COCO-Stuff: Thing and Stuff Classes in Context

Holger Caesar[1] Jasper Uijlings[2] Vittorio Ferrari[1 2]
University of Edinburgh[1] Google AI Perception[2]

Abstract

Semantic classes can be either things (objects with a well-defined shape, e.g. car, person) or stuff (amorphous background regions, e.g. grass, sky). While lots of classification and detection works focus on thing classes, less attention has been given to stuff classes. Nonetheless, stuff classes are important as they allow to explain important aspects of an image, including (1) scene type; (2) which thing classes are likely to be present and their location (through contextual reasoning); (3) physical attributes, material types and geometric properties of the scene. To understand stuff and things in context we introduce COCO-Stuff[1], which augments all 164K images of the COCO 2017 dataset with pixel-wise annotations for 91 stuff classes. We introduce an efficient stuff annotation protocol based on superpixels, which leverages the original thing annotations. We quantify the speed versus quality trade-off of our protocol and explore the relation between annotation time and boundary complexity. Furthermore, we use COCO-Stuff to analyze: (a) the importance of stuff and thing classes in terms of their surface cover and how frequently they are mentioned in image captions; (b) the spatial relations between stuff and things, highlighting the rich contextual relations that make our dataset unique; (c) the performance of a modern semantic segmentation method on stuff and thing classes, and whether stuff is easier to segment than things.

1. Introduction

Most of the recent object detection efforts have focused on recognizing and localizing thing classes, such as *cat* and *car*. Such classes have a specific size [21, 27] and shape [21, 51, 55, 39, 17, 14], and identifiable parts (e.g. a *car* has *wheels*). Indeed, the main recognition challenges [18, 43, 35] are all about things. In contrast, much less attention has been given to stuff classes, such as *grass* and *sky*, which are amorphous and have no distinct parts (e.g. a piece of *grass* is still *grass*). In this paper we ask: Is this strong focus on things justified?

To appreciate the importance of stuff, consider that it makes up the majority of our visual surroundings. For example, *sky*, *walls* and most *ground* types are stuff. Furthermore, stuff often determines the type of a scene, so it can be very descriptive for an image (e.g. in a beach scene the *beach* and *water* are the essential elements, more so than *people* and *volleyball*). Stuff is also crucial for reasoning about things: Stuff captures the 3D layout of the scene and therefore heavily constrains the possible locations of things. The contact points between stuff and things are critical for determining depth ordering and relative positions of things, which supports understanding the relations between them. Finally, stuff provides context helping to recognize small or uncommon things, e.g. a metal thing in the sky is likely an *aeroplane*, while a metal thing in the water is likely a *boat*. For all these reasons, stuff plays an important role in scene understanding and we feel it deserves more attention.

Figure 1: *(left) An example image, (middle) its thing annotations in COCO [35] and (right) enriched stuff and thing annotations in COCO-Stuff. Just having the* train, person, bench *and* potted plant *does not tell us much about the context of the scene, but with stuff and thing labels we can infer the position and orientation of the* train*, its stuff-thing interactions (*train *leaving the* station*) and thing-thing interactions (*person *waiting for a different* train*). This is also visible in the captions written by humans. Whereas the captions only mention one thing (*train*), they describe a multitude of different stuff classes (*track, station, bridge, tower, railway*), stuff-thing interactions (*train *leaving the* station, train *crossing beneath a city* bridge*) and spatial arrangements (on, beneath).*

In this paper we introduce the COCO-Stuff dataset, which augments the popular COCO [35] with pixel-wise annotations for a rich and diverse set of 91 stuff classes. The original COCO dataset already provides outline-level annotation for 80 thing classes. The additional stuff annotations enable the study of stuff-thing interactions in the complex COCO images. To illustrate the added value of our stuff an-

[1] http://calvin.inf.ed.ac.uk/datasets/coco-stuff

notations, Fig. 1 shows an example image, its annotations in COCO and COCO-Stuff. The original COCO dataset offers location annotations only for the *train*, *potted plant*, *bench* and *person*, which are not sufficient to understand what the scene is about. Indeed, the image captions written by humans (also provided by COCO) mention the *train*, its interaction with stuff (i.e. *track*), and the spatial arrangements of the *train* and its surrounding stuff. All these elements are necessary for scene understanding and show how COCO-Stuff offers much more comprehensive annotations.

This paper makes the following contributions: (1) We introduce COCO-Stuff, which augments the original COCO dataset with stuff annotations. (2) We introduce an annotation protocol for COCO-Stuff which leverages the existing thing annotations and superpixels. We demonstrate both the quality and efficiency of this protocol (Sec. 3). (3) Using COCO-Stuff, we analyze the role of stuff from multiple angles (Sec. 4): (a) the importance of stuff and thing classes in terms of their surface cover and how frequently they are mentioned in image captions; (b) the spatial relations between stuff and things, highlighting the rich contextual relations that make COCO-Stuff unique; (c) we compare the performance of a modern semantic segmentation method on thing and stuff classes.

Hoping to further promote research on stuff and stuff-thing contextual relations, we release COCO-Stuff and the trained segmentation models online[1].

2. Related Work

Defining things and stuff. The literature provides definitions for several aspects of stuff and things, including: (1) Shape: Things have characteristic shapes (*car*, *cat*, *phone*), whereas stuff is amorphous (*sky*, *grass*, *water*) [21, 59, 28, 51, 55, 39, 17, 14]. (2) Size: Things occur at characteristic sizes with little variance, whereas stuff regions are highly variable in size [21, 2, 27]. (3) Parts: Thing classes have identifiable parts [56, 19], whereas stuff classes do not (e.g. a *piece of grass* is still *grass*, but a *wheel* is not a *car*). (4) Instances: Stuff classes are typically not countable [2] and have no clearly defined instances [14, 25, 53]. (5) Texture: Stuff classes are typically highly textured [21, 27, 51, 14]. Finally, a few classes can be interpreted as both stuff and things, depending on the image conditions (e.g. a large number of *people* is sometimes considered a *crowd*).

