Panoptic Segmentation

Alexander Kirillov[1,2] Kaiming He[1] Ross Girshick[1] Carsten Rother[2] Piotr Dollár[1]
[1]Facebook AI Research (FAIR) [2]HCI/IWR, Heidelberg University, Germany

Abstract

We propose and study a novel panoptic segmentation *(PS) task. Panoptic segmentation unifies the typically distinct tasks of* semantic segmentation *(assign a class label to each pixel) and* instance segmentation *(detect and segment each object instance). The proposed task requires generating a* coherent *scene segmentation that is rich and complete, an important step toward real-world vision systems. While early work in computer vision addressed related image/scene parsing tasks, these are not currently popular, possibly due to lack of appropriate metrics or associated recognition challenges. To address this, we first propose a novel* panoptic quality *(PQ) metric that captures performance for all classes (stuff and things) in an interpretable and unified manner. Using the proposed metric, we perform a rigorous study of both human and machine performance for PS on three existing datasets, revealing interesting insights about the task. Second, we are working to introduce panoptic segmentation tracks at upcoming recognition challenges. The aim of our work is to revive the interest of the community in a more unified view of image segmentation.*

1. Introduction

In the early days of computer vision, *things* – countable objects such as people, animals, tools – received the dominant share of attention. Questioning the wisdom of this trend, Adelson [1] elevated the importance of studying systems that recognize *stuff* – amorphous regions of similar texture or material such as grass, sky, road. This dichotomy between stuff and things persists to this day, reflected in both the division of visual recognition tasks and in the specialized algorithms developed for stuff and thing tasks.

Studying stuff is most commonly formulated as a task known as *semantic segmentation*, see Figure 1b. As stuff is amorphous and uncountable, this task is defined as simply assigning a class label to each pixel in an image (note that semantic segmentation treats thing classes as stuff). In contrast, studying things is typically formulated as the task of *object detection* or *instance segmentation*, where the goal is to detect each object and delineate it with a bounding box or segmentation mask, respectively, see Figure 1c. While seemingly related, the datasets, details, and metrics for these two visual recognition tasks vary substantially.

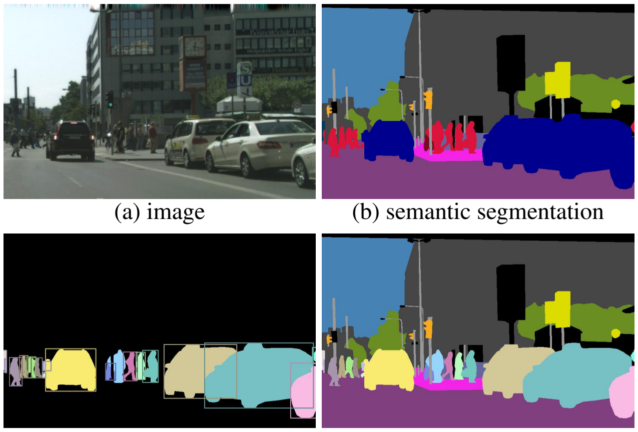
(a) image
(b) semantic segmentation
(c) instance segmentation
(d) panoptic segmentation

Figure 1: For a given (a) image, we show *ground truth* for: (b) semantic segmentation (per-pixel class labels), (c) instance segmentation (per-object mask and class label), and (d) the proposed *panoptic segmentation* task (per-pixel class+instance labels). The PS task: (1) encompasses both stuff and thing classes, (2) uses a simple but general format, and (3) introduces a uniform evaluation metric for all classes. Panoptic segmentation generalizes both semantic and instance segmentation and we expect the unified task will present novel challenges and enable innovative new methods.

The schism between semantic and instance segmentation has led to a parallel rift in the methods for these tasks. Stuff classifiers are usually built on fully convolutional nets [26] with dilations [46, 5] while object detectors often use object proposals [15] and are region-based [33, 14]. Overall algorithmic progress on these tasks has been incredible in the past decade, yet, something important may be overlooked by focussing on these tasks in isolation.

A natural question emerges: *Can there be a reconciliation between stuff and things?* And what is the most effective design of a unified vision system that generates rich and coherent scene segmentations? These questions are particularly important given their relevance in real-world applications, such as autonomous driving or augmented reality.

Interestingly, while semantic and instance segmentation dominate current work, in the pre-deep learning era there was interest in the joint task described using various names such as *scene parsing* [38], *image parsing* [39], or *holistic scene understanding* [45]. Despite its practical relevance, this general direction is not currently popular, perhaps due to lack of appropriate metrics or recognition challenges.

In our work we aim to revive this direction. We propose a task that: *(1) encompasses both stuff and thing classes, (2) uses a simple but general output format, and (3) introduces a uniform evaluation metric*. To clearly disambiguate with previous work, we refer to the resulting task as *panoptic segmentation* (PS). The definition of 'panoptic' is "including everything visible in one view", in our context panoptic refers to a unified, global view of segmentation.

The **task format** we adopt for panoptic segmentation is simple: each pixel of an image must be assigned a semantic label and an instance id. Pixels with the same label and id belong to the same object; for stuff labels the instance id is ignored. See Figure 1d for a visualization. This format has been adopted previously, especially by methods that produce non-overlapping instance segmentations [18, 24, 2]. We adopt it for our joint task that includes stuff and things.

A fundamental aspect of panoptic segmentation is the **task metric** used for evaluation. While numerous existing metrics are popular for either semantic or instance segmentation, these metrics are best suited either for stuff or things, respectively, but not both. We believe that the use of disjoint metrics is one of the primary reasons the community generally studies stuff and thing segmentation in isolation. To address this, we introduce the *panoptic quality* (PQ) metric in §4. PQ is *simple* and *informative* and most importantly can be used to measure the performance for both stuff and things in a *uniform* manner. Our hope is that the proposed joint metric will aid in the broader adoption of the joint task.

The panoptic segmentation task encompasses both semantic and instance segmentation but introduces new algorithmic challenges. Unlike semantic segmentation, it requires differentiating individual object instances; this poses a challenge for fully convolutional nets. Unlike instance segmentation, object segments must be *non-overlapping*; this presents a challenge for region-based methods that operate on each object independently. Generating coherent image segmentations that resolve inconsistencies between stuff and things is an important step toward real-world uses.

As both the ground truth and algorithm format for PS must take on the same form, we can perform a detailed study of *human performance* on panoptic segmentation. This allows us to understand the PQ metric in more detail, including detailed breakdowns of recognition *vs*. segmentation and stuff *vs*. things performance. Moreover, measuring human PQ helps ground our understanding of machine performance. This is important as it will allow us to monitor performance saturations on various datasets for PS.

