Learning Instance Occlusion for Panoptic Segmentation

Justin Lazarow* Kwonjoon Lee* Kunyu Shi* Zhuowen Tu
University of California San Diego
{jlazarow, kwl042, kshi, ztu}@ucsd.edu

Abstract

Panoptic segmentation requires segments of both "things" (countable object instances) and "stuff" (uncountable and amorphous regions) within a single output. A common approach involves the fusion of instance segmentation (for "things") and semantic segmentation (for "stuff") into a non-overlapping placement of segments, and resolves overlaps. However, instance ordering with detection confidence do not correlate well with natural occlusion relationship. To resolve this issue, we propose a branch that is tasked with modeling how two instance masks should overlap one another as a binary relation. Our method, named OCFusion, is lightweight but particularly effective in the instance fusion process. OCFusion is trained with the ground truth relation derived automatically from the existing dataset annotations. We obtain state-of-the-art results on COCO and show competitive results on the Cityscapes panoptic segmentation benchmark.

1. Introduction

Image understanding has been a long standing problem in both human perception [1] and computer vision [25]. The *image parsing* framework [35] is concerned with the task of decomposing and segmenting an input image into constituents such as objects (text and faces) and generic regions through the integration of image segmentation, object detection, and object recognition. Scene parsing is similar in spirit and consists of both non-parametric [33] and parametric [40] approaches.

After the initial development, the problem of image understanding was studied separately as object detection (or extended to instance segmentation) and semantic segmentation. Instance segmentation [27, 28, 5, 20, 10, 29, 39, 15] requires the detection and segmentation of each *thing* (countable object instance) within an image, while semantic segmentation [30, 34, 9, 24, 2, 41, 40] provides a dense per-pixel classification without distinction between instances within the same *thing* category. Kirillov *et al*. [17] proposed the panoptic segmentation task that combines the strength of semantic segmentation and instance segmentation. In this task, *each pixel* in an image is assigned either to a background class (*stuff*) or to a specific foreground object (an *instance* of *things*).

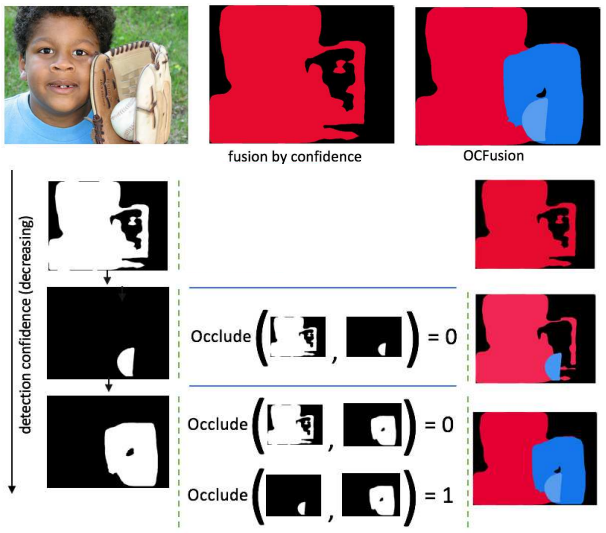


Figure 1: **An illustration of fusion using masks sorted by detection confidence alone [17] *vs*. with the ability to query for occlusions (OCFusion; ours).** Occlude$(A, B) = 0$ in occlusion head means mask B should be placed on top of mask A. Mask R-CNN proposes three instance masks listed with decreasing confidence. The heuristic of [17] occludes all subsequent instances after the "person", while our method retains them in the final output by querying the occlusion head.

A common approach for panoptic segmentation has emerged in a number of works [16, 19, 38] that relies on combining the strong baseline architectures used in semantic segmentation and instance segmentation into either a separate or shared architecture and then *fusing* the results from the semantic segmentation and instance segmentation branches into a single panoptic output. Since there is no expectation of consistency in proposals between semantic and instance segmentation branches, conflicts must be resolved. Furthermore, one must resolve conflicts *within* the instance segmentation branch as it proposes segmentations indepen-

* indicates equal contribution.