Several works have shown that different techniques are required for the detection of stuff and things [51, 53, 31, 14]. Moreover, several works have shown that stuff is a useful contextual cue to detect things and vice versa [41, 27, 31, 38, 45].

Stuff-only datasets. Early stuff datasets [6, 15, 34, 9] focused on texture classification and had simple images completely covered with a single textured patch. The more recent Describable Textures Dataset [12] instead collects textured patches in the wild, described by human-centric attributes. A related task is material recognition [44, 4, 5]. Although the recent Materials in Context dataset [5] features realistic and difficult images, they are mostly restricted to indoor scenes with man-made materials. For the task of semantic segmentation, the Stanford Background dataset [24] offers pixel-level annotations for seven common stuff categories and a single *foreground* category (confounding all thing classes). All stuff-only datasets above have no distinct thing classes, which make them inadequate to study the relations between stuff an thing classes.

Dataset	Images	Classes	Stuff classes	Thing classes	Year
MSRC 21 [46]	591	21	6	15	2006
KITTI [23]	203	14	9	4	2012
CamVid [7]	700	32	13	15	2008
Cityscapes [13]	25,000	30	13	14	2016
SIFT Flow [36]	2,688	33	15	18	2009
Barcelona [50]	15,150	170	31	139	2010
LM+SUN [52]	45,676	232	52	180	2010
PASCAL Context [38]	10,103	540	152	388	2014
NYUD [47]	1,449	894	190	695	2012
ADE20K [63]	25,210	2,693	1,242	1,451	2017
COCO-Stuff	163,957	172	91	80	2018

Table 1: *An overview of datasets with pixel-level stuff and thing annotations. COCO-Stuff is the largest existing dataset with dense stuff and thing annotations. The number of stuff and thing classes are estimated given the definitions in Sec. 2. Sec. 3.3 shows that COCO-Stuff also has more* usable *classes than any other dataset.*

Thing-only datasets. These datasets have bounding box or outline-level annotations of things, e.g. PASCAL VOC [18], ILSVRC [43], COCO [35]. They have pushed the state-of-the-art in Computer Vision, but the lack of stuff annotations limits the ability to understand the whole scene.

Stuff and thing datasets. Some datasets have pixel-wise stuff and thing annotations (Table 1). Early datasets like MSRC 21 [46], NYUD [47], CamVid [7] and SIFT Flow [36] annotate less than 50 classes on less than 5,000 images. More recent large-scale datasets like Barcelona [50], LM+SUN [52], PASCAL Context [38], Cityscapes [13] and ADE20K [63] annotate tens of thousands of images with hundreds of classes. We compare COCO-Stuff to these datasets in Sec. 3.3.

Annotating datasets. Dense pixel-wise annotation of images is extremely costly. Several works use interactive segmentation methods [42, 57, 10] to speedup annotation; others annotate superpixels [61, 22, 40]. Some works operate in a weakly supervised scenario, deriving full image annotations starting from manually annotated squiggles [3, 60] or points [3, 30]. These approaches take less time, but typically lead to lower quality.

In this work we introduce a new annotation protocol to obtain high quality pixel-wise stuff annotations at low human costs by using superpixels and by exploiting the existing detailed thing annotations of COCO [35] (Sec. 3.2).

3. The COCO-Stuff dataset

The Common Objects in COntext (COCO) [35] dataset is a large-scale dataset of images of high complexity. COCO has been designed to enable the study of thing-thing interactions, and features images of complex scenes with many small objects, annotated with very detailed outlines. However, COCO is missing stuff annotations. In this paper we augment COCO by adding dense pixel-wise stuff annotations. Since COCO is about complex, yet natural scenes containing substantial areas of stuff, COCO-Stuff enables the exploration of rich relations between things and stuff. Therefore COCO-Stuff offers a valuable stepping stone towards complete scene understanding.

Fig. 2 presents several annotated images from the COCO-Stuff dataset, showcasing the complexity of the images, the large number and diversity of stuff classes, the high level of accuracy of the annotations, and the completeness in terms of surface coverage of the annotations. We have annotated all 164K images in COCO 2017: training (118K), val (5K), test-dev (20K) and test-challenge (20K).

3.1. Defining stuff labels.

COCO-Stuff contains 172 classes: 80 thing, 91 stuff, and 1 class *unlabeled*. The 80 thing classes are the same as in COCO [35]. The 91 stuff classes are curated by an expert annotator. The class *unlabeled* is used in two situations: if a label does not belong to any of the 171 predefined classes, or if the annotator cannot infer the label of a pixel.

Before annotation, we choose to predefine our label set. This contrasts with a common choice in semantic segmentation to have annotators use free-form text labels [50, 52, 38]. However, using free-form labels leads to several problems. First of all, it leads to an extremely large number of classes, many having only a handful of examples. This makes most classes unusable for recognition purposes, as observed in [38, 63]. Furthermore, different annotators typically use several synonyms to indicate the same class. These need to be merged a posteriori [50, 58]. Even after merging, classes might not be consistently annotated. For example, PASCAL Context [38] includes the classes *bridge* and *footbridge*, which are in a parent-child relationship. If one image has *bridge* annotations and another image has *footbridge* annotations, both can describe the same concept (i.e. *footbridge*), or the *bridge* can be another type of *bridge* and therefore describe a different concept. Similarly, in SIFT Flow [36] some images have *field* annotations, whereas others have *grass* annotations. These concepts are semantically overlapping, but are neither synonymous nor in a parent-child relationship. A region with a *grass field* could be annotated as *grass* or as *field* depending on the annotator.

