions and suggestions can also be found in FiftyOne (2020); Porzi et al. (2019).

As discussed in the Experiments section of the main document, computing panoptic quality can be defined as a two step procedure: First, segment matching of detections is performed. Each ground truth instance is matched with a prediction if the IoU of the two is greater than 0.5. After matching, each prediction falls into one of three groups: *TP* (matched pairs), *FP* (unmatched predictions), and *FN* (unmatched ground truth annotations). In the second step, these groups are then used to compute the final PQ. If an image in the COCO dataset contains more than 10 instances of the same category, often the ground-truth annotations group those instances collectively as an "iscrowd" category. Each annotation has a parameter that indicates if the annotation is an "iscrowd". During usual quantitative metric calculation, the predictions made over regions that belong to "iscrowd" are ignored. However, the "iscrowd" flag has been left unset for many annotations. This omission within the ground-truth data causes successful detection of instances present in complex scenes to be evaluated as FPs (despite being TPs), contributing negatively to the quantitative performance. Moreover, it was found that many smaller sized instances in the COCO dataset have not been annotated at all, which caused smaller sized predictions to be considered as false positives.

Some examples of incorrect labelling issues are shown in Figure 11. We show the input image, ground truth and predicted results with green boxes indicating all of the cases that were tabulated as false positives. In the first row, we see that a majority of the "false positives" are of the chair category; these should have been counted as true positives, but only a few of the corresponding chairs are labelled in the ground truth data. Our approach is also able to detect many correct instances on the "iscrowd" region. The second row shows an example where segments are considered as false positives due to the "iscrowd" flag being set to 0 for instances of the book category. Notice that the entire set of books to the left is labelled as a single instance (with the "iscrowd" flag set to 0), which is inconsistent with some books being labelled individually on the right. Finally, the last row shows correct detections of many instances of the person category that are very small in size, but the ground truth annotations are missing and therefore our correct detections are included in the total of false positives.

To accommodate this sort of noisy labelling in the COCO dataset, we propose a modification to the segment matching algorithm which reduces the negative effect of such incorrect false positives. First, to account for the unset "iscrowd" flag, we compute the overlap area of a prediction with the corresponding ground truth segment. If the prediction has a significant overlap (> 0.5) with a ground truth segment of the same category, we consider it as a True Positive. The segments predicted over "iscrowd" having the same class are also deemed True Positives to gauge the detection performance over the complex "iscrowd" region. Secondly, false positives that have a small area (less than 64×64) are also not considered while calculating Panoptic Quality as they might lack the annotation in ground truth. Table 1 in the main document shows the performance using the modified segment matching algorithm. All other tables show metrics calculated using the original algorithm.

C IMPLEMENTATION DETAILS

In this section, we discuss the training process and design choices in detail, and explain how the output of the instance head is generated in the inference phase. We also expand on some of the points discussed in the main paper.

Model training. Our panoptic segmentation architecture is trained with a total of 7 loss terms: panoptic-head loss (pixelwise cross-entropy loss for the unified panoptic output), semantic-head loss (pixelwise cross-entropy loss and RoI loss), and instance-head loss (4 loss terms for box classification, foreground attention, box regression and mask segmentation). Each loss has a weighting factor associated with it to maintain balance. We use standard stochastic gradient descent with momentum with a weight decay of 0.0001. All the input images are resized to dimensions where the shorter side is 800 and the maximum possible largest side is 133. All images undergo horizontal flipping and per-channel normalization.

In the inference phase, the first step of the instance head is to get the class scores and box offsets from the detection branch. The RoIs with class scores and box offset predictions are then filtered to only retain RoIs that have confidence scores greater than 0.6, followed by non-maximum suppression to reject duplicate predictions. The RoIs that remain are then fed to the mask segmentation branch that generates binary masks for each RoI and subject attention predictions. To combine mask predictions, a pruning process is used. First, RoIs (now with masks added) are sorted in the decreasing order of confidence. Each mask is then interpolated to the image scale and place onto an empty canvas. In the final output, there will be one canvas for each *foreground* class that has the same spatial size of the input image. If any mask happens to have an overlap greater than 0.3 with another that was placed earlier, the mask being processed is discarded. Otherwise, the non-overlapping portion of the mask is copied to the canvas. In this way, logits for each *foreground* category are calculated and passed to the final fusion head for final panoptic logits prediction.