Finally we perform an initial study of machine performance for PS. To do so, we define a simple and likely suboptimal heuristic that combines the output of two *independent* systems for semantic and instance segmentation via a series of post-processing steps that merges their outputs (in essence, a sophisticated form of non-maximum suppression). Our heuristic establishes a baseline for PS and gives us insights into the main algorithmic challenges it presents.

We study both human and machine performance on three popular segmentation datasets that have both stuff and things annotations. This includes the Cityscapes [6], ADE20k [49], and Mapillary Vistas [31] datasets. For each of these datasets, we obtained results of state-of-the-art methods directly from the challenge organizers. In the future we will extend our analysis to COCO [22] on which stuff is being annotated [4]. Together our results on these datasets form a solid foundation for the study of both human and machine performance on panoptic segmentation.

We are currently working with challenge organizers from the COCO [22], Vistas [31], and ADE20k [49] datasets to feature a panoptic segmentation track. We believe including a PS track alongside existing instance and semantic segmentation tracks on these popular recognition datasets will help lead to a broader adoption of the proposed joint task.

2. Related Work

Novel datasets and tasks have played a key role throughout the history of computer vision. They help catalyze progress and enable breakthroughs in our field, and just as importantly, they help us measure and recognize the progress our community is making. For example, ImageNet [34] helped drive the recent popularization of deep learning techniques for visual recognition [20] and exemplifies the potential transformational power that datasets and tasks can have. Our goals for introducing the panoptic segmentation task are similar: to challenge our community, to drive research in novel directions, and to enable both expected and unexpected innovation. We review related tasks next.

Object detection tasks. Early work on face detection using ad-hoc datasets (*e.g*., [40, 42]) helped popularize bounding-box object detection. Later, pedestrian detection datasets [8] helped drive progress in the field. The PASCAL VOC dataset [9] upgraded the task to a more diverse set of general object classes on more challenging images. More recently, the COCO dataset [22] pushed detection towards the task of instance segmentation. By framing this task and providing a high-quality dataset, COCO helped define a new and exciting research direction and led to many recent breakthroughs in instance segmentation [32, 21, 14]. Our general goals for panoptic segmentation are similar.

Semantic segmentation tasks. Semantic segmentation datasets have a rich history [35, 23, 9] and helped drive key innovations (*e.g.*, fully convolutional nets [26] were developed using [23, 9]). These datasets contain both stuff and thing classes, but don't distinguish individual object instances. Recently the field has seen numerous new segmentation datasets including Cityscapes [6], ADE20k [49], and Mapillary Vistas [31]. These datasets actually support both semantic and instance segmentation, and each has opted to have a separate track for the two tasks. Importantly, they contain all of the information necessary for PS. In other words, *the panoptic segmentation task can be bootstrapped on these datasets without any new data collection.*

Multitask learning. With the success of deep learning for many visual recognition tasks, there has been substantial interest in *multitask learning* approaches that have broad competence and can solve multiple diverse vision problems in a single framework [19, 28, 30]. *E.g.*, UberNet [19] solves multiple low to high-level visual tasks, including object detection and semantic segmentation, using a single network. While there is significant interest in this area, we emphasize that panoptic segmentation is *not* a multitask problem but rather a single, *unified* view of image segmentation. Specifically, the multitask setting allows for independent and potentially inconsistent outputs for stuff and things, while PS requires a single coherent scene segmentation.

Joint segmentation tasks. In the pre-deep learning era, there was substantial interest in generating coherent scene interpretations. The seminal work on image parsing [39] proposed a general bayesian framework to jointly model segmentation, detection, and recognition. Later, approaches based on graphical models studied consistent stuff and thing segmentation [45, 37, 38, 36]. While these methods shared a common motivation, there was no agreed upon task definition, and different output formats and varying evaluation metrics were used, including separate metrics for evaluating results on stuff and thing classes. In recent years this direction has become less popular, perhaps for these reasons.

In our work we aim to revive this general direction, but in contrast to earlier work, we focus on the task itself. Specifically, as discussed, PS: (1) addresses both stuff and thing classes, (2) uses a simple format, and (3) introduces a uniform metric for both stuff and things. Previous work on joint segmentation uses varying formats and disjoint metrics for evaluating stuff and things. Methods that generate non-overlapping instance segmentations [18, 3, 24, 2] use the same format as PS, but these methods typically only address thing classes. By addressing both stuff and things, using a simple format, and introducing a uniform metric, we hope to encourage broader adoption of the joint task.

Amodal segmentation task. In [50] objects are annotated *amodally*: the full extent of each region is marked, not just the visible. Our work focuses on segmentation of all *visible* regions, but an extension of panoptic segmentation to the amodal setting is an interesting direction for future work.

3. Panoptic Segmentation Format

Task format. The format for panoptic segmentation is simple to define. Given a predetermined set of L semantic classes encoded by $\mathcal{L} := \{0, \dots, L-1\}$, the task requires a *panoptic segmentation algorithm* to map each pixel i of an image to a pair $(l_i, z_i) \in \mathcal{L} \times \mathbb{N}$, where l_i represents the semantic class of pixel i and z_i represents its instance id. The z_i's group pixels of the same class into distinct segments. Ground truth annotations are encoded identically. Ambiguous or out-of-class pixels can be assigned a special void label; *i.e.*, not all pixels must have a semantic label.

Stuff and thing labels. The semantic label set consists of subsets $\mathcal{L}^{\text{St}}$ and $\mathcal{L}^{\text{Th}}$, such that $\mathcal{L} = \mathcal{L}^{\text{St}} \cup \mathcal{L}^{\text{Th}}$ and $\mathcal{L}^{\text{St}} \cap \mathcal{L}^{\text{Th}} = \emptyset$. These subsets correspond to *stuff* and *thing* labels, respectively. When a pixel is labeled with $l_i \in \mathcal{L}^{\text{St}}$, its corresponding instance id z_i is irrelevant. That is, for stuff classes all pixels belong to the same instance (*e.g.*, the same *sky*). Otherwise, all pixels with the same (l_i, z_i) assignment, where $l_i \in \mathcal{L}^{\text{Th}}$, belong to the same instance (*e.g.*, the same *car*), and conversely, all pixels belonging to a single instance must have the same (l_i, z_i). The selection of which classes are stuff *vs.* things is a design choice left to the creator of the dataset, just as in previous datasets.

Relationship to semantic segmentation. The PS task format is a strict generalization of the format for semantic segmentation. Indeed, both tasks require each pixel in an image to be assigned a semantic label. If the ground truth does not specify instances, or all classes are stuff, then the task formats are identical (although the task metrics differ). In addition, inclusion of thing classes, which may have multiple instances per image, differentiates the tasks.

