COCO-Stuff: Thing and Stuff Classes in Context

Holger Caesar Jasper Uijlings Vittorio Ferrari

University of Edinburgh

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 (determined through contextual reasoning); (3) physical attributes, material types and geometric properties of the scene.

To understand stuff and things in context we annotate 10,000 images of the COCO dataset with a broad range of stuff classes, using a specialized stuff annotation protocol allowing us to efficiently label each pixel. On this dataset, we analyze several aspects: (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 importance of several visual criteria to discriminate stuff and thing classes; (c) we study the spatial relations between stuff and things, highlighting the rich contextual relations that make our dataset unique. Furthermore, we show experimentally how modern semantic segmentation methods perform on stuff and thing classes and answer the question whether stuff is easier to segment than things.

We release our new dataset and the trained models online, hopefully promoting further research on stuff and stuff-thing contextual relations.

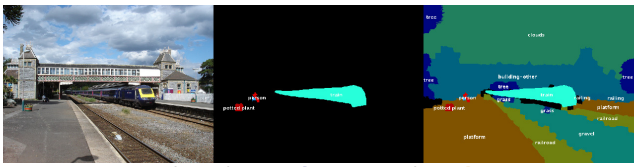
1. Introduction

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 [25, 29] and shape [25, 49, 54, 39, 21, 18], and have identifiable parts (e.g. a car has wheels). Indeed, the main recognition challenges [22, 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 internal 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 is 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 balls). Furthermore, stuff is 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.

In this paper we introduce COCO-Stuff, a new dataset which augments the popular COCO dataset [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. Compared to other datasets [38, 36], COCO-Stuff allows us to study stuff-thing interactions in more complex images, as COCO contains mainly scenes with many small objects, and with a greater diversity of thing and stuff classes. Fig. 1 shows an example of an image, its annotations in COCO [35] 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 the stuff (i.e. track), and the spatial arrangements of the train and its surrounding stuff. All these elements are necessary for scene understanding. As the figure shows, our dataset offers much more comprehensive annotations.

Using our new dataset, we provide a detailed analysis of the role and importance of stuff, and of the contextual relations between stuff and thing classes. In particular we examine: (1) What visual cues are important to distinguish things from stuff? (Sec. 4.2) (2) How important is stuff in



A large long **train** on a steel **track**.
A blue and yellow transit **train** leaving the **station**.
A **train** crossing beneath a city **bridge** with brick **towers**.
A **train** passing by an over **bridge** with a railway **track** (..).
A **train** is getting ready to leave the train **station**.

Figure 1. *(left) An example image, (middle) its thing annotations in COCO [35] and (right) the enriched stuff and thing annotations in our COCO-Stuff. Just having the train, a person, a bench and a 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 motion, stuff-thing interactions (train leaving the station) and thing-thing interactions (person waiting for a different train). This level of analysis 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).*

terms of human scene descriptions (according to the COCO captions)? (Sec. 4.1) (3) How important is stuff in terms of image surface cover? (Sec. 4.1) (4) How are stuff and things contextually related to each other? (Sec. 4.3) (5) How well do modern semantic segmentation methods [14, 37] perform on stuff and things? (Sec. 4.4) (6) Is recognizing stuff really easier than recognizing things, as claimed in [48, 49, 51, 36, 30]? (Sec. 4.4)

We release our new dataset and the trained semantic segmentation models at `http://calvin.inf.ed.ac.uk/datasets/coco-stuff`, hoping to further promote research on stuff and stuff-thing contextual relations.

2. Related Work

2.1. Stuff and things

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 [25, 59, 30, 49, 54, 39, 21, 18] (*sky*, *grass*, *water*). (2) Size: Things occur at characteristic sizes with little variance, whereas stuff regions are highly variable in size [25, 2, 29]. (3) Parts: Thing classes have identifiable parts [56, 23], 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 often not countable [2] and have no clearly defined instances [18, 27, 51]. (5) Texture: Stuff classes are typically highly textured [25, 29, 49, 18]. 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 [49, 51, 33, 18]. Moreover, several works have shown that stuff is a useful contextual cue to detect things and vice versa [41, 29, 33, 38].