Since our instance head performs spatially aware object classification, we also need mask information for training. Therefore, we move away from the training strategy of Mask R-CNN and generate training data for object classification that includes mask information. The RoIs used for training still consist of a mixture of foreground and background samples. For each image, we use the ground-truth RoIs and their corresponding masks as the foreground samples. The ratio of foreground to background RoIs is set to 1:4. The background RoIs are chosen from a pool of RoIs that have an IoU between 0 and 0.5 with any ground-truth box. After combining the two groups of samples, each batch then is trimmed to limit the RoI batch size to 512. Features extracted using the pretrained weights of the backbone form the training data for learning the parameters of the compositional model.

Learning compositional model parameters. To perform classification in the instance head, we use a compositional model that matches features at every spatial position (i, j) on the 2D lattice of the RoI feature map with a reference set. This reference set of features represent frequently observed sub-parts of instances of some *thing* class. Our method aims to learn representations of higher level features or object parts rather than per pixel representations learnt by Behl et al. (2020). We assume that patterns / parts may be shared across classes, rather than assigning a constant number of centroids per class (Behl et al., 2020). Similarity functions also differ, Behl et al. (2020) uses cosine similarity with softmax vs. posteriors w.r.t. each component in a multivariate GMM. There are also some works that attempt to model instances in a part-group fashion. Bolya et al. (2019) produces "image sized" prototype masks combined with mask co-efficients for mask assembly as opposed to compositionality for object classification, while Arnab & Torr (2017) uses a disjointly trained pre-existing detector. Shape priors are used by an instance CRF to generate masks. In contrast to this, our method uses sub-parts of objects to make class predictions, with no assumptions about shape information.

Our compositional model consists of M multivariate Gaussians that correspond to these commonly observed patterns. To learn the parameters of these Gaussians, we use the pre-trained backbone network to collect all features using the ground truth information of all *thing* classes. We leverage the available mask information to identify which locations on the 2D lattice of the RoI feature map correspond to instances of *thing* classes. We use the feature maps extracted using RoIAlign to learn parameters of the compositional model.

First, we bring the ground-truth mask of each RoI to size $k \times k$ using bilinear interpolation. Then we collect feature vectors $f_{i,j}$ for all locations (i, j) having a mask value of 1 (after applying a threshold of 0.5). Similar to Girshick et al. (2015), we also add an extra "background" class to improve learning within the classification pipeline. A subset of these background RoIs are sampled from regions that correspond to *stuff* classes in the training images. Using these features, we apply the standard Expectation-Maximization algorithm to obtain $\{\pi_m, \mu_m, \Sigma_m\}$ for $m = 1, ..., M$. The parameters of the compositional model are learned before training the remaining portions of the task heads.

To obtain the parameters of our compositional model, with M multivariate Gaussians, we use the pretrained backbone network to collect all features using the ground truth boxes and corresponding masks of all instances of foreground classes. This is shown in Figure 12, which shows an example from the COCO dataset. In the left half of the figure, we show the input image with each RoI annotated in green. On the right, we show the extracted RoIs and place corresponding masks on top of them. The highlighted parts in the rightmost column correspond to $m_{i,j} = 1$. Similar to Ren et al. (2015), we also add an extra *background* class to improve learning of the classification pipeline. A subset of these background RoIs are sampled from regions that correspond to *stuff* in the input image.

Figure 7: Additional results using the ResNet-50 FPN backbone on the COCO *Val* dataset. Left to right, the columns show input images, ground-truth annotations, baseline results using UPSNet, and our results using a compositional model. In the top row, notice that our system has correctly detected a person who is severely occluded by the batter, at the left side of the batter, while also detecting a significantly higher number of spectator instances, even though some of those instances have not been annotated individually in the ground truth. In the second row, occluded *Person* instances present in the background have also been detected with high precision. In the remaining rows, our system has performed better than the baseline for many of the foreground objects.