Relationship to instance segmentation. The instance segmentation task requires a method to segment each object instance in an image. However, it allows overlapping segments, whereas the panoptic segmentation task permits only one semantic label and one instance id to be assigned to each pixel. Hence, for PS, no overlaps are possible by construction. In the next section we show that this difference plays an important role in performance evaluation.

Confidence scores. Like semantic segmentation, but unlike instance segmentation, we do *not* require confidence scores associated with each segment for PS. This makes the panoptic task *symmetric* with respect to humans and machines: both must generate the same type of image annotation. It also makes evaluating human performance for PS simple. This is in contrast to instance segmentation, which is not easily amenable to such a study as human annotators do not provide explicit confidence scores (though a single precision/recall point may be measured). We note that confidence scores give downstream systems more information, which can be useful, so it may still be desirable to have a PS algorithm generate confidence scores in certain settings.

4. Panoptic Segmentation Metric

In this section we introduce a new metric for panoptic segmentation. We begin by noting that existing metrics are specialized for either semantic or instance segmentation and cannot be used to evaluate the joint task involving both stuff and thing classes. Previous work on joint segmentation sidestepped this issue by evaluating stuff and thing performance using independent metrics (*e.g.* [45, 37, 38, 36]). However, this introduces challenges in algorithm development, makes comparisons more difficult, and hinders communication. We hope that introducing a unified metric for stuff and things will encourage the study of the unified task.

Before going into further details, we start by identifying the following desiderata for a suitable metric for PS:

Completeness. The metric should treat stuff and thing classes in a uniform way, capturing all aspects of the task.

Interpretability. We seek a metric with identifiable meaning that facilitates communication and understanding.

Simplicity. In addition, the metric should be simple to define and implement. This improves transparency and allows for easy reimplementation. Related to this, the metric should be efficient to compute to enable rapid evaluation.

Guided by these principles, we propose a new *panoptic quality* (PQ) metric. PQ measures the quality of a predicted panoptic segmentation relative to the ground truth. It involves two steps: (1) segment matching and (2) PQ computation given the matches. We describe each step next then return to a comparison to existing metrics.

4.1. Segment Matching

We specify that a predicted segment and a ground truth segment can match only if their intersection over union (IoU) is strictly greater than 0.5. This requirement, together with the non-overlapping property of a panoptic segmentation, gives a *unique matching*: there can be at most one predicted segment matched with each ground truth segment.

Theorem 1. *Given a predicted and ground truth panoptic segmentation of an image, each ground truth segment can have at most one corresponding predicted segment with IoU strictly greater than 0.5 and vice verse.*

Proof. Let g be a ground truth segment and p_1 and p_2 be two predicted segments. By definition, $p_1 \cap p_2 = \emptyset$ (they do not overlap). Since $|p_i \cup g| \geq |g|$, we get the following:

$$\text{IoU}(p_i, g) = \frac{|p_i \cap g|}{|p_i \cup g|} \leq \frac{|p_i \cap g|}{|g|} \quad \text{for } i \in \{1, 2\}\,.$$

Summing over i, and since $|p_1 \cap g| + |p_2 \cap g| \leq |g|$ due to the fact that $p_1 \cap p_2 = \emptyset$, we get:

$$\text{IoU}(p_1, g) + \text{IoU}(p_2, g) \leq \frac{|p_1 \cap g| + |p_2 \cap g|}{|g|} \leq 1\,.$$

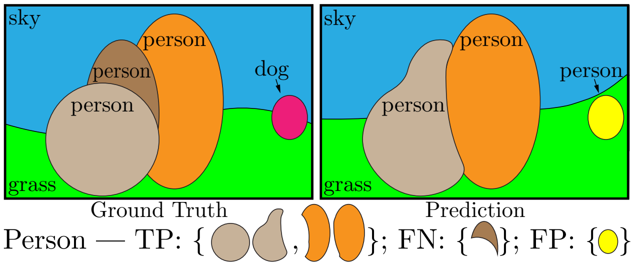

Figure 2: Toy illustration of ground truth and predicted panoptic segmentations of an image. Pairs of segments of the same color have IoU larger than 0.5 and are therefore matched. We show how the segments for the *person* class are partitioned into true positives TP, false negatives FN, and false positives FP.

Therefore, if $\text{IoU}(p_1, g) > 0.5$, then $\text{IoU}(p_2, g)$ has to be smaller than 0.5. Reversing the role of p and g can be used to prove that only one ground truth segment can have IoU with a predicted segment strictly greater than 0.5. □

The requirement that matches must have IoU greater than 0.5, which in turn yields the unique matching theorem, achieves two of our desired properties. First, it is *simple* and efficient as correspondences are unique and trivial to obtain. Second, it is *interpretable* and easy to understand (and does not require solving a complex matching problem as is commonly the case for these types of metrics [13, 44]).

Note that due to the uniqueness property, for IoU > 0.5, any reasonable matching strategy (including greedy and optimal) will yield an identical matching. For smaller IoU other matching techniques would be required; however, in the experiments we will show that lower thresholds are unnecessary as matches with IoU ≤ 0.5 are rare in practice.

4.2. PQ Computation

We calculate PQ for each class independently and average over classes. This makes PQ insensitive to class imbalance. For each class, the unique matching splits the predicted and ground truth segments into three sets: true positives (TP), false positives (FP), and false negatives (FN), representing matched pairs of segments, unmatched predicted segments, and unmatched ground truth segments, respectively. An example is illustrated in Figure 2. Given these three sets, PQ is defined as:

$$\text{PQ} = \frac{\sum_{(p,g) \in TP} \text{IoU}(p, g)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}\,. \quad (1)$$

PQ is intuitive after inspection: $\frac{1}{|TP|}\sum_{(p,g) \in TP} \text{IoU}(p, g)$ is simply the average IoU of matched segments, while $\frac{1}{2}|FP| + \frac{1}{2}|FN|$ is added to the denominator to penalize segments without matches. Note that all segments receive equal importance regardless of their area. Furthermore, if we multiply and divide PQ by the size of the TP set, then

PQ can be seen as the multiplication of a *segmentation quality* (SQ) term and a *recognition quality* (RQ) term:

$$\text{PQ} = \underbrace{\frac{\sum_{(p,g)\in TP} \text{IoU}(p,g)}{|TP|}}_{\text{segmentation quality (SQ)}} \times \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\text{recognition quality (RQ)}}. \quad (2)$$

Written this way, RQ is the familiar F_1 score [41] widely used for quality estimation in detection settings [29]. SQ is simply the average IoU of matched segments. We find the decomposition of PQ = SQ × RQ to provide insight for analysis. We note, however, that the two values are not independent since SQ is measured only over matched segments.