Dataset	Images	Classes	Stuff classes	Thing classes	Year
MSRC 21 [45]	591	21	6	15	2006
KITTI semantics [40]	203	14	9	4	2016
CamVid [11]	700	32	13	15	2008
Cityscapes [16]	25,000	30	13	14	2016
SIFT Flow [36]	2,688	33	15	18	2009
Barcelona [48]	15,150	170	31	139	2010
LM+SUN [50]	45,676	232	52	180	2010
PASCAL Context [38]	10,103	540	152 (26)	388	2014
NYUD [46]	1,449	894	190 (54)	695	2012
COCO-Stuff	10,000	172	91 (**91**)	80	2016

Table 1. *An overview of datasets with pixel-level stuff and thing annotations. The number of stuff and thing classes are estimated given the definitions in Sec. 2.1. As PASCAL Context, NYUD and COCO-Stuff all have a large number of stuff classes, we report also the number of classes covering more than 96,000 pixels over the whole dataset (i.e. large enough to be usable, according to [38]). COCO-Stuff has a larger number of usable stuff classes than any other dataset, and in fact* all *its stuff classes are usable.*

2.2. Datasets

Stuff-only datasets. Early stuff datasets [10, 19, 34, 13] focused on texture classification and had simple images completely covered with a single textured patch. The more recent Describable Textures Dataset [15] instead collects textured patches in the wild, described by human-centric attributes. A related task is material recognition [44, 7, 8]. Although the recent Materials in Context dataset [8] 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 [26] 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.

Thing-only datasets. These datasets have bounding box or outline-level annotations of things, e.g. PASCAL VOC [22], 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 [45], NYUD [46], CamVid [11] and SIFT Flow [36] annotate less than 50 classes on less than 5,000 images. More recent large-scale datasets like Barcelona [48], LM+SUN [50], PASCAL Context [38] and Cityscapes [16] annotate tens of thousands of images with hundreds of classes. We compare our COCO-Stuff to these datasets at the end of Sec. 3.

Annotating datasets. Dense pixel-wise annotation of images is extremely costly. Several works use interactive segmentation methods [42, 57] to speedup annotation. Some works operate in a weakly supervised scenario, deriving full image annotations starting from manually annotated squiggles [6, 60] or points [6, 31]. These approaches take less time, but typically lead to lower quality.

In this work we provide a scheme to obtain high quality pixel-wise stuff annotations at low human costs by using superpixels and exploiting existing detailed thing annotations of COCO [35] (Sec. 3).

3. The COCO-Stuff dataset

The Common Objects in COntext (COCO) [35] is a large-scale dataset of images of unprecedented complexity. COCO has been designed specifically 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 [35]. In this paper we augment part of COCO by adding pixel-wise stuff annotations, which results in complete annotations in terms of nameable classes (both stuff and things) and their locations. Since COCO was designed to be about complex, yet natural scenes containing substantial areas of stuff, COCO-Stuff enables the exploration of rich relations between things and stuff. Therefore our dataset is an important 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.

Selecting images. We sampled 10,000 images from the 82,783 training images in COCO 2014. These images are sampled to preserve the distribution of labels of the entire dataset. Using the same sampling technique we divide the images into 9,000 train and 1,000 test images. This determines an official split that can be used for evaluating various algorithms (e.g. we present semantic segmentation results in Sec. 4.4). In total there are 2.5 billion annotated pixels in COCO-Stuff.

Defining stuff labels. COCO-Stuff contains 172 classes: 80 thing classes, 91 stuff classes and 1 class *unlabeled*. The 80 thing classes are the same as in COCO 2014 [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 the pixel.

Before annotation, we choose to predefine our label set. This contrasts with the common choice to have annotators use free-form text labels [48, 50, 38]. However, that 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]. Furthermore, different annotators typically use several synonyms to indicate the same class. These need to be merged a posteriori [48, 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. When two annotators annotate different images, two problems might arise: they could use *bridge* and *footbridge* to describe the same concept (i.e. *footbridge*), or they could both use *bridge* to describe two different concepts (e.g. *train bridge* and *footbridge*). It is unclear which one the annotator intended in each case. 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 (Fig. 3), e.g. classes like *cloth* and *curtain* have *textile* as parent, while classes like *moss* and *tree* have *vegetation* as parent. 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.