Figure 8: Representative results of our approach using ResNet-101 on the COCO *Val* dataset. All of these cases represent complex scenes involving occlusion. Our system has performed better than the baseline in each case.

Figure 9: More representative results of our approach using the ResNet-101 backbone. Rows 1, 3 and 5 contain examples of crowded scenes where our approach shows an improvement over the baseline. It is interesting to note that our approach detects and segments many instances that are not present in the ground truth labels. The example in row 2 is particularly difficult, and yet it is segmented well by our approach despite having an unusual perspective with a partially visible *zebra* instance occluding another instance of the same class.

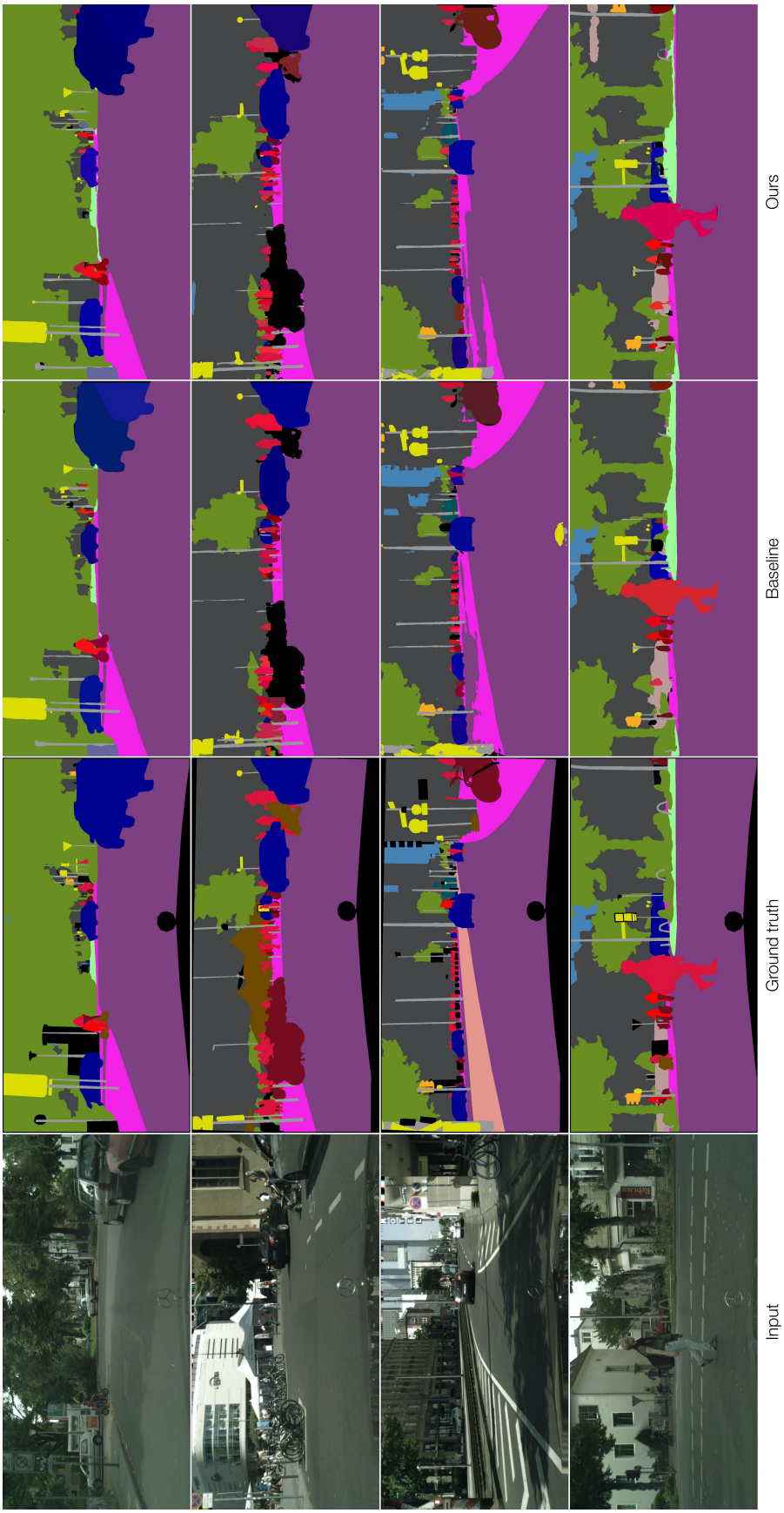

Figure 10: Results on the Cityscapes dataset using the ResNet-50 FPN backbone. As seen in the figure, our approach is able to recognize many occluded instances better than the baseline.