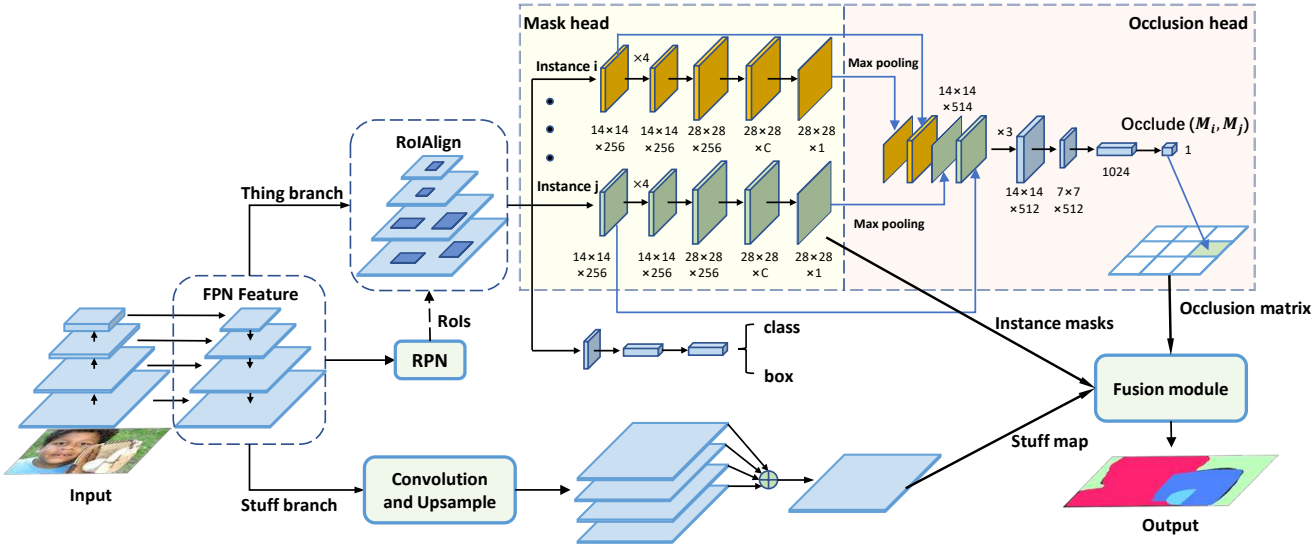


Figure 2: **Illustration of the overall architecture.** The FPN is used as a shared backbone for both thing and stuff branches. In thing branch, Mask R-CNN will generate instance mask proposals, and the occlusion head will output binary values $Occlude(M_i, M_j)$ (Equation 1) for each pair of mask proposals M_i and M_j with *appreciable* overlap (larger than a threshold) to indicate occlusion relation between them. Occlusion head architecture is described in Section 2.4. Fusion process is described in 2.3.

dent of each other. While a pixel in the panoptic output can only be assigned to a single class and instance, instance segmentation proposals are often overlapping.

To handle these issues, Kirillov *et al*. [17] proposed a fusion process similar to non-maximum suppression (NMS) that favors instance proposals over semantic proposals. However, we observe that occlusion relationships between different objects do not correlate well with object detection confidences used in this NMS-like fusion procedure [17], which therefore generally leads to poor performance when an instance that overlaps another (*e.g*., a tie on a shirt in Figure 3a) has lower detection confidence than the instance it should occlude. This can cause a large number of instances that Mask R-CNN *successfully* proposes fail to exist in the panoptic prediction (shown in Figure 1).

Therefore, in this work, we focus on enriching the fusion process established by [17] with a binary relationship between *instances* to determine occlusion ordering. We propose adding an additional branch (occlusion head) to the instance segmentation pipeline tasked with determining which of two instance masks should lie on top of (or below) the other to resolve occlusions in the fusion process. The proposed occlusion head can be fine-tuned easily on top of an existing Panoptic Feature Pyramid Networks (FPNs) [16] architecture with minimal difficulty. We call our approach fusion with occlusion head (OCFusion). OCFusion brings significant performance gains on the COCO and Cityscapes panoptic segmentation benchmarks with low computational cost.

2. Learning Instance Occlusion for Panoptic Fusion

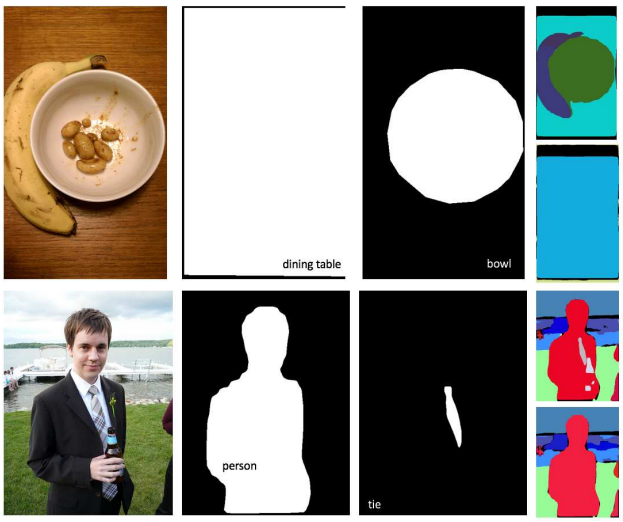
We adopt the coupled approach of [16] that uses a shared Feature Pyramid Network (FPN) [21] backbone with a top-down process for semantic segmentation branch and Mask R-CNN [10] for instance segmentation branch.

In this section, we first discuss the instance occlusion problem arising within the fusion heuristic introduced in [17] and then introduce OCFusion method to address the problem. The overall approach is shown in Figure 2.

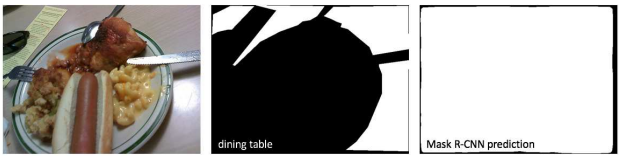
2.1. Fusion by confidence

The fusion protocol in [17] adopts a greedy strategy during inference in an iterative manner. Instance proposals are first sorted in order of decreasing detection confidence. In each iteration, the proposal is skipped if its intersection with the mask of all already assigned pixels is above a certain ratio of τ. Otherwise, pixels in this mask that have yet to be assigned are assigned to the instance in the output. After all instance proposals of some minimum detection threshold are considered, the semantic segmentation is merged into the output by considering its pixels corresponding to each "stuff" class. If the number of pixels exceeds some threshold after removing already assigned pixels, then these pixels are assigned to the corresponding "stuff" category. Pixels that are unassigned after this entire process are considered void predictions and have special treatment in the panoptic scoring process. We denote this type of fusion as *fusion by confidence*.

Softening the greed. The main weakness of the greedy fusion process is the complete reliance on detection confidences (*e.g.* for Mask R-CNN, those from the box classification score) for a tangential task. Detection scores not only have little to do with mask quality (*e.g.*, [13]), but they also do not incorporate any knowledge of *layout*. If they are used in such a way, higher detection scores would imply a more foreground ordering. Often this is detrimental since Mask R-CNN exhibits behavior that can assign near-maximum confidence to very large objects (*e.g.* see dining table images in Figure 3b) that are both of poor mask quality and not truly foreground. It is common to see images with a significant number of true instances suppressed in the panoptic output by a single instance with large area that was assigned the largest confidence.