We now demonstrate that 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

Figure 2. *Annotated images from the COCO-Stuff dataset. To emphasize the depth ordering of stuff and thing classes we use bright colors for thing classes and darker colors for stuff classes.*

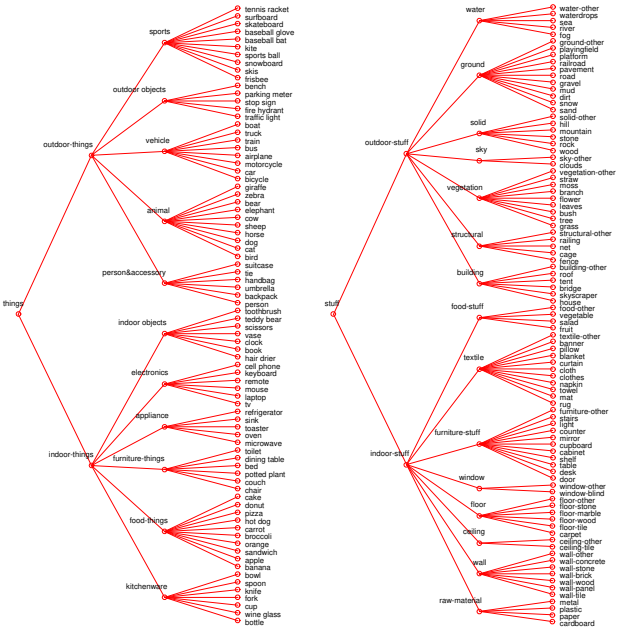


Figure 3. *The label hierarchy of the COCO-Stuff dataset. The dataset consists of thing classes and stuff classes. These are further divided into outdoor and indoor classes. Both are divided into super-categories (*vehicle, food, wall*). The labels used by the annotators form the leaf nodes of the tree.*

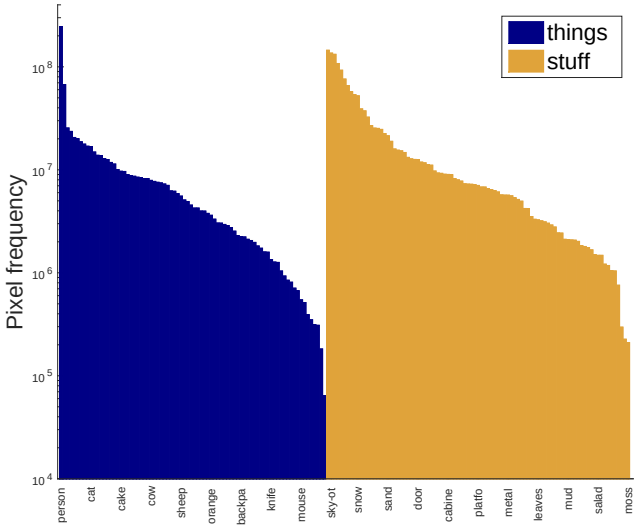


Figure 4. *Pixel-level frequencies for all classes in COCO-Stuff. Classes are categorized as stuff or things and sorted within their category. We can see that stuff and thing classes follow a similar pixel frequency distribution. Some representative class names are reported below the plot.*

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 12% 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. 4) 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.

Efficient stuff annotation. 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 homogenenous size (Fig. 5). Superpixels remove the need for precisely annotating the 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 results in a labeling task natural for humans (as opposed to superpixel algorithms which yield regions that greatly vary in size [24]). We accelerate annotation further by providing

annotators a size-adjustable paintbrush tool, which enables labeling large regions of stuff very efficiently (Fig. 5b).

We improve annotation efficiency even further by leveraging the highly accurate thing outlines available from COCO [35] (Fig. 5c). We show annotators images with thing outlines overlaid, and pixels belonging to things are clamped and unaffected by the annotator's brush. This results in a very lightweight experience, where the annotator merely needs to select a stuff class (like *snow* in the example) 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, sometimes even 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.