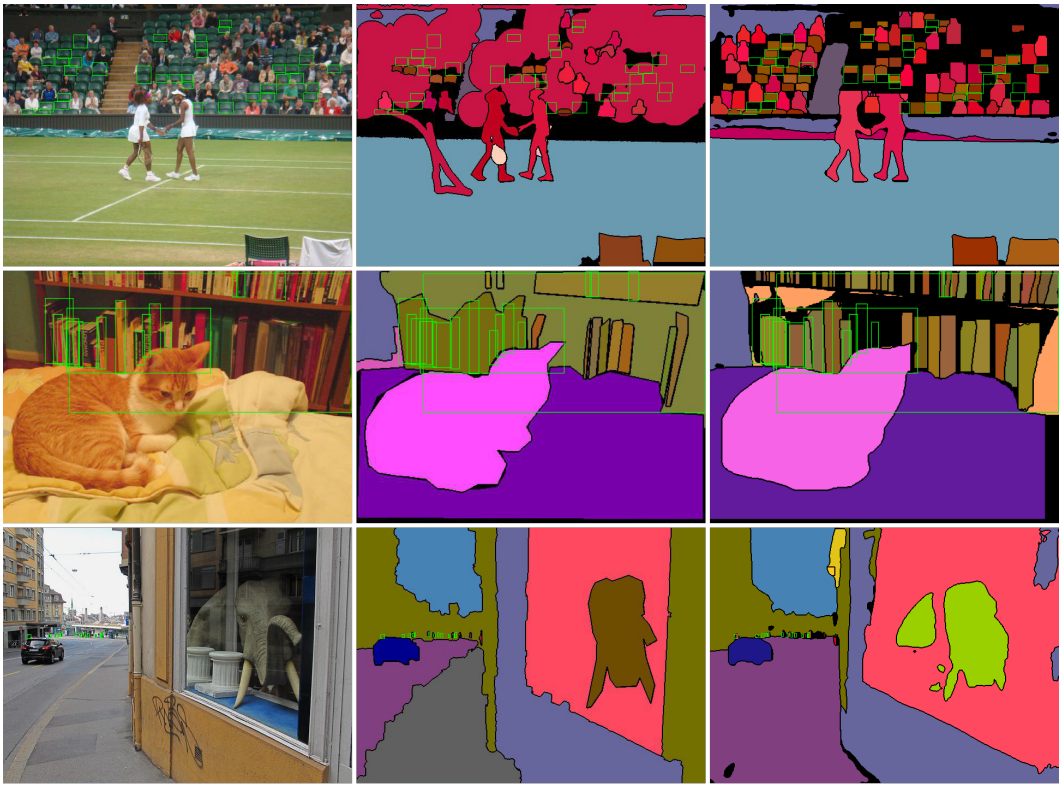

Figure 11: Examples from the COCO dataset that show points discussed in section B. The green boxes indicate instances that were detected by our system and were flagged as False Positives. However, close examination reveals that almost all of these detections were actually correct.

Figure 12: *(left)* Example image from the COCO dataset with ground-truth RoIs annotated in green. *(middle)* RoIs extracted from the image. *(right)* RoIs with ground-truth masks superimposed. Highlighted cells indicate parts of the instance.

Table 3: Panoptic segmentation results on the MS-COCO 2018 *Val* dataset. Superscripts 'Th' and 'St' denote numbers for *thing* and *stuff* classes respectively. (*: computed using noisy annotations.)