(a)



(b)

Figure 3: **Images and ground truth masks from the COCO dataset. (a)** is an example where even predicting the ground truth mask creates ambiguity when attempting to assign pixels to instances in a greedy manner. The **baseline fusion process** [17] is unable to properly assign these as shown in the **2nd and 4th** images of the rightmost column whereas **our method** is able to handle the occlusion relationship present as shown in the **1st and 3rd** images of the rightmost column. **(b)** is an example where Mask R-CNN baseline produces an instance prediction that occludes the entire image and creates the same ambiguity in (a) despite an unambiguous ground truth annotation.

Our approach softens this greedy fusion process with an occlusion head that is dedicated to predicting the binary relation between instances with appreciable overlap so that instance occlusions can be properly handled.

2.2. Occlusion head formulation

Consider two masks M_i and M_j proposed by an instance segmentation model, and denote their intersection as $I_{ij} = M_i \cap M_j$. We are interested in the case where one of the masks is heavily occluded by the other. Therefore, we consider their respective intersection ratios $R_i = \text{Area}(I_{ij})/\text{Area}(M_i)$ and $R_j = \text{Area}(I_{ij})/\text{Area}(M_j)$ where $\text{Area}(M)$ denotes the number of "on" pixels in mask M. As noted in Section 2.1, the fusion process considers the intersection of the current instance proposal with the mask consisting of all already claimed pixels. Here, we are looking at the intersection between two masks and denote the threshold as ρ. If either $R_i \geq \rho$ or $R_j \geq \rho$, we define these two masks as having appreciable overlap. In this case, we must then decide which instance the pixels in I_{ij} should belong to. We attempt to answer this by learning a binary relation $\text{Occlude}(M_i, M_j)$ such that whenever M_i and M_j have appreciable intersection:

$$\text{Occlude}(M_i, M_j) = \begin{cases} 1 \text{ if } M_i \text{ should be placed on top of } M_j \\ 0 \text{ if } M_j \text{ should be placed on top of } M_i. \end{cases} \quad (1)$$

2.3. Fusion with occlusion head

We now describe our modifications to the inference-time fusion heuristic of [17] that incorporates $\text{Occlude}(M_i, M_j)$ in Algorithm 1.

After the instance fusion component has completed, the semantic segmentation is then incorporated as usual, only considering pixels assigned to stuff classes and determining whether the number of unassigned pixels corresponding to the class in the current panoptic output exceeds some threshold, *e.g.*, 4096. The instance fusion process is illustrated in Figure 1.

2.4. Occlusion head architecture

We implement $\text{Occlude}(M_i, M_j)$ as an additional "head" within Mask R-CNN [10]. Mask R-CNN already contains two heads: a box head that is tasked with taking region proposal network (RPN) proposals and refining the bounding box as well as assigning classification scores, while the mask head predicts a fixed size binary mask (usually 28×28) for all classes independently from the output of the box head. Each head derives its own set of features from the underlying FPN. We name our additional head, the "occlusion head" and implement it as a binary classifier that takes two (soft) masks M_i and M_j along with their respective FPN features (determined by their respective boxes) as input. The classifier output is interpreted as the value of $\text{Occlude}(M_i, M_j)$.

Algorithm 1 Fusion with Occlusion Head.

P is $H \times W$ matrix, initially empty.
ρ is a hyperparameter, the minimum intersection ratio for occlusion.
τ is a hyperparameter.

```
for each proposed instance mask M_i do
    C_i = M_i − P        ▷ pixels in M_i that are not assigned in P
    for j < i do                    ▷ each already merged segment
        I_ij is the intersection between mask M_i and M_j.
        R_i = Area(I_ij)/Area(M_i).
        R_j = Area(I_ij)/Area(M_j).
        if R_i ≥ ρ or R_j ≥ ρ then          ▷ significant intersection
            if Occlude(M_i, M_j) = 1 then
                C_i = C_i ⋃ (C_j ⋂ I_ij).
                C_j = C_j − I_ij.
            end if
        end if
    end for
    if Area(C_i)/Area(M_i) ≤ τ then
        continue
    else
        assign the pixels in C_i to the panoptic mask P.
    end if
end for
```

The architecture of occlusion head is inspired by [13] as shown in Figure 2. For two mask representations M_i and M_j, we apply max pooling to produce a 14×14 representation and concatenate each with the corresponding RoI features to produce the input to the head. Three layers of 3×3 convolutions with 512 feature maps and stride 1 are applied before a final one with stride 2. The features are then flattened before a 1024 dimensional fully connected layer and finally projected to a single logit.

2.5. Ground truth occlusion

We use ground truth panoptic mask along with ground truth instance masks to derive ground truth occlusion relation. We pre-compute the intersection between all pairs of masks with appreciable overlap. We then find the pixels corresponding to the intersection of the masks in the panoptic ground truth. We determine the instance occlusion based on which instance owns the majority of pixels in the intersection. We store the resulting "occlusion matrix" for each image in an $N_i \times N_i$ matrix where N_i is the number of instances in the image and the value at position (i, j) is either -1, indicating no occlusion, or encodes the value of Occlude(i, j).