We use 10 expert annotators to label COCO-Stuff. This annotation protocol yields an annotation time of only 3 minutes to annotate stuff in one of the COCO images, which are highly complex (Fig. 2).

Comparison to other datasets. COCO-Stuff is much richer in both the number of stuff and thing classes than MSRC 21 [45], KITTI [40], CamVid [11], Cityscapes [16] and SIFT Flow [36] (Table 1). Compared to the Barcelona [48] and LM+SUN [50] datasets, it has 3× and 2× more stuff classes, respectively.

PASCAL Context [38] provides complete pixel-wise annotations for PASCAL VOC 2010. However, they used free-form label annotations which leads to annotations at different granularities and hence ambiguities, as discussed above. In contrast, in COCO-Stuff all labels are mutually exclusive and at a comparable level of granularity. Furthermore, while PASCAL Context has a seemingly impressive number of 540 classes, in practice most classes are too rare to be usable: [38, 18, 37, 17, 62, 5] only use the 60 most frequent classes, out of which only 28 are stuff classes (Table 1). Under the same definition of 'frequent enough to be usable' (i.e. $>$ 96,000 pixels), all COCO-Stuff stuff classes qualify. This is 91 stuff classes, 3× more than in PASCAL Context. Moreover, these classes have systematic names and are nicely organized in a meaningful hierarchy.

Finally, PASCAL Context is annotated with overlapping polygons. Therefore some pixels have multiple contradictory labels at the boundaries between objects/stuff. In our dataset instead, each pixel has exactly one label.

To conclude, for all the reasons above, we feel that COCO-Stuff is the most complete dataset of pixel-level annotated thing and stuff classes. Moreover, by building on COCO it also has natural language captions, which further supports rich scene understanding.

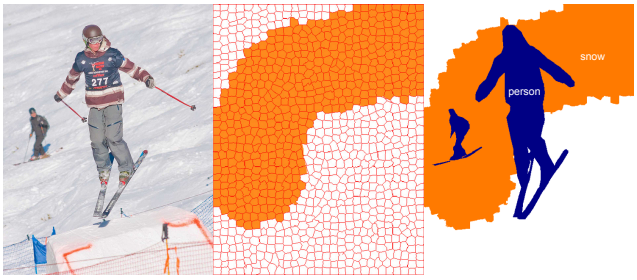


Figure 5. *Example of a) an image, b) the superpixel-based stuff annotation and c) the final labeling. The annotator can quickly annotate large stuff regions (snow) with a single mouse stroke using a paintbrush tool. Thing (person) annotations are copied from the COCO dataset. The transparency of each 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.*

4. Analysis of stuff and things

In this section we leverage COCO-Stuff to analyze various aspects of stuff and the relations between stuff and things: we analyze the relative importance of stuff and thing classes (Sec. 4.1); learn to discriminate between stuff and things (Sec. 4.2); study spatial contextual relations between stuff and things (Sec. 4.3); and analyze the behavior of semantic segmentation methods on stuff and things (Sec. 4.4).

4.1. Importance of stuff and things

We quantify here 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 COCO-Stuff annotations. Table 2 shows that the majority of pixels are stuff (66.2%). We also compute statistics for the 422K *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 64.3% of the regions are stuff and 35.7% 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 [53] to automatically detect nouns. Then we manually categorize the 200 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*).

Level	Stuff	Things
Pixels	**66.2%**	33.8%
Regions	**64.3%**	35.7%
Caption nouns	30.9%	**69.1%**

Table 2. *Relative frequency of stuff and thing classes in pixel-level annotations and caption nouns in COCO-Stuff.*

Table 2 shows the relative frequency of these nouns. Stuff covers about a third of the nouns (30.9%). This clearly shows the importance of stuff according to the COCO image captions.

4.2. Discriminating between stuff and things

Existing definitions of stuff and things state that they have different characteristics (Sec. 2.1). In this section we train a classifier to distinguish between stuff and thing classes based on several visual cues, and study their relative importance. We represent each class with a single feature vector summarizing its pixel-wise measurements over the whole dataset. Discriminative features will inform us about the special characteristics of stuff.