Method	Backbone	PQ	SQ	RQ	PQ^{Th}	SQ^{Th}	RQ^{Th}	PQ^{St}	SQ^{St}	RQ^{St}
Single Stage										
DeeperLab (Yang et al., 2019)	Xception-71	33.8	-	-	-	-	-	-	-	-
Hou et al.(Hou et al., 2020)	ResNet-50 FPN	37.1	-	-	41.0	-	-	31.3	-	-
PCV (Wang et al., 2020)	ResNet-50 FPN	37.5	77.7	47.2	40.0	78.4	50.0	33.7	76.5	42.9
Panoptic DeepLab (Cheng et al., 2020)	Xception-71	40.2	-	-	44.4	-	-	33.8	-	-
Two Stage										
JSIS-Net (de Geus et al., 2018)		26.9	72.4	35.7	29.3	72.1	39.2	23.3	73.0	30.4
AUNet (Li et al., 2019)	ResNet-50 FPN	38.6	76.4	47.5	46.2	80.2	56.2	27.1	70.8	34.5
AdaptIS (Sofiiuk et al., 2019)	ResNet-50 FPN	41.8	78.4	51.3	47.8	81.3	58.0	32.8	74.1	41.1
Panoptic FPN (Kirillov et al., 2019a)	ResNet-50 FPN	39.0	-	-	45.9	-	-	28.7	-	-
OANet (Liu et al., 2019)	ResNet-50 FPN	39.0	77.1	47.8	48.3	81.4	58.0	24.9	70.6	32.5
SOGNet (Yang et al., 2020)	ResNet-50 FPN	43.7	-	-	50.6	-	-	33.6	-	-
SpatialFlow (Chen et al., 2020)	ResNet-50 FPN	39.3	-	-	45.1	-	-	30.5	-	-
Single-Shot (Weber et al., 2020)	ResNet-50 FPN	32.4	-	-	34.8	-	-	28.6	-	-
Naiyu Gao et al. (Gao et al., 2021)	ResNet-50 FPN	40.2	-	-	45.3	-	-	32.3	-	-
OCFusion (Lazarow et al., 2020)	ResNet-50 FPN	42.5	-	-	49.1	-	-	32.5	-	-
UPSNet (Xiong et al., 2019)	ResNet-50 FPN	42.5	78.5	52.5	48.1	79.2	59.2	33.9	77.4	42.3
Ours*	ResNet-50 FPN	40.3	78.2	50.0	44.6	78.8	55.0	33.9	77.4	42.3
Method	Backbone	PQ	SQ	RQ	PQ^{Th}	SQ^{Th}	RQ^{Th}	PQ^{St}	SQ^{St}	RQ^{St}
UPSNet (Xiong et al., 2019)	ResNet-101 FPN	46.7	80.5	56.9	53.2	81.2	64.6	36.9	79.5	45.4
Ours*	ResNet-101 FPN	44.9	80.5	54.7	50.3	81.6	60.9	36.8	78.4	45.3

Table 4: Panoptic segmentation results on the Cityscapes *Val* dataset. Superscripts 'Th' and 'St' denote numbers for *thing* and *stuff* classes respectively.

Method	Backbone	PQ	PQ^{Th}	PQ^{St}
Single Stage				
Hou et al. (Hou et al., 2020)	ResNet-50 FPN	58.8	52.1	63.7
Two Stage				
Panoptic FPN (Kirillov et al., 2019a)	ResNet-101 FPN	58.1	52.0	62.5
SOGNet (Yang et al., 2020)	ResNet-50 FPN	60.0	56.7	62.5
SpatialFlow (Chen et al., 2020)	ResNet-101 FPN	59.6	55.0	63.1
TASCNet (Li et al., 2018a)	ResNet-50 FPN	59.3	56.3	61.5
Seamless (Porzi et al., 2019)		60.3	56.1	63.3
OCFusion (Lazarow et al., 2020)	ResNet-50 FPN	59.3	53.5	63.6
UPSNet (Xiong et al., 2019)	ResNet-50 FPN	58.7	53.2	62.6
Ours	ResNet-50 FPN	58.2	52.2	62.7

Table 5: Results of experiments with varying sizes of the compositional model. Clusters represent the count of mixture components. As seen in the table, the RQ and Recall metrics improve as number of clusters are increased. To maintain speed of training, we show all results using a compositional model of 320 components ($C \times 4$ where C is the number of classes). However, performance improvements can be expected if size of the compositional model is increased.