2.6. Occlusion head training

During training, the occlusion head is designed to first find pairs of predicted masks that match to different ground truth instances. Then, the intersection between these pairs of masks is computed, and the ratio of the intersection to mask area taken. A pair of masks is added for consideration when one of these ratios is at least as large as the pre-determined threshold ρ. We then subsample the set of all pairs meeting this criterion to decrease computational cost. It is desirable for the occlusion head to reflect the consistency of Occlude, therefore we also invert all pairs so that Occlude$(M_i, M_j) = 0 \iff$ Occlude$(M_j, M_i) = 1$ whenever the pair (M_i, M_j) meets the intersection criteria. This also mitigates class imbalance. Since this is a binary classification problem, the overall loss L_o from the occlusion head is given by the binary cross-entropy over all subsampled pairs of masks that meet the intersection criteria.

3. Related work

Next, we discuss in detail the difference between OCFusion and the existing approaches for panoptic segmentation, occlusion ordering, and non-maximum suppression.

Panoptic segmentation. The task of panoptic segmentation is introduced in [17] along with a baseline where predictions from instance segmentation (Mask R-CNN [10]) and semantic segmentation (PSPNet [40]) are combined via a heuristics-based fusion strategy. A stronger baseline based on a single Feature Pyramid Network (FPN) [21] backbone followed by multi-task heads consisting of semantic and instance segmentation branches is concurrently proposed by [19, 18, 16, 38]. On top of this baseline, attention layers are added in [19] to the instance segmentation branch, which are guided by the semantic segmentation branch; a loss term enforcing consistency between things and stuff predictions is then introduced in [18]; a parameter-free panoptic head which computes the final panoptic mask by pasting instance mask logits onto semantic segmentation logits is presetned in [38]. These works have been making steady progress in panoptic segmentation, but their focus is not to address the problem for explicit reasoning of instance occlusion.

Occlusion ordering and layout learning. Occlusion handling is a long-studied computer vision task [36, 8, 32, 11]. In the context of semantic segmentation, occlusion ordering has been adopted in [33, 3, 42]. A repulsion loss is added to a pedestrian detection algorithm [37] to deal with the crowd occlusion problem, but it focuses on detection only, without instance segmentation.

In contrast, we study the occlusion ordering problem for instance maps in panoptic segmentation. Closest to our method is the recent work of [23], which proposes a panoptic head to resolve this issue in a similar manner to [38] but instead with a learnable convolution layer. Since our occlusion head can deal with two arbitrary masks, it offers more flexibility over these approaches which attempt to "rerank" the masks in a linear fashion [38, 23]. Furthermore, the approach of [23] is based off how a *class* should be placed on

top of *another class* (akin to semantic segmentation) while we explicitly model the occlusion relation between arbitrary *instances*. This allows us to leverage the *intra-class occlusion relations* such as "which of these two persons should occlude the other?", and we show this leads to a gain in Figure 7 and Table 9. In a nutshell, we tackle the occlusion problem in a scope that is more general than [23] with noticeable performance advantage, as shown in Table 2 and Table 3.

Learnable NMS. One can relate resolving occlusions to non-maximum suppression (NMS) that is applied to *boxes*, while our method tries to suppress intersections between masks. Our method acts as a *learnable* version of NMS for instance masks with similar computations to the analogous ideas for boxes such as [12].

4. Experiments

4.1. Implementation details

We extend the Mask R-CNN benchmark framework [26], built on top of PyTorch, to implement our architecture. Batch normalization [14] layers are frozen and not fine-tuned for simplicity. We perform experiments on the COCO dataset [22] [17] as well as the Cityscapes dataset [4] with panoptic annotations.

We find the most stable and efficient way to train the occlusion head is by fine-tuning with all other parameters frozen. We add a single additional loss only at fine-tuning time so that the total loss during panoptic training is $L = \lambda_i(L_c + L_b + L_m) + \lambda_s L_s$ where L_c, L_b, and L_m are the box head classification loss, bounding-box regression loss, and mask loss while L_s is the semantic segmentation cross-entropy loss. At fine-tuning time, we only minimize L_o, the classification loss from the occlusion head. We subsample 128 mask occlusions per image.

During fusion, we only consider instance masks with detection confidence of at least 0.5 or 0.6 and reject segments during fusion when their overlap ratio with the existing panoptic mask (after occlusions are resolved) exceeds $\tau = 0.5$ on COCO and $\tau = 0.6$ on Cityscapes. Lastly, when considering the segments of *stuff* generated from the semantic segmentation, we only consider those which have at least 4096 pixels remaining after discarding those already assigned on COCO and 2048 on Cityscapes.

Semantic head. On COCO, repeat the combination of 3×3 convolution and $2\times$ bilinear upsampling until $1/4$ scale is reached, following the design of [16]. For the model with ResNeXt-101 backbone, we replace each convolution with deformable convolution [6]. For ResNet-50 backbone, we additionally add one experiment that adopts the design from [38] which uses 2 layers of deformable convolution followed by a bilinear upsampling to the $1/4$ scale. On Cityscapes, we adopt the design from [38].

COCO. The COCO 2018 panoptic segmentation task consists of 80 *thing* and 53 *stuff* classes. We use 2017 dataset which has a split of 118k/5k/20k for training, validation and testing respectively.

Cityscapes. Cityscapes consists of 8 *thing* classes and 11 *stuff* classes. We use only *fine* dataset with a split of 2975/500/1525 for training, validation and testing respectively.

COCO training. We train the FPN-based architecture described in [16] for 90K iterations on 8 GPUs with 1 image per GPU. The base learning rate of 0.02 is reduced by 10 at both 60k and 80k iterations. We then proceed to fine-tune with the occlusion head for 2500 more iterations. We choose $\lambda_i = 1.0$ and $\lambda_s = 0.5$ while for the occlusion head we choose the intersection ratio ρ as 0.2. For models with ResNet-50 and ResNet-101 backbone, we use random horizontal flipping as data augmentation. For model with ResNeXt-101 backbone, we additionally use scale jitter (with scale of shorter image edge equals to $\{640, 680, 720, 760, 800\}$).