Features and protocol. For each pixel we compute the following features: hue, saturation, value, objectness, and normalized image coordinates. To compute objectness for a pixel, we compute the objectness score for 10,000 bounding box proposals per image with [3]. We then define the objectness score for a pixel to be the average score over all boxes that contain it, as done in [6, 55]. This score indicates how likely it is for a pixel to belong to *any* thing class [3].

For each cue and class, we compute a relative frequency histogram of all pixels in the dataset that belong to that class. We train a binary linear Support Vector Machine (SVM) classifier that classifies each of the 171 classes in COCO-Stuff as stuff or things. Due to the low number of samples we train and evaluate the SVMs in a leave-one-out manner: For each class in turn, we train on the remaining classes and test on the current class.

Results. Table 3 reports the classification accuracy for each cue, i.e. the percentage of *classes* correctly recognized as either things or stuff. Due to the slight imbalance in the number of stuff and thing classes, chance level is at 53.2%. Value, hue and saturation perform significantly above chance at 59.1%, 60.2% and 67.8%, respectively. The relative image position cue performs suprisingly well at 87.7%, confirming the positional bias observed in many prior works [32, 52, 61, 28, 4, 9]. The more complex objectness measure performs best at 90.6%. Using all cues we achieve the best result of 92.4%.

Fig. 6 shows a histogram of SVM decision values for all classes (using all cues at the same time). We can see that the SVMs separates stuff and things very well, with only a few errors. For each histogram bin we print examples of true positives and false positives of our classifiers. We can see that certain food classes (*salad*, *fruit*, *food-other*), as well as furniture classes (*chair*, *couch*) are misclassified. A more detailed analysis of the different cues shows that stuff classes tend to be generally brighter than things, less saturated, more green than red, have lower objectness values, and are less likely to be in the center of an image.

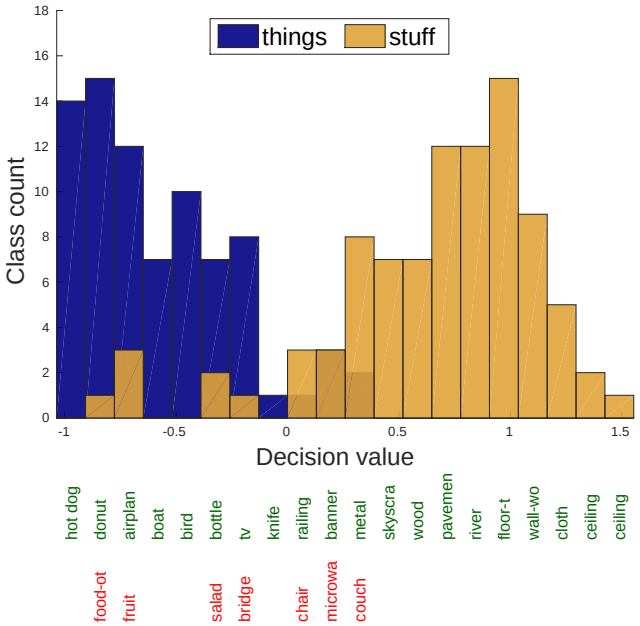


Figure 6. *Histogram of the decision values of our SVM classifiers trained to recognize each class in COCO-Stuff as either* thing *or* stuff. *The SVMs use the combined set of visual cues from Table 3. Underneith the histogram we print the class names of examples of true positives (green) and false positives (red) of the classifier for the respective decision value bin.*

Cue	Stuff-thing accuracy
Chance level	53.2%
Value	59.1%
Hue	60.2%
Saturation	67.8%
Position	87.7%
Objectness	90.6%
Combined	92.4%

Table 3. *Binary classification accuracy in recognizing each class in COCO-Stuff as either thing or stuff, based on different cues.*

4.3. 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. *train* (Fig. 7, leftmost column). The explanation is analoguous for all other classes. For every image containing a train, we extract a set of train regions, i.e. connected components of train pixels in the annotation map. Next we compute a