To prevent such inconsistencies, we decided to predefine a set of mutually exclusive stuff classes, similarly to how the COCO thing classes were defined. Additionally, we organized our classes into a label hierarchy, e.g. classes like *cloth* and *curtain* have *textile* as parent, while classes like *moss* and *tree* have *vegetation* as parent (Fig. 3). The super-categories *textile* and *vegetation* have *indoor* and *outdoor* as parents, respectively. The top-level nodes in our hierarchy are generic classes *stuff* and *thing*.

To choose our set of stuff labels, the expert annotator used the following criteria: stuff classes should (1) be mutually exclusive; (2) in their ensemble, cover the vast majority of the stuff surface appearing in the dataset; (3) be frequent enough; (4) have a good level of granularity, around the base level for a human. However, these criteria conflict with each other: if we label all vegetations as *vegetation*, the labels are too general. On the other extreme, if we create a separate class for every single type of *vegetation*, the labels are too specific and infrequent. Therefore, as shown in Fig. 3, for every super-category like *vegetation*, we explicitly list its most frequent subclasses as choices for the annotator to pick (e.g. *straw*, *moss*, *bush* and *grass*). And there is one additional subclass *vegetation-other* to be picked to label any other case of *vegetation*. This achieves the coverage goal, while avoiding to scatter the data over many small classes. For some super-categories (*floor*, *wall* and *ceiling*) we are particularly interested in the material they are made of. Therefore we include the material type in the class definition (e.g. *wall-brick*, *wall-concrete* and *wall-wood*). This enables further analysis of the materials present in a scene.

Our label set fulfills all design criteria (1-4): (1) the mutual exclusivity of labels is by design and enforced through having annotators only use the leaves of our hierarchy as labels (Fig. 3). For the other criteria we need to look at pixel-level frequencies after dataset collection: (2) only 6% of the pixels are *unlabeled*, which is satisfactory; (3, 4) interestingly, all our stuff classes have pixel frequencies in the same range of the COCO thing classes (Fig. 5) and they also follow a similar distribution and granularity (Fig. 3). Intuitively, having both thing and stuff classes follow similar distributions makes the dataset well suited to analyze stuff-thing relations.

3.2. Annotation protocol and analysis

Protocol. We developed a very efficient protocol, specialized for labeling stuff classes at the pixel-level. We first partition each image into 1,000 superpixels using SLICO [1], which adheres very well to boundaries and gives superpixels of homogeneous size (Fig. 4). Superpixels remove the need for manually delineating the exact boundaries between two regions of different classes. As superpixels respect boundaries, it is enough to mark which superpixels belong to which class, which is a lot faster to do. Moreover, the evenly spaced and sized SLICO superpixels result in a labeling task natural for humans (as opposed to superpixel algorithms which yield regions that greatly vary in size [20]).

Figure 2: *Annotated images from the COCO-Stuff dataset with dense pixel-level annotations for stuff and things. To emphasize the depth ordering of stuff and thing classes we use bright colors for thing classes and darker colors for stuff classes.*

We accelerate the annotation process by providing annotators a size-adjustable paintbrush tool, which enables labeling large regions of stuff very efficiently (Fig. 4b).

We improve annotation efficiency even further by leveraging the highly accurate thing outlines available from COCO [35] (Fig. 4c). We show annotators images with *thing overlays*, and pixels belonging to things are clamped and unaffected by the annotator's brush. This results in a lightweight experience, where the annotator merely needs to select a stuff class (like *snow*) and brush over the foreground object. In fact, because of the high annotation accuracy of COCO things, our technique results in extremely precise stuff outlines at stuff-thing boundaries, often beyond the accuracy of superpixel boundaries.

As a final element in our protocol, we present our stuff labels to the annotators using the full hierarchy. In initial trials we found that, compared to presenting them in a list, this reduces the look-up time of labels significantly. This annotation protocol yields an annotation time of only three minutes to annotate stuff in one of the COCO images, which are very complex (Fig. 2). We release the superpixels and the annotation tool online to allow for further analysis.

We annotated 10K images with our protocol using in-house annotators. Afterwards, we collaborated with the startup Mighty AI to adapt our protocol for crowdsourcing and annotate all remaining images of COCO-Stuff.

Analysis of superpixels. We study here the quality-speed trade-off of using superpixels. We ask a single annotator to annnotate 10 COCO images three times, once for each of three different modalities: (1) superpixel annotation, as we do for COCO-Stuff; (2) polygon annotation, the de facto standard [13, 38, 63] and (3) freedraw annotation, which consists of directly annotating pixels with a very accurate size-adjustable paintbrush tool, but *without* aid from superpixels. The freedraw annotations attempt to get as close to pixel-level accuracy as possible, and we use them as ground-truth reference in this analysis.

Table 2 shows the results for superpixel, polygon and freedraw annotation. Compared to the freedraw reference, polygons and superpixels are much faster (1.5x and 2.8x). Computing pixel-level labeling agreement w.r.t. freedraw reveals that both polygons and superpixels lead to very accurate annotations (96%-97%). We also asked the annotator to re-annotate the images with the same modality, enabling to measure 'self agreement'. Interestingly the self agreement of freedraw is in the same range as the agreement of superpixels and polygons w.r.t. freedraw. This shows that the differences across annotation modalities are of similar magnitude to the natural variations within a single modality, even by a single annotator. Hence, all three modalities are about as accurate.