Cityscapes training. We randomly rescale each image by 0.5 to $2\times$ (scale factor sampled from a uniform distribution) and construct each batch of 16 (4 images per GPU) by randomly cropping images of size 512×1024. We train for 65k iterations with a base learning rate of 0.01 with decay at 40k and 55k iterations. We fine-tune the occlusion head for 5000 more iterations. We choose $\lambda_i = \lambda_s = 1.0$ with intersection ratio ρ as 0.1. We do not pretrain on COCO.

Panoptic segmentation metrics. We adopt the panoptic quality (PQ) metric from [17] to measure panoptic segmentation performance. This single metric captures both segmentation and recognition quality. PQ can be further broken down into scores specific to *things* and *stuff*, denoted PQ^{Th} and PQ^{St}, respectively.

Multi-scale testing. We adopt the same scales as [38] for both COCO and Cityscapes multi-scale testing. For the stuff branch, we average the multi-scale semantic logits of semantic head. For the thing branch, we average the multi-scale masks and choose not to do bounding box augmentation for simplicity.

Method	**Backbone**	**PQ**	**PQ^{Th}**	**PQ^{St}**
Baseline	ResNet-50	39.5	46.5	29.0
OCFusion	ResNet-50	41.3	49.4	29.0
relative improvement		+1.8	+3.0	+0.0
Baseline	ResNet-101	41.0	47.9	30.7
OCFusion	ResNet-101	43.0	51.1	30.7
relative improvement		+2.0	+3.2	+0.0

Table 1: **Comparison to our implementation of Panoptic FPN [16] baseline model on the MS-COCO *val* dataset**.

Method	Backbone	m.s. test	PQ	PQ^{Th}	PQ^{St}
JSIS-Net [7]	ResNet-50		26.9	29.3	23.3
Panoptic FPN [16]	ResNet-50		39.0	45.9	28.7
Panoptic FPN [16]	ResNet-101		40.3	47.5	29.5
AUNet [19]	ResNet-50		39.6	49.1	25.2
UPSNet* [38]	ResNet-50		42.5	48.5	33.4
UPSNet* [38]	ResNet-50	✓	43.2	49.1	34.1
OANet [23]	ResNet-50		39.0	48.3	24.9
OANet [23]	ResNet-101		40.7	50.0	26.6
AdaptIS [31]	ResNet-50		35.9	40.3	29.3
AdaptIS [31]	ResNet-101		37	41.8	29.9
AdaptIS [31]	ResNeXt-101		42.3	49.2	31.8
OCFusion	ResNet-50		41.3	49.4	29.0
OCFusion*	ResNet-50		42.5	49.1	32.5
OCFusion	ResNet-101		43.0	51.1	30.7
OCFusion*	ResNeXt-101		45.7	53.1	34.5
OCFusion	ResNet-50	✓	41.9	49.9	29.9
OCFusion*	ResNet-50	✓	43.3	50.0	33.8
OCFusion	ResNet-101	✓	43.5	51.5	31.5
OCFusion*	ResNeXt-101	✓	**46.3**	**53.5**	**35.4**

Table 2: **Comparison to prior work on the MS-COCO *val* dataset.** m.s. stands for multi-scale testing. *Used deformable convolution.

Method	Backbone	m.s. test	PQ	PQ^{Th}	PQ^{St}
JSIS-Net [7]	ResNet-50		27.2	29.6	23.4
Panoptic FPN [16]	ResNet-101		40.9	48.3	29.7
OANet [23]	ResNet-101		41.3	50.4	27.7
AUNet [19]	ResNeXt-152	✓	46.5	**55.9**	32.5
UPSNet* [38]	ResNet-101	✓	46.6	53.2	36.7
AdaptIS [31]	ResNeXt-101		42.8	50.1	31.8
OCFusion*	ResNeXt-101	✓	**46.7**	54.0	**35.7**

Table 3: **Comparison to prior work on the MS-COCO *test-dev* dataset.** m.s. stands for multi-scale testing. *Used deformable convolution.

4.2. COCO panoptic benchmark

We obtain state-of-the-art results on COCO Panoptic Segmentation validation set with and without multi-scale testing as is shown in 2. We also obtain single model state-of-the-art results on the COCO test-dev set, as shown in Table 3. In order to show the effectiveness of our method, we compare to our baseline model in Table 1, and the results show that our method consistently provides significant gain on PQ^{Th} as well as PQ.

4.3. Cityscapes panoptic benchmark

We obtain competitive results on the Cityscapes validation set and the best results among models with a ResNet-50 backbone, shown in Table 5. Table 4 shows our strong relative improvement over the baseline on PQ^{Th} as well as PQ.

Method	PQ	PQ^{Th}	PQ^{St}
Baseline	58.6	51.7	63.6
OCFusion	59.3	53.5	63.6
relative improvement	+0.7	+1.7	+0.0

Table 4: **Comparison to our implementation of Panoptic FPN [16] baseline model on the Cityscapes *val* dataset**. All results are based on a ResNet-50 backbone.

Method	m.s. test	PQ	PQ^{Th}	PQ^{St}
Panoptic FPN [16]		57.7	51.6	62.2
AUNet [19]		56.4	52.7	59.0
UPSNet* [38]		59.3	54.6	62.7
UPSNet* [38]	✓	60.1	55.0	63.7
AdaptIS [31]		59.0	**55.8**	61.3
OCFusion*		59.3	53.5	63.6
OCFusion*	✓	**60.2**	54.0	**64.7**

Table 5: **Comparison to prior work on the Cityscapes *val* dataset.** All results are based on a ResNet-50 backbone. m.s. stands for multi-scale testing. *Used deformable convolution.