Our definition of PQ achieves our desiderata. It measures performance of all classes in a uniform way using a simple and interpretable formula. We conclude by discussing how we handle void regions and groups of instances [22].

Void labels. There are two sources of void labels in the ground truth: (a) out of class pixels and (b) ambiguous or unknown pixels. As often we cannot differentiate these two cases, we don't evaluate predictions for void pixels. Specifically: (1) during matching, all pixels in a predicted segment that are labeled as void in the ground truth are removed from the prediction and do not affect IoU computation, and (2) after matching, unmatched predicted segments that contain a fraction of void pixels over the matching threshold are removed and do not count as false positives. Finally, outputs may also contain void pixels; these do not affect evaluation.

Group labels. A common annotation practice [6, 22] is to use a group label instead of instance ids for adjacent instances of the same semantic class if accurate delineation of each instance is difficult. For computing PQ: (1) during matching, group regions are not used, and (2) after matching, unmatched predicted segments that contain a fraction of pixels from a group of the same class over the matching threshold are removed and do not count as false positives.

4.3. Comparison to Existing Metrics

We conclude by comparing PQ to existing metrics for semantic and instance segmentation.

Semantic segmentation metrics. Common metrics for semantic segmentation include pixel accuracy, mean accuracy, and IoU [26]. These metrics are computed based only on pixel outputs/labels and completely ignore object-level labels. For example, IoU is the ratio between correctly predicted pixels and total number of pixels in either the prediction or ground truth for each class. As these metrics ignore instance labels, they are not well suited for evaluating thing classes. Finally, please note that IoU for semantic segmentation is distinct from our segmentation quality (SQ), which is computed as the average IoU over *matched segments*.

Instance segmentation metrics. The standard metric for instance segmentation is Average Precision (AP) [22, 13]. AP requires each object segment to have a confidence score to estimate a precision/recall curve. Note that while confidence scores are quite natural for object detection, they are not used for semantic segmentation. Hence, AP cannot be used for measuring the output of semantic segmentation, or likewise of PS (see also the discussion of confidences in §3).

Panoptic quality. PQ treats all classes (stuff and things) in a uniform way. We note that while decomposing PQ into SQ and RQ is helpful with interpreting results, PQ is *not* a combination of semantic and instance segmentation metrics. Rather, SQ and RQ are computed for every class (stuff and things), and measure segmentation and recognition quality, respectively. PQ thus unifies evaluation over all classes. We support this claim with rigorous experimental evaluation of PQ in §7, including comparisons to IoU and AP for semantic and instance segmentation, respectively.

5. Panoptic Segmentation Datasets

To our knowledge only three public datasets have both dense semantic and instance segmentation annotations: Cityscapes [6], ADE20k [49], and Mapillary Vistas [31]. We use all three datasets for panoptic segmentation. In addition, in the future we will extend our analysis to COCO [22] on which stuff is currently being annotated [4][1].

Cityscapes [6] has 5000 images (2975 train, 500 val, and 1525 test) of ego-centric driving scenarios in urban settings. It has dense pixel annotations (97% coverage) of 19 classes among which 8 have instance-level segmentations.

ADE20k [49] has over 25k images (20k train, 2k val, 3k test) that are densely annotated with an open-dictionary label set. For the 2017 Places Challenge[2], 100 thing and 50 stuff classes that cover 89% of all pixels are selected. We use this closed vocabulary in our study.

Mapillary Vistas [31] has 25k street-view images (18k train, 2k val, 5k test) in a wide range of resolutions. The 'research edition' of the dataset is densely annotated (98% pixel coverage) with 28 stuff and 37 thing classes.

6. Human Performance Study

One advantage of panoptic segmentation is that it enables measuring human performance. Aside from this being interesting as an end in itself, human performance studies allow us to understand the task in detail, including details of our proposed metric and breakdowns of human performance along various axes. This gives us insight into intrinsic challenges posed by the task without biasing our analysis by algorithmic choices. Furthermore, human studies help ground machine performance (discussed in §7) and allow us to calibrate our understanding of the task.

[1] In addition to stuff annotations being incomplete, COCO instance segmentations contain overlaps. We plan on collecting depth ordering for all pairs of overlapping instances in COCO to resolve these overlaps.

[2] http://placeschallenge.csail.mit.edu

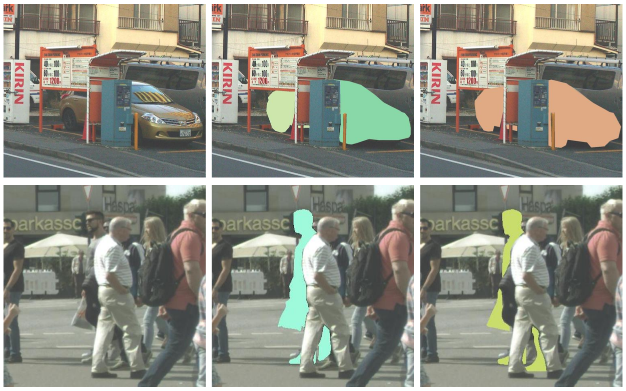

Figure 3: **Segmentation flaws.** Images are zoomed and cropped. Top row (Vistas image): both annotators identify the object as a car, however, one splits the car into two cars. Bottom row (Cityscapes image): the segmentation is genuinely ambiguous.

	PQ	PQ$^{\text{St}}$	PQ$^{\text{Th}}$	SQ	SQ$^{\text{St}}$	SQ$^{\text{Th}}$	RQ	RQ$^{\text{St}}$	RQ$^{\text{Th}}$
Cityscapes	69.7	71.3	67.4	84.2	84.4	83.9	82.1	83.4	80.2
ADE20k	67.1	70.3	65.9	85.8	85.5	85.9	78.0	82.4	76.4
Vistas	57.5	62.6	53.4	79.5	81.6	77.9	71.4	76.0	67.7

Table 1: **Human performance for stuff *vs.* things**. Panoptic, segmentation, and recognition quality (PQ, SQ, RQ) averaged over classes (PQ=SQ×RQ per class) are reported as percentages. Perhaps surprisingly, we find that human performance on each dataset is relatively similar for both stuff and things.

Human annotations. To enable human performance analysis, dataset creators graciously supplied us with 30 doubly annotated images for Cityscapes, 64 for ADE20k, and 46 for Vistas. For Cityscapes and Vistas, the images are annotated independently by different annotators. ADE20k is annotated by a single well-trained annotator who labeled the same set of images with a gap of six months. To measure panoptic quality (PQ) for human annotators, we treat one annotation for each image as ground truth and the other as the prediction. Note that the PQ is symmetric w.r.t. the ground truth and prediction, so order is unimportant.