Furthermore we simulate our stuff annotation protocol on two other datasets which were originally annotated with polygons: SIFT Flow [36] and PASCAL Context [38]. For each image we label each superpixel with the majority stuff label in the ground-truth annotations. We then overlay the existing thing annotations. This protocol achieves 98.3% agreement with the ground-truth on SIFT Flow and 98.4% on PASCAL Context. These findings show that superpixel annotation is faster than conventional polygon annotation, while providing almost the same annotations.

We found that a dominant factor for the differences in annotation time across images is their boundary complexity. Boundary complexity is defined as the ratio of pixels that have any neighboring pixel with a different semantic label (as in the boundary evaluation in [8, 32, 33]). Fig. 6 analyzes the relationship between boundary complexity and annotation time of an image using different annotation modalities. The linear trendlines show that there is a clear correlation between annotation time and boundary complexity. We

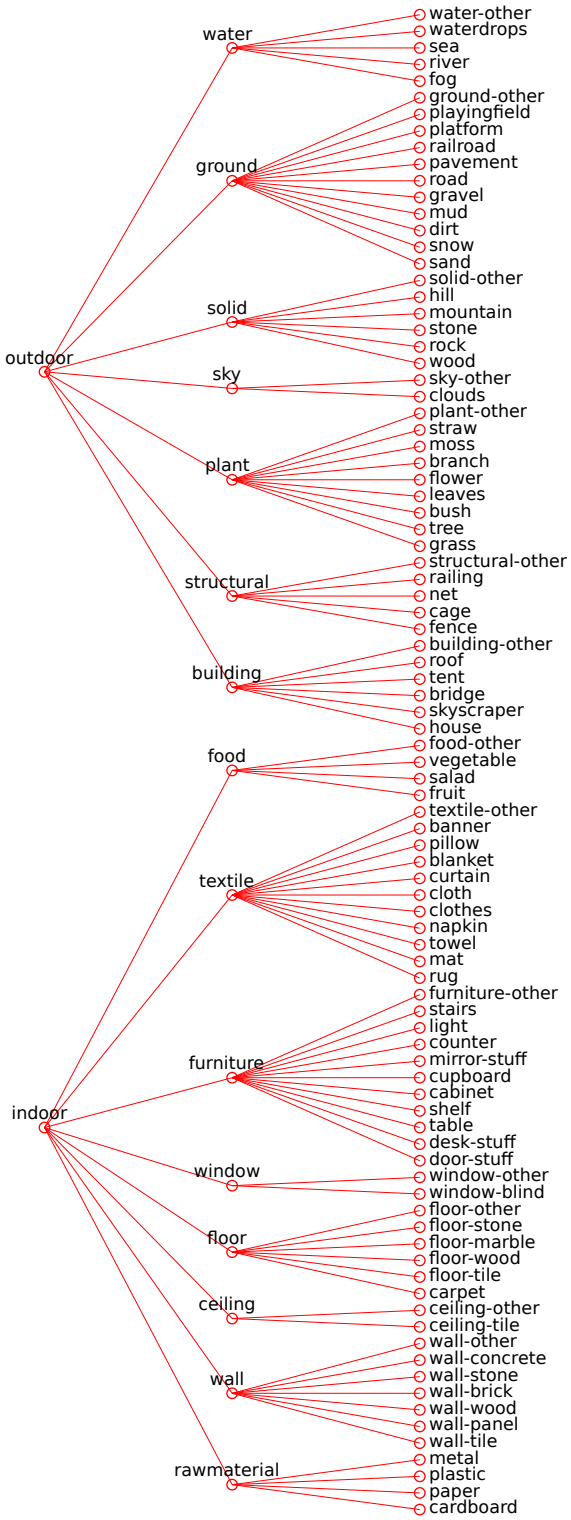

Figure 3: *The stuff label hierarchy of the COCO-Stuff dataset. Stuff classes are divided into* outdoor *and* indoor*, each further divided into super-categories (e.g.* floor, plant*), and finally into leaf-level classes (e.g.* marble floor, grass*). The labels used by the annotators form the leaf nodes of the tree. Furniture classes can be interpreted as either things or stuff, depending on the imaging conditions. A full list of descriptions is available online[1].*

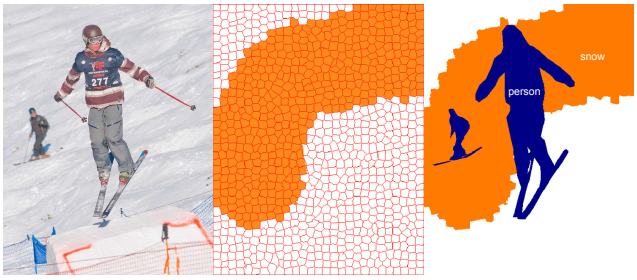

Figure 4: *Example of a) an image, b) the superpixel-based stuff annotation and c) the final labeling. The annotator can quickly annotate large stuff regions (*snow*) with a single mouse stroke using a paintbrush tool. Thing (*person*) annotations are copied from the COCO dataset. The transparency of each layer can be regulated to get a better overview. This approach dramatically reduces annotation time and yields a very accurate labeling, especially at stuff-thing boundaries.*