4.4. Occlusion head performance

In order to better gauge the performance of the occlusion head, we determine its classification accuracy on both COCO and Cityscapes validation dataset at $\rho = 0.20$ with ResNet-50 backbone. We measure the accuracy of the occlusion head in predicting the true ordering given ground truth boxes and masks. The occlusion head classification accuracy on COCO and Cityscapes is 91.58% and 93.60%, respectively, which validates the effectiveness of OCFusion.

4.5. Inference time analysis

We analyze the computational cost of our method and empirically show the inference time overhead of our method compared to the baseline model. While our method incurs an $O(n^2)$ cost in order to compute pairwise intersections, where n is the number of instances, this computation is only needed for the subset of masks whose detection confidence is larger than a threshold (0.5 or 0.6 usually) as dictated by the Panoptic FPN [16] baseline. This filtering greatly limits

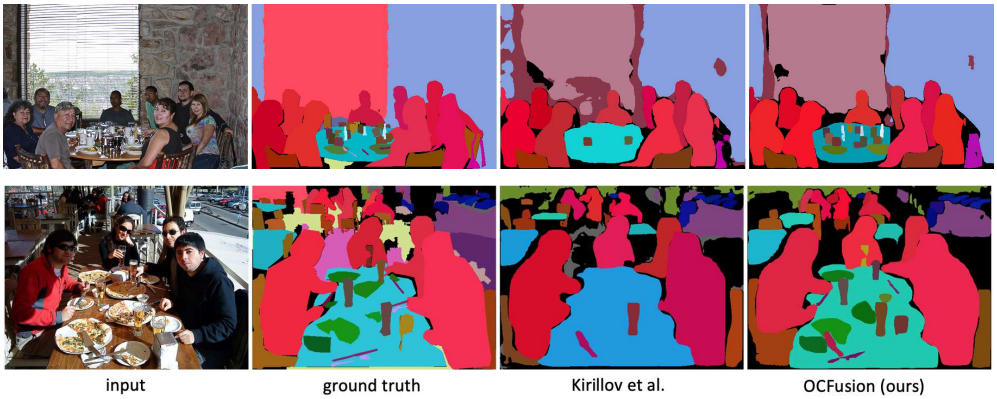


Figure 4: **Comparison against Kirillov et al. [16] which uses fusion by confidence.**