Human performance. First, Table 1 shows human performance on each dataset, along with the decomposition of PQ into segmentation quality (SQ) and recognition quality (RQ). As expected, humans are not perfect at this task, which is consistent with studies of annotation quality from [6, 49, 31]. Visualizations of human segmentation and classification errors are shown in Figures 3 and 4, respectively.

We note that Table 1 establishes a measure of annotator agreement on each dataset, *not* an upper bound on human performance. We further emphasize that numbers are not comparable across datasets and should not be used to assess dataset quality. The number of classes, percent of annotated pixels, and scene complexity vary across datasets, each of which significantly impacts annotation difficulty.

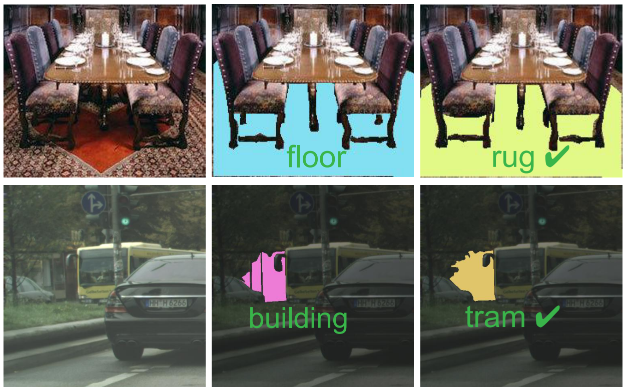

Figure 4: **Classification flaws.** Images are zoomed and cropped. Top row (ADE20k image): simple misclassification. Bottom row (Cityscapes image): the scene is extremely difficult, tram is the correct class for the segment. Many errors are difficult to resolve.

	PQ$^{\text{S}}$	PQ$^{\text{M}}$	PQ$^{\text{L}}$	SQ$^{\text{S}}$	SQ$^{\text{M}}$	SQ$^{\text{L}}$	RQ$^{\text{S}}$	RQ$^{\text{M}}$	RQ$^{\text{L}}$
Cityscapes	35.1	62.3	84.8	67.8	81.0	89.9	51.5	76.5	94.1
ADE20k	49.9	69.4	79.0	78.0	84.0	87.8	64.2	82.5	89.8
Vistas	35.6	47.7	69.4	70.1	76.6	83.1	51.5	62.3	82.6

Table 2: **Human performance *vs.* scale**, for small (S), medium (M) and large (L) objects. Scale plays a large role in determining human accuracy for panoptic segmentation. On large objects both SQ and RQ are above 80 on all datasets, while for small objects RQ drops precipitously. SQ for small objects is quite reasonable.

Stuff *vs.* things. PS requires segmentation of both stuff and things. In Table 1 we also show PQ$^{\text{St}}$ and PQ$^{\text{Th}}$ which is the PQ averaged over stuff classes and thing classes, respectively. For Cityscapes and ADE20k human performance for stuff and things are close, on Vistas the gap is a bit larger. Overall, this implies stuff and things have similar difficulty, although thing classes are somewhat harder. In Figure 5 we show PQ for every class in each dataset, sorted by PQ. Observe that stuff and things classes distribute fairly evenly. This implies that the proposed metric strikes a good balance and, indeed, is successful at unifying the stuff and things segmentation tasks without either dominating the error.

Small *vs.* large objects. To analyze how PQ varies with object size we partition the datasets into small (S), medium (M), and large (L) objects by considering the smallest 25%, middle 50%, and largest 25% of objects in each dataset, respectively. In Table 2, we see that for large objects human performance for all datasets is quite good. For small objects, RQ drops significantly implying human annotators often have a hard time finding small objects. However, if a small object is found, it is segmented relatively well.

IoU threshold. By enforcing an overlap greater than 0.5 IoU, we are given a unique matching by Theorem 1. However, is the 0.5 threshold reasonable? An alternate strategy is to use no threshold and perform the matching by solving

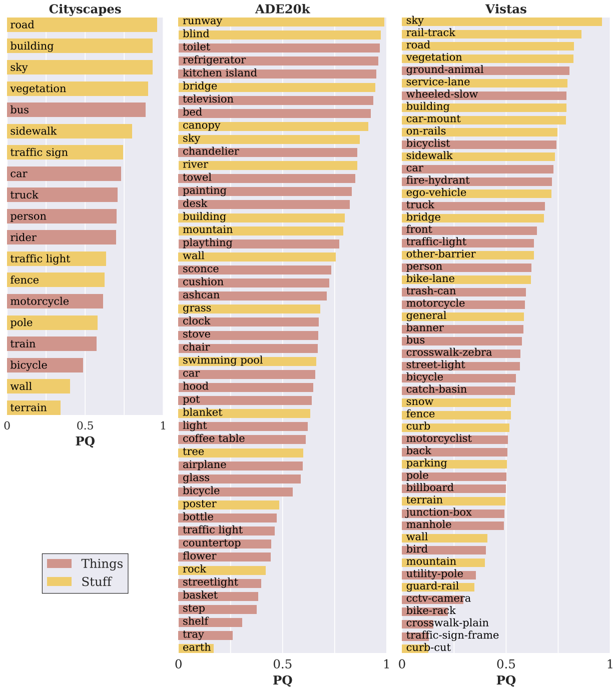

Figure 5: **Per-Class Human performance, sorted by PQ**. Thing classes are shown in red, stuff classes in orange (for ADE20k every other class is shown, classes without matches in the dual-annotated tests sets are omitted). Things and stuff are distributed fairly evenly, implying PQ balances their performance.

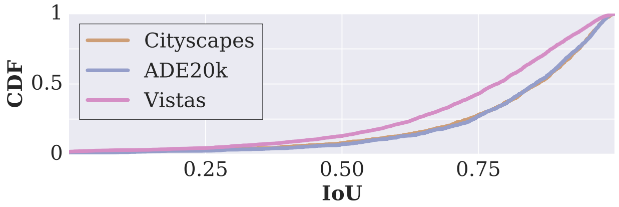

Figure 6: **Cumulative density functions of overlaps** for matched segments in three datasets when matches are computed by solving a maximum weighted bipartite matching problem [43]. After matching, less than 16% of matched objects have IoU below 0.5.