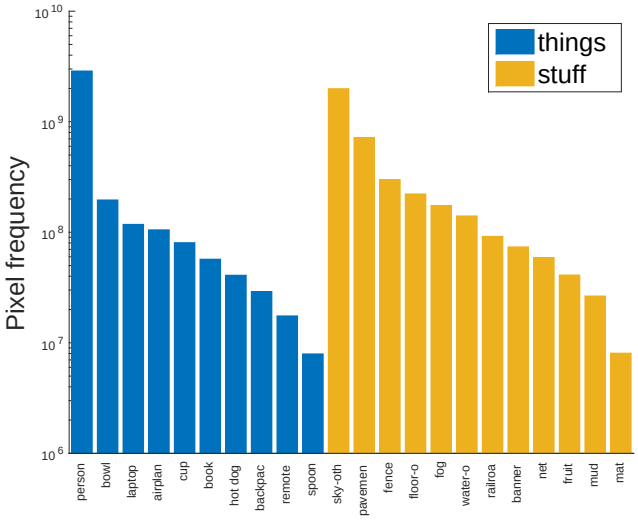

Figure 5: *Pixel-level frequencies for some of the classes in the trainval set of COCO-Stuff. For clarity, we show about* $1/8$ *of all classes. We can see that stuff and thing classes follow a similar pixel frequency distribution.*

Modality	Speedup	Reference agreement	Self agreement
Superpixels	2.8	96.1%	98.7%
Polygons	1.5	97.3%	97.0%
Freedraw	1.0	-	96.6%

Table 2: *A quantitative comparison of different stuff annotation modalities. We use freedraw annotation as a reference in the 'Speedup' and 'Reference agreement' columns. The self-agreement between repeated runs of the same annotation modality decreases with weaker constraints on the possible labelings.*

can see that the slopes of the freedraw and polygon annotation trendlines are 3.4x and 2.0x steeper than for superpixels. This is one of the main reasons why superpixels yield such big improvements in annotation time on average.

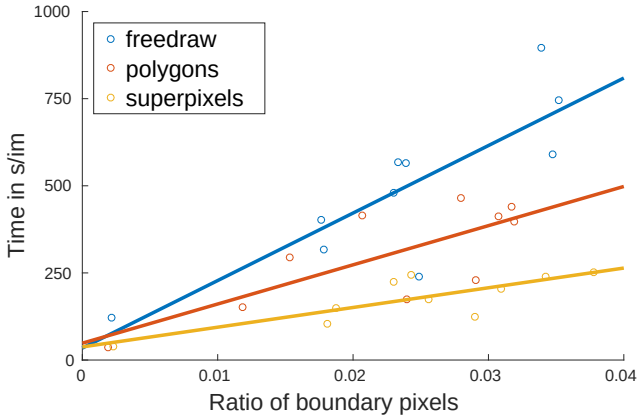

Figure 6: *Annotation time versus image boundary complexity. Each circle indicates an image annotated using one of three modalities. The trendlines show that annotation time for some modalities increases faster with boundary complexity than for others.*

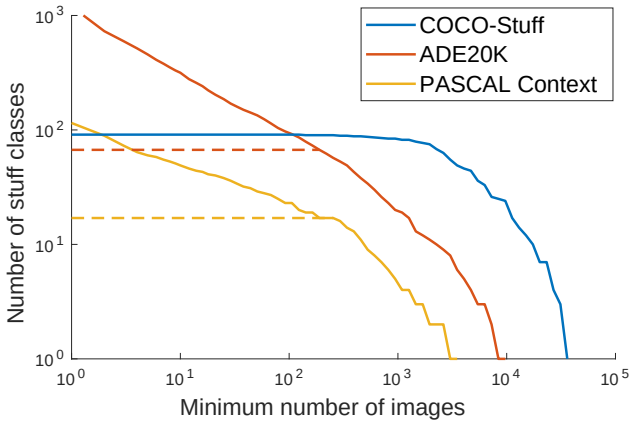

Figure 7: *The number of stuff classes occurring in at least x images for varying thresholds of x. Solid lines indicate the full datasets, dashed lines the versions with only usable classes. Statistics are computed on the trainval sets of three datasets.*

Analysis of thing overlays. We analyze thing overlays in terms of the annotation speedup they bring and the quality they lead to. For this we perform superpixel and freedraw annotation with and without thing overlays. We achieve significant speedups when using thing overlays with freedraw annotation (1.8x) and also with superpixel annotation (1.2x). Furthermore, the agreement of superpixel annotation w.r.t. the freedraw reference is identical with and without thing overlays (96.1% in both cases). This shows that thing overlays achieve a significant speedup without any loss in quality.

Moreover, 46.8% of the boundary pixels in COCO-Stuff have a neighboring pixel that belongs to a thing class. Therefore using thing overlays significantly decreases the boundary complexity and leads to a larger speedup for freedraw annotation than for superpixel annotation.

Across-annotator agreement. Following [63, 13] we annotate 30 images by 3 annotators each. For each image we compute the label agreement between each pair of annotators and average over all pairs. The mean label agreement in COCO-Stuff is 73.6%, compared to 66.8% for ADE20K [63].

3.3. Comparison to other datasets.

COCO-Stuff has the largest number of images of any semantic segmentation dataset (164K). In particular, MSRC 21 [46], KITTI [23], CamVid [7], SIFT Flow [36] and NYUD [47] all have less than 5,000 images (Table 1). COCO-Stuff is also much richer in both the number of stuff and thing classes than MSRC 21 [46], KITTI [23], CamVid [7], Cityscapes [13] and SIFT Flow [36]. Compared to the Barcelona [50] and LM+SUN [52] datasets, it has 3× and 2× more stuff classes, respectively.