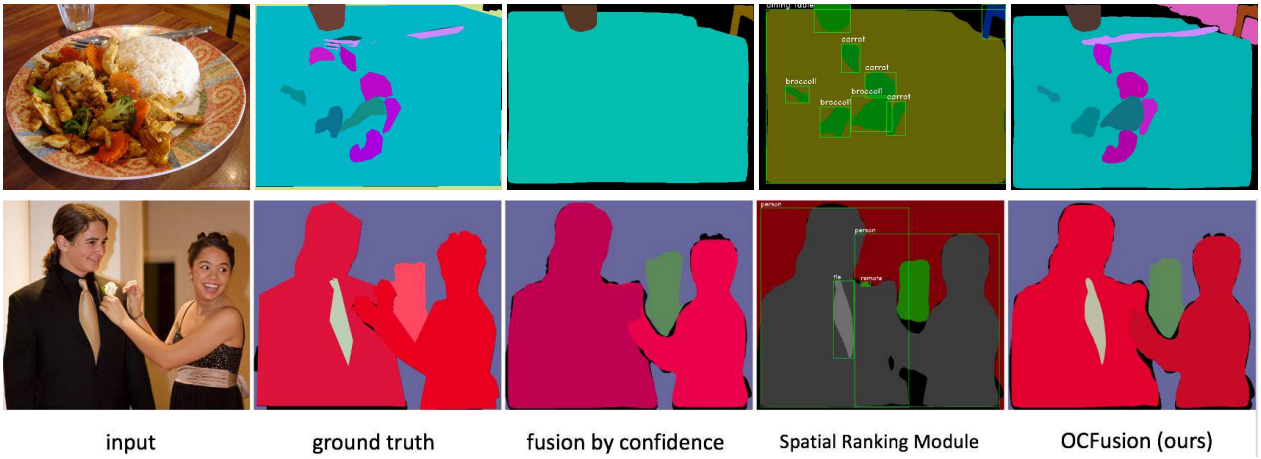


Figure 5: **Comparison against Spatial Ranking Module [23].**

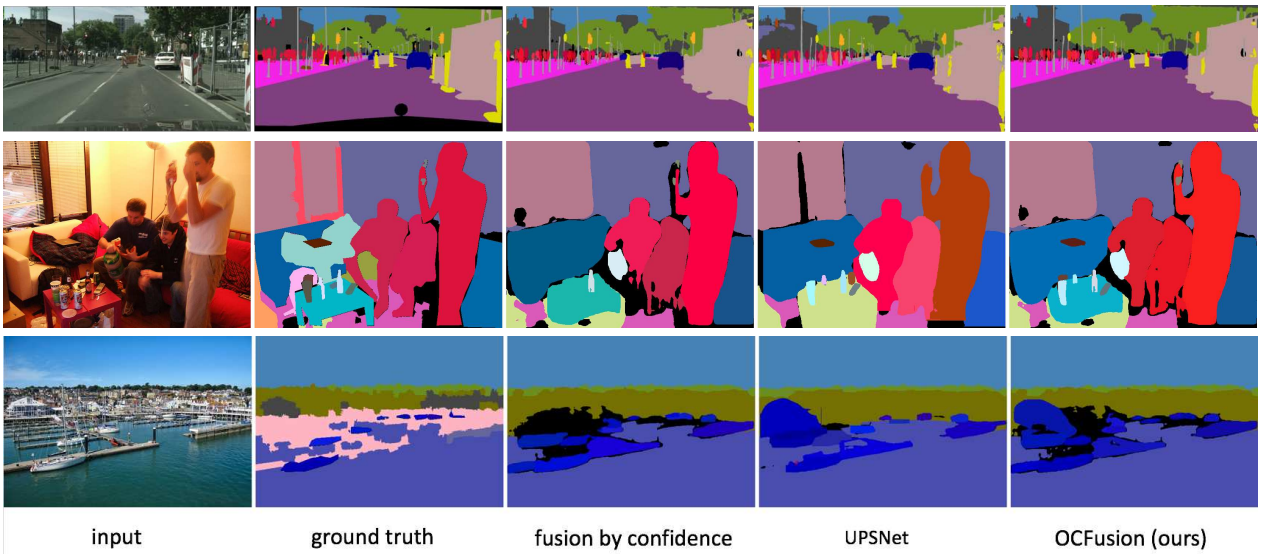


Figure 6: **Comparison against UPSNet [38].**

the practical magnitude of n. Furthermore, only the subset of remaining mask pairs that have appreciable overlap (larger than ρ) requires evaluation by the occlusion head. We measure this inference time overhead in Table 6. OCFusion incurs a modest 2.0% increase in computational time on COCO and 4.7% increase on Cityscapes.

Method	COCO	Cityscapes
Baseline	153	378
OCFusion	156	396
Change in runtime (ms)	+3	+18

Table 6: **Runtime (ms) overhead per image.** Runtime results are averaged over the entire COCO and Cityscapes validation dataset. We use a single GeForce GTX 1080 Ti GPU and Xeon(R) CPU E5-2687W CPU.

4.6. Visual comparisons

Since panoptic segmentation is a relatively new task, the most recent papers offer only comparisons against the baseline presented in [17]. We additionally compare with a few other recent methods [23, 38].

We first compare our method against [16] in Figure 4 as well as two recent works: UPSNet [38] in Figure 6 and the Spatial Ranking Module of [23] in Figure 5. The latter two have similar underlying architectures alongside modifications to their fusion process. We note that except for comparisons between [16], the comparison images shown are those *included in the respective papers and not of our own choosing*. Overall, we see that our method is able to preserve a significant number of instance occlusions lost by other methods while maintaining more realistic fusions, *e.g.*, the arm is entirely above the man versus sinking behind partly as in "fusion by confidence".

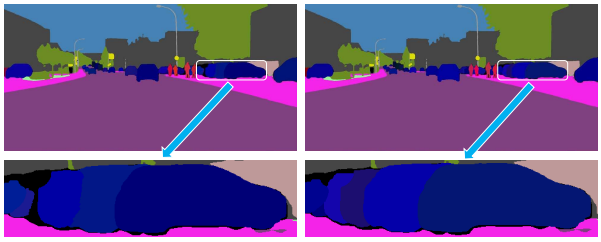


Figure 7: **Comparison for w/o (left) or w/ (right) intra-class capability enabled.** Best viewed in color.

4.7. Ablation experiments

We study the sensitivity of our method to the hyperparameters τ and ρ in Table 7 for COCO and Table 8 for Cityscapes. We also include the number of examples of occlusions we are able to collect at the given ρ denoted as N. Naturally, a larger ρ leads to less spurious occlusions but decreases the overall number of examples that the occlusion head is able to learn from.

(τ, ρ)	0.05	0.10	0.20
0.4	41.27 (Th: 49.43, St: 28.97)	41.22 (Th: 49.33, St: 28.97)	41.20 (Th: 49.30, St: 28.97)
0.5	41.20 (Th: 49.32, St: 28.95)	41.15 (Th: 49.23, St: 28.95)	41.24 (Th: 49.29, St: 29.10)
0.6	41.09 (Th: 49.15, St: 28.93)	41.03 (Th: 49.03, St: 28.93)	41.02 (Th: 49.02, St: 28.93)
N	192,519	157,784	132,165

Table 7: **COCO Hyperparameter Ablation: PQ**

(τ, ρ)	0.05	0.10	0.20
0.4	58.76 (Th: 52.10, St: 63.62)	59.15 (Th: 53.00, St: 63.62)	59.07 (Th: 52.80, St: 63.63)
0.5	59.18 (Th: 53.09, St: 63.61)	59.26 (Th: 53.28, St: 63.61)	59.22 (Th: 53.19, St: 63.61)
0.6	59.21 (Th: 53.17, St: 63.61)	59.33 (Th: 53.46, St: 63.60)	58.70 (Th: 51.96, St: 61.60)
N	33,391	29,560	6,617

Table 8: **Cityscapes Hyperparameter Ablation: PQ**

Intra-class instance occlusion in Cityscapes is a challenging problem, also noted in [10]. Since we can enable inter-class or intra-class occlusion query ability independently, we show ablation results in Table 9 that highlight the importance of being able to handle intra-class occlusion on. We believe this sets our method apart from others, *e.g.*, [23] which simplifies the problem by handling inter-class occlusion only. Additionally, Figure 7 shows a visual comparison between resulting panoptic segmentations when intra-class occlusion handling is toggled on Cityscapes. Only the model with intra-class handling enabled can handle the occluded cars better during the fusion process.

Inter-class	Intra-class	PQ	PQ^{Th}	PQ^{St}
		58.6	51.7	63.6
✓		59.2 (+0.5)	53.0 (+1.3)	63.6 (+0.0)
✓	✓	59.3 (+0.7)	53.5 (+1.7)	63.6 (+0.0)

Table 9: **Ablation study on different types of occlusion on the Cityscapes *val* dataset.** ✓means capability enabled.