Clusters	PQ	PQ^{Th}	SQ^{Th}	RQ^{Th}	Recall
160	38.1	42.3	77.9	52.7	0.5289
320	38.2	42.2	77.7	52.7	0.5319
640	38.4	42.6	77.9	53.1	0.5334

Table 6: Results of experiments to compare the data augmentation to increase training data size per image for the detection branch of the instance head. Including a higher count of *background* RoIs, allows the network to learn the large variation of *background* RoIs with different IoU thresholds. As seen in the table, showing more of these RoIs improves performance of the detection branch.

Augmentation	PQ	PQ^{Th}	SQ^{Th}	RQ^{Th}	Recall
Without	39.9	44.1	78.6	54.6	0.5357
With	40.1	44.4	79.7	54.8	0.5456

Table 7: The compositional model is trained using features that are sampled from both *foreground* objects and *background* regions. Inclusion of *background* features helps the compositional model determine if some position on the *RoI lattice* corresponds to the background. The results of A/B testing that justify the inclusion of *background* features are shown in this table. As seen from the metrics, including *background* features improves detection performance.

Background	PQ	PQ^{Th}	SQ^{Th}	RQ^{Th}	Recall
Without	39.9	43.9	78.4	54.4	0.5440
With	40.1	44.4	79.7	54.8	0.5456

Table 8: A comparison of instance detection performance using the ResNet-50 backbone of our approach against the baseline (Xiong et al., 2019) for some classes in the MS-COCO 2018 *Val* dataset. The *Person* and *Car* classes have the highest frequency in the dataset and have an improved recall of 2.59 and 2.99 percentage points respectively. Smaller sized instances (by area) of categories such as *Traffic Light*, *Bird*, *Book* and fruits are also detected well.

Class	Baseline		Ours		Recall		*Increase in Recall*
	TP	FN	TP	FN	Baseline	Ours	
Person	7884	2891	8163	2612	0.7317	0.7576	0.0259
Bicycle	131	183	134	180	0.4172	0.4268	0.0096
Car	1172	733	1229	676	0.6152	0.6451	0.0299
Motorcycle	211	156	216	151	0.5749	0.5886	0.0136
Airplane	108	35	109	34	0.7552	0.7622	0.0070
Truck	176	213	180	209	0.4524	0.4627	0.0103
Boat	196	228	205	219	0.4623	0.4835	0.0212
Traffic Light	349	285	371	263	0.5505	0.5852	0.0347
Parking Meter	35	25	37	23	0.5833	0.6167	0.0333
Bench	126	279	128	277	0.3111	0.3160	0.0049
Bird	187	239	204	222	0.4390	0.4789	0.0399
Cow	247	121	253	115	0.6712	0.6875	0.0163
Bear	55	16	56	15	0.7746	0.7887	0.0141
Backpack	78	282	85	275	0.2167	0.2361	0.0194
Umbrella	237	170	250	157	0.5823	0.6143	0.0319
Handbag	119	412	128	403	0.2241	0.2411	0.0169
Tie	62	190	65	187	0.2460	0.2579	0.0119
Suitcase	150	143	151	142	0.5119	0.5154	0.0034
Baseball Glove	78	70	82	66	0.5270	0.5541	0.0270
Bottle	553	456	578	431	0.5481	0.5728	0.0248
Wine Glass	143	195	155	183	0.4231	0.4586	0.0355
Cup	462	400	502	360	0.5360	0.5824	0.0464
Banana	114	256	140	230	0.3081	0.3784	0.0703
Apple	69	165	78	156	0.2949	0.3333	0.0385
Carrot	134	230	154	210	0.3681	0.4231	0.0549
Cake	158	149	167	140	0.5147	0.5440	0.0293
Chair	670	1082	757	995	0.3824	0.4321	0.0497
Potted Plant	153	187	154	186	0.4500	0.4529	0.0029
Book	287	841	362	766	0.2544	0.3209	0.0665
Scissors	12	23	13	22	0.3429	0.3714	0.0286
Toothbrush	16	41	17	40	0.2807	0.2982	0.0175