PASCAL Context [38] and ADE20K [63] are the most similar datasets to COCO-Stuff. On the surface they appear to have a very large numbers of classes (540 and 2,693), but in practice most classes are rare. The authors of those datasets define a set of classes deemed *usable* for experiments (i.e. the most frequent 60 classes in PASCAL Context and 150 classes in ADE20K). In Fig. 7 we show the number of stuff classes that occur in at least x images, for varying thresholds x, on the trainval sets of three datasets. COCO-Stuff has more usable stuff classes than PASCAL Context and ADE20K for *any* threshold, e.g. for $x = 1,000$, there are 5 stuff classes in PASCAL Context, 20 in ADE20K and 84 in COCO-Stuff. This means that 92% of the stuff classes in COCO-Stuff occur in at least $1,000$ images. Furthermore, both PASCAL Context and ADE20K use free-form label names, which lead to annotations at different granularities and hence ambiguities, as discussed in Sec. 3.1. In contrast, in COCO-Stuff all labels are mutually exclusive and at a comparable level of granularity. Finally, PASCAL Context and ADE20K are annotated with overlapping polygons. Hence some pixels have multiple conflicting labels at the boundaries between things and stuff. In COCO-Stuff instead, each pixel has exactly one label.

To conclude, COCO-Stuff is a very large dataset of highly complex images. It has the largest number of usable stuff and thing classes with pixel-level annotations. Moreover, by building on COCO it also has natural language captions, further supporting rich scene understanding.

4. Analysis of stuff and things

In this section we leverage COCO-Stuff to analyze various relations between stuff and things: we analyze the relative importance of stuff and thing classes (Sec. 4.1); study spatial contextual relations between stuff and things (Sec. 4.2); and analyze the behavior of semantic segmentation methods on stuff and things (Sec. 4.3). To preserve the integrity of the test set annotations, all experiments in this section are run on the trainval set of COCO-Stuff.

Level	Stuff	Things
Pixels	**69.1%**	30.9%
Regions	**69.4%**	30.6%
Caption nouns	38.2%	**61.8%**

Table 3: *Relative frequency of stuff and thing classes in pixel-level annotations and caption nouns in COCO-Stuff.*

4.1. Importance of stuff and things

We quantify the relative importance of stuff and things using two criteria: surface cover and human descriptions.

Surface cover. We measure the frequencies of stuff and thing pixels in the COCO-Stuff annotations. Table 3 shows that the majority of pixels are stuff (69.1%). We also compute statistics for the labeled *regions* in COCO-Stuff, i.e. connected components in the pixel annotation map. We use such regions as a proxy for class instances, as stuff classes do not have instances. We see that 69.4% of the regions are stuff and 30.6% things.

Human descriptions. Although stuff classes cover the majority of the image surface, one might argue they are just irrelevant background pixels. The COCO dataset is annotated with five captions per image [35], which have been written explicitly to describe its content, and therefore capture the most relevant aspects of the image for a human. To emphasize the importance of stuff for scene understanding, we also analyze these captions, counting how many nouns point to things and stuff respectively. We use a Part-Of-Speech (POS) tagger [54] to automatically detect nouns. Then we manually categorize the 600 most frequent nouns as stuff (e.g. *street*, *field*, *water*, *building*, *beach*) or things (e.g. *man*, *dog*, *train*), ignoring nouns that do not represent physical entities (e.g. *game*, *view*, *day*).

Table 3 shows the relative frequency of these nouns. Stuff covers more than a third of the nouns (38.2%). This clearly shows the importance of stuff according to the COCO image captions.

4.2. Spatial context between stuff and things

Methodology. We analyze spatial context by considering the relative image position of one class with respect to another. For simplicity, here we explain how to compute the spatial context for one particular reference class, i.e. *car* (Fig. 8, second column). The explanation is analogous for all other classes. For every image containing a *car*, we extract a set of *car* regions, i.e. connected components of *car* pixels in the annotation map. Next we compute a histogram of image pixels surrounding the *car* regions, with two spatial dimensions (distance, angle) and one dimension for the class label. To determine in which spatial bin a certain pixel lands, we (1) compute the distance between the pixel and the nearest point in the *car* region (normalized by image size); (2) compute the relative angle with respect to the center of mass of the *car* region.

Results. Fig. 8 shows the spatial context of eight reference classes. This visualization reveals several interesting contextual relations. *Trains* are typically found above *railroads* (thing-stuff). *TVs* are typically found in front of *persons* (thing-thing). *Tiled walls* occur above *tiled floors* (stuff-stuff), and *roads* are flanked by *persons* on both sides (stuff-thing). Note that these contextual relations are not necessarily symmetric: most *cars* appear above a *road*, but many *roads* have other things above them.

For each reference class and spatial bin we also show the conditional probability of the most likely other class as a measure of confidence (Fig. 8, bottom). In most cases the highest confidence is in regions above (*sky*, *wall*, *ceiling*) or below (*road*, *pavement*, *snow*) the reference region, but rarely to the left or right. Since vertical relations are mostly support relations (e.g. 'on top of'), this suggests that support is the most informative type of context. For some classes the highest confidence region is also very close to the reference region, often indicating that another object is attached to the reference one (*person* close to *backpack*).

As the figure shows, some classes have a rich and diverse context, composed of many other classes (e.g. *tv*, *road*), while some classes have a simpler context (e.g. *snowboards* always appear in the middle of *snow*). We quantify the complexity of a reference class as the entropy of the conditional probability distribution, averaged over all other classes and spatial bins. The classes with highest mean entropy are *wood*, *metal* and *person*, and those with the lowest are *snowboard*, *airplane* and *playingfield*. On average, we find that stuff classes have a significantly higher mean entropy than things (3.40 vs. 3.02), showing they appear in more varied contexts. We also find that the mean entropy is rather constant over distances (small: 3.21, big: 3.23) and directions (left: 3.19, right: 3.18, down: 3.20, up: 3.15).

Comparing the mean entropy of different datasets, taking into account all classes, we find that COCO-Stuff has the highest (3.22), followed by the 60 *usable* classes of PASCAL Context (2.42), the 150 *usable* classes of ADE20K (2.18) and SIFT Flow (1.20). This shows the contextual richness of COCO-Stuff.