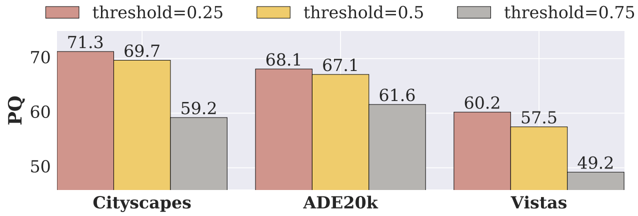

Figure 7: **Human performance for different IoU thresholds.** The difference in PQ using a matching threshold of 0.25 *vs*. 0.5 is relatively small. For IoU of 0.25 matching is obtained by solving a maximum weighted bipartite matching problem. For a threshold greater than 0.5 the matching is unique and much easier to obtain.

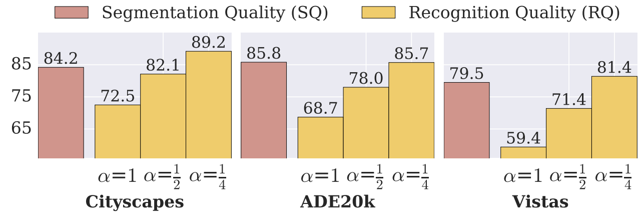

Figure 8: **SQ *vs*. RQ** for different α, see (3). Lowering α reduces the penalty of unmatched segments and thus increases the reported RQ (SQ is not affected). We use α of 0.5 throughout but by tuning α one can balance the influence of SQ and RQ in the final metric.

a maximum weighted bipartite matching problem [43]. The optimization will return a matching that maximizes the sum of IoUs of the matched segments. We perform the matching using this optimization and plot the cumulative density functions of the match overlaps in Figure 6. Less than 16% of the matches have IoU overlap less than 0.5, indicating that relaxing the threshold should have minor effect.

To verify this intuition, in Figure 7 we show PQ computed for different IoU thresholds. Notably, the difference in PQ for IoU of 0.25 and 0.5 is relatively small, especially compared to the gap between IoU of 0.5 and 0.75, where the change in PQ is larger. Furthermore, many matches at lower IoU are false matches. Therefore, given that the matching for IoU of 0.5 is not only unique, but also simple and intuitive, we believe that the default choice of 0.5 is reasonable.

SQ *vs*. RQ balance. Our RQ definition is equivalent to the F_1 score. However, other choices are possible. Inspired by the generalized F_β score [41], we can introduce a parameter α that enables tuning the penalty for recognition errors:

$$\mathrm{RQ}^{\alpha} = \frac{|TP|}{|TP| + \alpha|FP| + \alpha|FN|}\,. \tag{3}$$

By default α is 0.5. Lowering α reduces the penalty of unmatched segments and thus increases RQ (SQ is not affected). Since PQ=SQ×RQ, this changes the relative effect of PS *vs*. RQ on the final PQ metric. In Figure 8 we show SQ and RQ for various α. The default α strikes a good balance between SQ and RQ. In principle, altering α can be used to balance the influence of segmentation and recognition errors on the final metric. In a similar spirit, one could also add a parameter β to balance influence of FPs *vs*. FNs.

7. Machine Performance Baselines

We now present simple machine baselines for panoptic segmentation. We are interested in three questions: (1) How do heuristic combinations of top-performing instance and semantic segmentation systems perform on panoptic segmentation? (2) How does PQ compare to existing metrics like AP and IoU? (3) How do the machine results compare to the human results that we presented previously?

Cityscapes	AP	AP^{NO}	PQ^{Th}	SQ^{Th}	RQ^{Th}
Mask R-CNN+COCO [14]	**36.4**	**33.1**	**54.0**	**79.4**	**67.8**
Mask R-CNN [14]	31.5	28.0	49.6	78.7	63.0
ADE20k	AP	AP^{NO}	PQ^{Th}	SQ^{Th}	RQ^{Th}
Megvii [27]	**30.1**	**24.8**	**41.1**	**81.6**	**49.6**
G-RMI [10]	24.6	20.6	35.3	79.3	43.2

Table 3: **Machine results on instance segmentation** (stuff classes ignored). Non-overlapping predictions are obtained using the proposed heuristic. AP^{NO} is AP of the non-overlapping predictions. As expected, removing overlaps harms AP as detectors benefit from predicting multiple overlapping hypotheses. Methods with better AP also have better AP^{NO} and likewise improved PQ.

Cityscapes	IoU	PQ^{St}	SQ^{St}	RQ^{St}
PSPNet multi-scale [48]	**80.6**	**66.6**	**82.2**	**79.3**
PSPNet single-scale [48]	79.6	65.2	81.6	78.0
ADE20k	IoU	PQ^{St}	SQ^{St}	RQ^{St}
CASIA_IVA_JD [12]	**32.3**	**27.4**	**61.9**	**33.7**
G-RMI [11]	30.6	19.3	58.7	24.3

Table 4: **Machine results on semantic segmentation** (thing classes ignored). Methods with better mean IoU also show better PQ results. Note that G-RMI has quite low PQ. We found this is because it hallucinates many small patches of classes not present in an image. While this only slightly affects IoU which counts *pixel* errors it severely degrades PQ which counts *instance* errors.

Algorithms and data. We want to understand panoptic segmentation in terms of existing well-established methods. Therefore, we create a basic PS system by applying reasonable heuristics (described shortly) to the output of existing top instance and semantic segmentation systems.

We obtained algorithm output for three datasets. For *Cityscapes*, we use the val set output generated by the current leading algorithms (PSPNet [48] and Mask R-CNN [14] for semantic and instance segmentation, respectively). For *ADE20k*, we received output for the winners of both the semantic [12, 11] and instance [27, 10] segmentation tracks on a 1k subset of test images from the 2017 Places Challenge. For *Vistas*, which is used for the LSUN'17 Segmentation Challenge, the organizers provide us with 1k test images and results from the winning entries for the instance and semantic segmentation tracks [25, 47].

Using this data, we start by analyzing PQ for the instance and semantic segmentation tasks separately, and then examine the full panoptic segmentation task. Note that our 'baselines' are very powerful and that simpler baselines may be more reasonable for fair comparison in papers on PS.

Instance segmentation. Instance segmentation algorithms produce overlapping segments. To measure PQ, we must first resolve these overlaps. To do so we develop a simple non-maximum suppression (NMS)-like procedure. We first sort the predicted segments by their confidence scores and remove instances with low scores. Then, we iterate over sorted instances, starting from the most confident. For each instance we first remove pixels which have been assigned to previous segments, then, if a sufficient fraction of the segment remains, we accept the non-overlapping portion, otherwise we discard the entire segment. All thresholds are selected by grid search to optimize PQ. Results on Cityscapes and ADE20k are shown in Table 3 (Vistas is omitted as it only had one entry to the 2017 instance challenge). Most importantly, AP and PQ track closely, and we expect improvements in a detector's AP will also improve its PQ.