5. Conclusion

We have introduced an *explicit* notion of instance occlusion to Mask R-CNN so that instances may be effectively fused when producing a panoptic segmentation. We assemble a dataset of occlusions already present in the COCO and Cityscapes datasets and then learn an additional head for Mask R-CNN tasked with predicting occlusion between two masks. Adding occlusion head on top of Panoptic FPN incurs minimal overhead, and we show that it is effective even when trained for few thousand iterations. In the future, we hope to explore how further understanding of occlusion, including relationships of *stuff*, could be helpful.

Acknowledgements. This work is supported by NSF IIS-1618477 and IIS-1717431. We thank Yifan Xu, Weijian Xu, Sainan Liu, Yu Shen, and Subarna Tripathi for valuable discussions.

References

[1] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987. 1

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018. 1

[3] Yi-Ting Chen, Xiaokai Liu, and Ming-Hsuan Yang. Multi-instance object segmentation with occlusion handling. In *CVPR*, 2015. 4

[4] 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 *CVPR*, 2016. 5

[5] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016. 1

[6] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 5

[7] Daan de Geus, Panagiotis Meletis, and Gijs Dubbelman. Panoptic segmentation with a joint semantic and instance segmentation network. *arXiv preprint arXiv:1809.02110*, 2018. 6

[8] Markus Enzweiler, Angela Eigenstetter, Bernt Schiele, and Dariu M Gavrila. Multi-cue pedestrian classification with partial occlusion handling. In *CVPR*, 2010. 4

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 1

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1, 2, 3, 4, 8

[11] Derek Hoiem, Andrew N Stein, Alexei A Efros, and Martial Hebert. Recovering occlusion boundaries from a single image. In *ICCV*, 2007. 4

[12] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression. In *CVPR*, 2017. 5

[13] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *CVPR*, 2019. 3, 4

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 5

[15] Long Jin, Zeyu Chen, and Zhuowen Tu. Object detection free instance segmentation with labeling transformations. *arXiv preprint arXiv:1611.08991*, 2016. 1

[16] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019. 1, 2, 4, 5, 6, 7, 8

[17] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019. 1, 2, 3, 4, 5, 8

[18] Jie Li, Allan Raventos, Arjun Bhargava, Takaaki Tagawa, and Adrien Gaidon. Learning to fuse things and stuff. *arXiv preprint arXiv:1812.01192*, 2018. 4

[19] Yanwei Li, Xinze Chen, Zheng Zhu, Lingxi Xie, Guan Huang, Dalong Du, and Xingang Wang. Attention-guided unified network for panoptic segmentation. In *CVPR*, 2019. 1, 4, 6

[20] Xiaodan Liang, Liang Lin, Yunchao Wei, Xiaohui Shen, Jianchao Yang, and Shuicheng Yan. Proposal-free network for instance-level object segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2978–2991, 2018. 1

[21] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 2, 4

[22] 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. 5

[23] Huanyu Liu, Chao Peng, Changqian Yu, Jingbo Wang, Xu Liu, Gang Yu, and Wei Jiang. An end-to-end network for panoptic segmentation. In *CVPR*, 2019. 4, 5, 6, 7, 8

[24] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[25] David Marr. Vision: A computational investigation into the human representation and processing of visual information, henry holt and co. *Inc., New York, NY*, 2(4.2), 1982. 1

[26] Francisco Massa and Ross Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. https://github.com/facebookresearch/maskrcnn-benchmark, 2018. Accessed: January 5, 2019. 5

[27] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollar. Learning to segment object candidates. In *NIPS*, 2015. 1

[28] Pedro O. Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *ECCV*, 2016. 1

[29] Hayko Riemenschneider, Sabine Sternig, Michael Donoser, Peter M Roth, and Horst Bischof. Hough regions for joining instance localization and segmentation. In *ECCV*. 2012. 1

[30] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006. 1

[31] Konstantin Sofiiuk, Olga Barinova, and Anton Konushin. Adaptis: Adaptive instance selection network. In *ICCV*, 2019. 6

[32] Jian Sun, Yin Li, Sing Bing Kang, and Heung-Yeung Shum. Symmetric stereo matching for occlusion handling. In *CVPR*, 2005. 4

[33] Joseph Tighe, Marc Niethammer, and Svetlana Lazebnik. Scene parsing with object instances and occlusion ordering. In *CVPR*, 2014. 1, 4

[34] Zhuowen Tu. Auto-context and its application to high-level vision tasks. In *CVPR*, 2008. 1

[35] Zhuowen Tu, Xiangrong Chen, Alan L Yuille, and Song-Chun Zhu. Image parsing: Unifying segmentation, detection, and recognition. *International Journal of computer vision*, 63(2):113–140, 2005. 1

[36] Xiaoyu Wang, Tony X Han, and Shuicheng Yan. An hog-lbp human detector with partial occlusion handling. In *ICCV*, 2009. 4

[37] Xinlong Wang, Tete Xiao, Yuning Jiang, Shuai Shao, Jian Sun, and Chunhua Shen. Repulsion loss: Detecting pedestrians in a crowd. In *CVPR*, 2018. 4

[38] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *CVPR*, 2019. 1, 4, 5, 6, 7, 8

[39] Ziyu Zhang, Sanja Fidler, and Raquel Urtasun. Instance-level segmentation for autonomous driving with deep densely connected mrfs. In *CVPR*, 2016. 1

[40] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1, 4

[41] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 1

[42] Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *CVPR*, 2017. 4