4.3. Semantic segmentation of stuff and things

We now analyze how a modern semantic segmentation method [11] performs on COCO-Stuff. We compare the performance on stuff and thing classes and hope to establish a baseline for future experiments on this dataset.

Protocol. We use the popular DeepLab V2 [11] based on the VGG-16 network [48] pre-trained on the ILSVRC classification dataset [43]. We use the following experimental protocol: train on the 118K training images and test on the 5K val images. To evaluate performance we use four criteria commonly used in the literature [37, 16, 8]: (1) *pixel accuracy* is the percentage of correctly labeled pixels in the

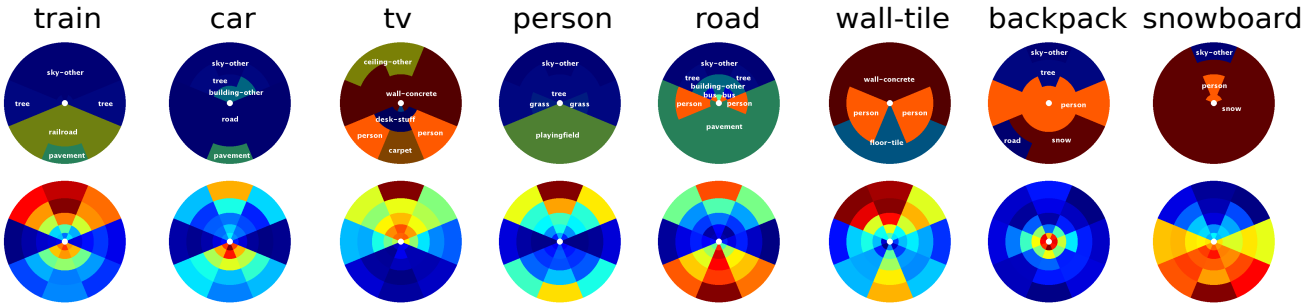

Figure 8: *Spatial context visualizations. (Top) Each disc is for a different reference class and shows the most likely other class at each direction and distance bin. (Bottom) The conditional probabilities of the most common class in each bin, as a measure of confidence. The values are normalized for each reference class and range from low (blue) to high (red). We also show examples for classes with high (person) and low (snowboard) mean entropy.*

dataset, (2) *class accuracy* computes the average of the per-class accuracies, (3) *mean Intersection-over-Union (IOU)* divides the number of pixels of the intersection of the predicted and ground-truth class by their union, averaged over classes [18], (4) *frequency weighted (FW) IOU* is per-class IOU weighted by the pixel-level frequency of each class.

Results for all images and classes. Table 4 shows the results using all images (row "118K (train)"). DeepLab achieves an mIOU of 33.2% over all classes. A detailed comparison of leading methods can be found online[1].

Benefits of a large dataset. One reason for the recent success of deep learning methods is the advent of large-scale datasets [43, 29, 63]. Inspired by [49], we want to test whether the performance of semantic segmentation models plateaus at current dataset sizes or whether it benefits from larger datasets. Following the above protocol, we train multiple DeepLab models with different amounts of training data, keeping all training parameters fixed. Table 4 shows the resulting performance on the validation set (rows from 1K to 118K). We can see that for all metrics, performance significantly increases as the training set grows. We hypothesize that even deeper network architectures [26] could benefit even more from large training sets.

Is stuff easier than things? Several works found that stuff is easier to segment than things [50, 28, 36, 51, 53, 62, 60, 63]. We argue that this is due to their choice of dataset, rather than a general observation. Most datasets only include a small number of very frequent and coarse-grained stuff classes, such as *sky* and *grass* (Table 1). In contrast, COCO-Stuff features a larger number of relevant stuff labels at a similar level of granularity as the existing thing labels. It has a similar number of stuff and thing classes, and a similar pixel frequency distribution for both (see Fig. 5).

As Table 4 (bottom) shows, on COCO-Stuff DeepLap performs substantially better on thing classes than on stuff. This shows that stuff is harder to segment than things in COCO-Stuff, a dataset where both stuff and things are similarly distributed. Therefore we argue that stuff is not generally easier than things.

Training images	Class accuracy	Pixel accuracy	Mean IOU	FW IOU
1K	24.1%	46.1%	15.9%	31.0%
5K	33.8%	52.7%	23.1%	37.5%
10K	36.9%	54.6%	25.5%	39.6%
20K	40.2%	57.5%	28.6%	42.6%
40K	43.0%	61.1%	31.4%	45.7%
80K	44.9%	63.4%	32.9%	47.4%
118K (train)	**45.1%**	**63.6%**	**33.2%**	**47.6%**
stuff	33.5%	58.2%	24.0%	45.6%
things	**58.3%**	**75.7%**	**43.6%**	**58.4%**

Table 4: *Rows 1K to 118K: Performance of Deeplab V2 with VGG-16 with varying amounts of training data. We can see that for all metrics, performance significantly increases for larger datasets. Last two rows: Performance of the same model on stuff and thing classes using all 118K training images in COCO.*

5. Conclusion

We introduced the large-scale COCO-Stuff dataset. COCO-Stuff enriches the COCO dataset with dense pixel-level stuff annotations. We used a specialized stuff annotation protocol to efficiently label each pixel. Our dataset features a diverse set of stuff classes. In combination with the existing thing annotations in COCO it allows us to perform a detailed analysis of stuff and the rich contextual relations that make our dataset unique. We have shown that (1) stuff is important: Stuff classes cover the majority of the image surface and more than a third of the nouns in human descriptions of an image; (2) many classes show frequent patterns of spatial context, and stuff classes appear in more varied contexts than things; (3) stuff is not generally easier to segment than things; (4) the larger training set that COCO-Stuff offers improves the semantic segmentation performance.