Semantic segmentation. Semantic segmentations have no overlapping segments by design, and therefore we can directly compute PQ. In Table 4 we compare mean IoU, a standard metric for this task, to PQ. For Cityscapes, the PQ gap between methods corresponds to the IoU gap. For ADE20k, the gap is much larger. This is because whereas IoU counts correctly predicted pixel, PQ operates at the level of instances. See the Table 4 caption for details.

Panoptic segmentation. To produce algorithm outputs for PS, we start from the non-overlapping instance segments from the NMS-like procedure described previously. Then, we combine those segments with semantic segmentation results by resolving any overlap between thing and stuff classes in favor of the thing class (*i.e.*, a pixel with a thing and stuff label is assigned the thing label and its instance id). This heuristic is imperfect but sufficient as a baseline.

Table 5 compares PQ^{St} and PQ^{Th} computed on the combined ('panoptic') results to the performance achieved from the separate predictions discussed above. For these results we use the winning entries from each respective competition for both the instance and semantic tasks. Since overlaps are resolved in favor of things, PQ^{Th} is constant while PQ^{St} is slightly lower for the panoptic predictions. Visualizations of panoptic outputs are shown in Figure 9.

Human *vs*. machine panoptic segmentation. To compare human *vs*. machine PQ, we use the machine panoptic predictions described above. For human results, we use the dual-annotated images described in §6 and use bootstrapping to obtain confidence intervals since these image sets are small. These comparisons are imperfect as they use different test images and are averaged over different classes (some classes without matches in the dual-annotated tests sets are omitted), but they can still give some useful signal.

We present the comparison in Table 6. For SQ, machines trail humans only slightly. On the other hand, machine RQ is dramatically lower than human RQ, especially on ADE20k and Vistas. This implies that recognition, *i.e.*, classification, is the main challenge for current methods. Overall, there is a significant gap between human and machine performance. We hope that this gap will inspire future research for the proposed panoptic segmentation task.

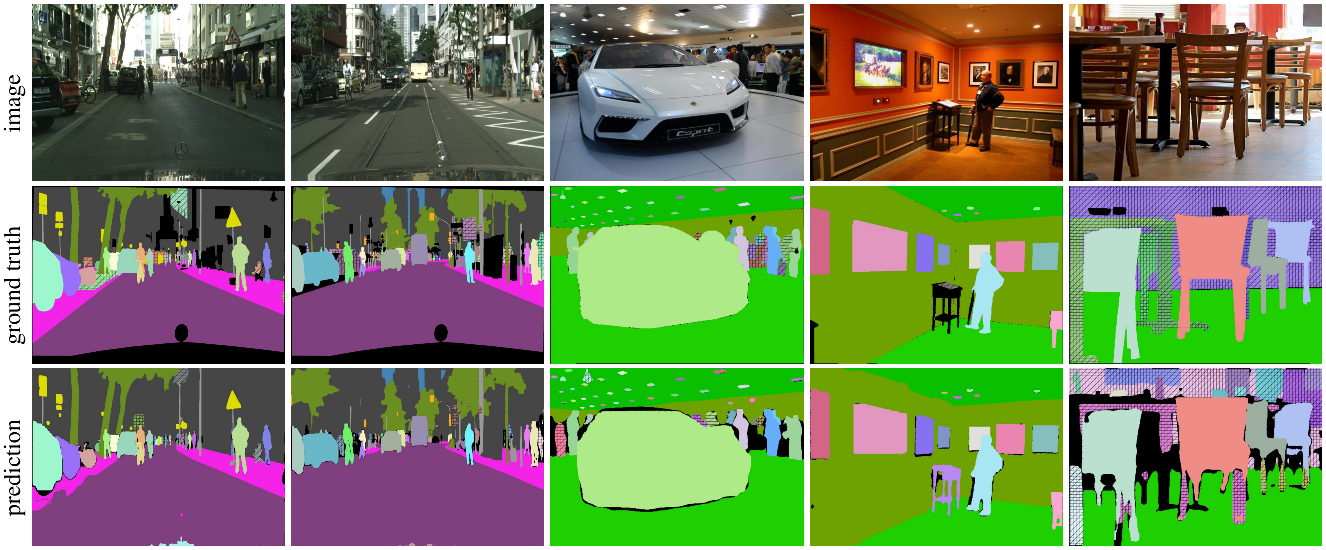

Figure 9: **Panoptic segmentation results** on Cityscapes (left two) and ADE20k (right three). Predictions are based on the merged outputs of state-of-the-art instance and semantic segmentation algorithms (see Tables 3 and 4). Colors for matched segments (IoU>0.5) match (crosshatch pattern indicates unmatched regions and black indicates unlabeled regions). Best viewed in color and with zoom.

Cityscapes	PQ	PQ^{St}	PQ^{Th}
machine-separate	n/a	66.6	54.0
machine-panoptic	61.2	66.4	54.0
ADE20k	PQ	PQ^{St}	PQ^{Th}
machine-separate	n/a	27.4	41.1
machine-panoptic	35.6	24.5	41.1
Vistas	PQ	PQ^{St}	PQ^{Th}
machine-separate	n/a	43.7	35.7
machine-panoptic	38.3	41.8	35.7

Table 5: **Panoptic *vs*. independent predictions.** The 'machine-separate' rows show PQ of semantic and instance segmentation methods computed independently (see also Tables 3 and 4). For 'machine-panoptic', we merge the non-overlapping thing and stuff predictions obtained from state-of-the-art methods into a true panoptic segmentation of the image. Due to the merging heuristic used, PQ^{Th} stays the same while PQ^{St} is slightly degraded.

Cityscapes	PQ	SQ	RQ	PQ^{St}	PQ^{Th}
human	$69.6^{+2.5}_{-2.7}$	$84.1^{+0.8}_{-0.8}$	$82.0^{+2.7}_{-2.9}$	$71.2^{+2.3}_{-2.5}$	$67.4^{+4.6}_{-4.9}$
machine	61.2	80.9	74.4	66.4	54.0
ADE20k	PQ	SQ	RQ	PQ^{St}	PQ^{Th}
human	$67.6^{+2.0}_{-2.0}$	$85.7^{+0.6}_{-0.6}$	$78.6^{+2.1}_{-2.1}$	$71.0^{+3.7}_{-3.2}$	$66.4^{+2.3}_{-2.4}$
machine	35.6	74.4	43.2	24.5	41.1
Vistas	PQ	SQ	RQ	PQ^{St}	PQ^{Th}
human	$57.7^{+1.9}_{-2.0}$	$79.7^{+0.8}_{-0.7}$	$71.6^{+2.2}_{-2.3}$	$62.7^{+2.8}_{-2.8}$	$53.6^{+2.7}_{-2.8}$
machine	38.3	73.6	47.7	41.8	35.7

Table 6: **Human *vs*. machine performance.** On each of the considered datasets human performance is much higher than machine performance (approximate comparison, see text for details). This is especially true for RQ, while SQ is closer. The gap is largest on ADE20k and smallest on Cityscapes. Note that as only a small set of human annotations is available, we use bootstrapping and show the the 5th and 95th percentiles error ranges for human results.