Acknowledgments. This work is supported by the ERC Starting Grant VisCul. The annotations were done by the crowdsourcing startup Mighty AI, and financed by Mighty AI and the Common Visual Data Foundation.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. on PAMI*, 34(11):2274–2282, 2012.

[2] E. Adelson. On seeing stuff : The perception of materials by humans and machines. In *SPIE proceedings series*, pages 1–12. Society of Photo-Optical Instrumentation Engineers, 2001.

[3] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. What's the point: Semantic segmentation with point supervision. In *ECCV*, 2016.

[4] S. Bell, P. Upchurch, N. Snavely, and K. Bala. OpenSurfaces: A richly annotated catalog of surface appearance. In *SIGGRAPH*, 2013.

[5] S. Bell, P. Upchurch, N. Snavely, and K. Bala. Material recognition in the wild with the materials in context database. In *CVPR*, 2015.

[6] P. Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover Publications, 1966.

[7] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Patt. Rec. Letters*, 30(2):88–97, 2009.

[8] H. Caesar, J. Uijlings, and V. Ferrari. Region-based semantic segmentation with end-to-end training. In *ECCV*, 2016.

[9] B. Caputo, E. Hayman, M. Fritz, and J.-O. Eklundh. Classifying materials in the real world. *Image and Vision Computing*, 28(1):150–163, 2010.

[10] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler. Annotating object instances with a Polygon-RNN. In *CVPR*, 2017.

[11] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *ICLR*, 2015.

[12] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *CVPR*, 2014.

[13] 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.

[14] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *CVPR*, 2015.

[15] K. Dana, B. Van Ginneken, S. Nayar, and J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics (TOG)*, 18(1):1–34, 1999.

[16] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

[17] I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *IEEE Trans. on PAMI*, 36(2):222–234, 2014.

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

[19] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Trans. on PAMI*, 32(9), 2010.

[20] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004.

[21] D. Forsyth, J. Malik, M. Fleck, H. Greenspan, T. Leung, S. Belongie, and C. Bregler. Finding pictures of objects in large collections of images. In *International Workshop on Object Representation in Computer Vision*, 1996.

[22] F. Galasso, R. Cipolla, and B. Schiele. Video segmentation with superpixels. In *ACCV*, 2012.

[23] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012.

[24] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009.

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

[26] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[27] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *ECCV*, 2008.

[28] A. Ion, J. Carreira, and C. Sminchisescu. Probabilistic joint image segmentation and labeling. In *NIPS*, pages 1827–1835, 2011.

[29] C. Ionescu, O. Vantzos, and C. Sminchisescu. Training deep networks with structured layers by matrix backpropagation. In *ICCV*, 2015.

[30] S. Jain and K. Grauman. Click carving: Segmenting objects in video with point clicks. In *Proceedings of the Fourth AAAI Conference on Human Computation and Crowdsourcing*, 2016.

[31] B. Kim, M. Sun, P. Kohli, and S. Savarese. Relating things and stuff by high-order potential modeling. In *ECCV*, 2012.

[32] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. *IJCV*, 2009.

[33] P. Krähenbühl and V. Koltun. Efficient inference in fully connected CRFs with gaussian edge potentials. In *NIPS*, pages 109–117, 2011.

[34] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Trans. on PAMI*, 27(8):1265–1278, 2005.

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

[36] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *IEEE Trans. on PAMI*, 33(12):2368–2382, 2011.

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

[38] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014.

[39] R. Mottaghi, S. Fidler, J. Yao, R. Urtasun, and D. Parikh. Analyzing semantic segmentation using hybrid human-machine CRFs. In *CVPR*, pages 3143–3150, 2013.

[40] J. Pont-Tuset, M. A. F. Guiu, and A. Smolic. Semi-automatic video object segmentation by advanced manipulation of segmentation hierarchies. In *International Workshop on Content-Based Multimedia Indexing*, 2015.

[41] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *ICCV*, 2007.

[42] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004.

[43] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *IJCV*, 2015.

[44] L. Sharan, R. Rosenholtz, and E. Adelson. Material perception: What can you see in a brief glance? *Journal of Vision*, 14(9), 2014.

[45] M. Shi, H. Caesar, and V. Ferrari. Weakly supervised object localization using things and stuff transfer. In *ICCV*, 2017.

[46] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006.

[47] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[48] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[49] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017.

[50] J. Tighe and S. Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. In *ECCV*, 2010.

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

[52] J. Tighe and S. Lazebnik. Superparsing - scalable nonparametric image parsing with superpixels. *IJCV*, 101(2):329–349, 2013.

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

[54] K. Toutanova, D. Klein, C. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, 2003.

[55] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 2013.

[56] J. Wang and A. Yuille. Semantic part segmentation using compositional model combining shape and appearance. In *CVPR*, 2015.

[57] T. Wang, B. Han, and J. Collomosse. Touchcut: Fast image and video segmentation using single-touch interaction. *CVIU*, 2014.

[58] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, and A. Oliva. SUN database: Exploring a large collection of scene categories. *IJCV*, pages 1–20, 2014.

[59] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from Abbey to Zoo. In *CVPR*, 2010.

[60] J. Xu, A. G. Schwing, and R. Urtasun. Learning to segment under various forms of weak supervision. In *CVPR*, 2015.

[61] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Parsing clothing in fashion photographs. In *CVPR*, 2012.

[62] R. Zhang, S. A. Candra, K. Vetter, and A. Zakhor. Sensor fusion for semantic segmentation of urban scenes. In *ICRA*, 2015.

[63] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ADE20K dataset. In *CVPR*, 2017.