8. Future of Panoptic Segmentation

Our goal is to drive research in novel directions by inviting the community to explore the new panoptic segmentation task. We believe that the proposed task can lead to expected and unexpected innovations. We conclude by discussing some of these possibilities and our future plans.

Motivated by simplicity, the PS 'algorithm' in this paper is based on the *heuristic* combination of outputs from top-performing instance and semantic segmentation systems. This approach is a basic first step, but we expect more interesting algorithms to be introduced. Specifically, we hope to see PS drive innovation in at least two areas: (1) Deeply integrated end-to-end models that simultaneously address the dual stuff-and-thing nature of PS. A number of instance segmentation approaches including [24, 2, 3, 18] are designed to produce non-overlapping instance predictions and could serve as the foundation of such a system. (2) Since a PS cannot have overlapping segments, some form of higher-level 'reasoning' may be beneficial, for example, based on extending learnable NMS [7, 16, 17] to PS. We hope that the panoptic segmentation task will invigorate research in these areas leading to exciting new breakthroughs in vision.

Finally, we are working with competition organizers to extend popular segmentation datasets to include a panoptic segmentation track. Currently the COCO [22], Vistas [31], and ADE20k [49] challenges are considering featuring a panoptic segmentation track in 2018. We hope this will lead to a broad adoption of the proposed joint task.

References

[1] E. H. Adelson. On seeing stuff: the perception of materials by humans and machines. In *Human Vision and Electronic Imaging*, 2001. 1

[2] A. Arnab and P. H. Torr. Pixelwise instance segmentation with a dynamically instantiated network. In *CVPR*, 2017. 2, 3, 9

[3] M. Bai and R. Urtasun. Deep watershed transform for instance segmentation. In *CVPR*, 2017. 3, 9

[4] H. Caesar, J. Uijlings, and V. Ferrari. COCO-Stuff: Thing and stuff classes in context. In *CVPR*, 2018. 2, 5

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*, 2016. 1

[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2, 3, 5, 6

[7] C. Desai, D. Ramanan, and C. C. Fowlkes. Discriminative models for multi-class object layout. *IJCV*, 2011. 9

[8] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *PAMI*, 2012. 2

[9] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes challenge: A retrospective. *IJCV*, 2015. 2, 3

[10] A. Fathi, N. Kanazawa, and K. Murphy. Places challenge 2017: instance segmentation, G-RMI team. 2017. 8

[11] A. Fathi, K. Yang, and K. Murphy. Places challenge 2017: scene parsing, G-RMI team. 2017. 8

[12] J. Fu, J. Liu, L. Guo, H. Tian, F. Liu, H. Lu, Y. Li, Y. Bao, and W. Yan. Places challenge 2017: scene parsing, CASIA_IVA_JD team. 2017. 8

[13] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, 2014. 4, 5

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017. 1, 2, 8

[15] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? *PAMI*, 2015. 1

[16] J. Hosang, R. Benenson, and B. Schiele. Learning non-maximum suppression. *PAMI*, 2017. 9

[17] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation networks for object detection. In *CVPR*, 2018. 9

[18] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother. InstanceCut: from edges to instances with multicut. In *CVPR*, 2017. 2, 3, 9

[19] I. Kokkinos. UberNet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, 2017. 3

[20] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[21] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, 2017. 2

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 2, 5, 9

[23] C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. *PAMI*, 2011. 2, 3

[24] S. Liu, J. Jia, S. Fidler, and R. Urtasun. SGN: Sequential grouping networks for instance segmentation. In *CVPR*, 2017. 2, 3, 9

[25] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. LSUN'17: insatnce segmentation task, UCenter winner team. 2017. 8

[26] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1, 3, 5

[27] R. Luo, B. Jiang, T. Xiao, C. Peng, Y. Jiang, Z. Li, X. Zhang, G. Yu, Y. Mu, and J. Sun. Places challenge 2017: instance segmentation, Megvii (Face++) team. 2017. 8

[28] J. Malik, P. Arbeláez, J. Carreira, K. Fragkiadaki, R. Girshick, G. Gkioxari, S. Gupta, B. Hariharan, A. Kar, and S. Tulsiani. The three R's of computer vision: Recognition, reconstruction and reorganization. *PRL*, 2016. 3

[29] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 2004. 5

[30] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, 2016. 3

[31] G. Neuhold, T. Ollmann, S. Rota Bulò, and P. Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *CVPR*, 2017. 2, 3, 5, 6, 9

[32] P. O. Pinheiro, R. Collobert, and P. Dollár. Learning to segment object candidates. In *NIPS*, 2015. 2

[33] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 2

[35] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recog. and segm. In *ECCV*, 2006. 2

[36] M. Sun, B. Kim, P. Kohli, and S. Savarese. Relating things and stuff via object property interactions. *PAMI*, 2014. 3, 4

[37] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *CVPR*, 2013. 3, 4

[38] J. Tighe, M. Niethammer, and S. Lazebnik. Scene parsing with object instances and occlusion ordering. In *CVPR*, 2014. 2, 3, 4

[39] Z. Tu, X. Chen, A. L. Yuille, and S.-C. Zhu. Image parsing: Unifying segmentation, detection, and recognition. *IJCV*, 2005. 2, 3

[40] R. Vaillant, C. Monrocq, and Y. LeCun. Original approach for the localisation of objects in images. *IEE Proc. on Vision, Image, and Signal Processing*, 1994. 2

[41] C. Van Rijsbergen. *Information retrieval*. London: Butterworths, 1979. 5, 7

[42] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001. 2

[43] D. B. West. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001. 7

[44] Y. Yang, S. Hallman, D. Ramanan, and C. C. Fowlkes. Layered object models for image segmentation. *PAMI*, 2012. 4

[45] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*, 2012. 2, 3, 4

[46] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 1

[47] Y. Zhang, H. Zhao, and J. Shi. LSUN'17: semantic segmentation task, PSPNet winner team. 2017. 8

[48] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, 2017. 8

[49] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ADE20K dataset. In *CVPR*, 2017. 2, 3, 5, 6, 9

[50] Y. Zhu, Y. Tian, D. Mexatas, and P. Dollár. Semantic amodal segmentation. In *CVPR*, 2017. 3