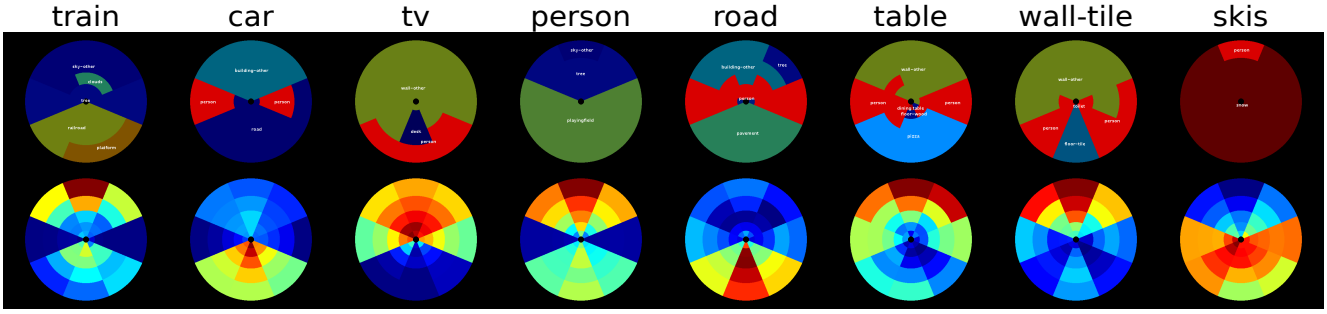


Figure 7. *Spatial context visualizations. (Top) Each disc is for a different reference class and shows the most likely other class at each direction and distance bin. (Bottom) The conditional probabilities of the most common class in each bin, as a measure of confidence. The values are normalized for each reference class and range from low (blue) to high (red). We also show examples for classes with high (person) and low (skis) mean entropy.*

3D histogram of image pixels surrounding the train 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 do (1) compute the distance between the pixel and the nearest point in the train region (normalized by image size); (2) compute the relative angle with respect to the center of mass of the train region.

Results. Fig. 7 shows the spatial context of 8 reference classes. This visualization reveals several interesting contextual relations. Trains are typically found above railroads and platforms (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. 7, bottom). In most cases the highest confidence is in regions above (*sky*, *building*, *wall*) or below (*road*, *table*, *person*, *floor*) 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.

As the figure shows, some classes have a rich and diverse context, composed of many other classes (e.g. *train*, *table*), while some classes have a simpler context (e.g. *skis* 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 *skis*, *hair drier* and *tie*. On average, we find that stuff classes have a significantly higher mean entropy than things (3.04 vs. 2.61), showing they appear in more varied contexts. We also find that the mean entropy is rather constant over distances (small: 2.80, big: 2.87) and directions (right/left/down: 2.77, up: 2.73).

Comparing the mean entropy of different datasets, we find that COCO-Stuff has the highest (2.85), followed by the top 60 classes of PASCAL Context (2.43) and SIFT Flow (1.25). This shows the contextual richness of COCO-Stuff.

Method	Classes	Class-average accuracy	Global accuracy	Mean IOU
FCN [37]	all	34.0%	52.0%	22.7%
	stuff	23.9%	50.4%	15.7%
	things	45.1%	61.7%	30.4%
DeepLab [14]	all	**38.1%**	**57.8%**	**26.9%**
	stuff	**26.9%**	**54.9%**	**18.6%**
	things	**50.5%**	**69.2%**	**36.2%**

Table 4. *Semantic segmentation results on COCO-Stuff for FCN and DeepLab.*

4.4. Semantic segmentation of stuff and things

We now analyze how modern semantic segmentation methods perform 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 Fully Convolutional Networks (FCN) [37] and DeepLab [14] as semantic segmentation methods. Both methods use the VGG-16 network [47] pre-trained on the ILSVRC classification dataset [43]. We follow the same experimental protocol for both models: train on 9,000 training images and test on 1,000 test images (as defined in Sec. 3). To evaluate test set performance we use the three following criteria, commonly used in the literature [37, 20, 12]: (1) global accuracy is the percentage of correctly labeled pixels in the dataset; (2) class-average 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 [22].

DeepLab vs FCN. As Table 4 shows, DeepLab outperforms FCN on all metrics and kinds of classes, confirming previous findings on other datasets [14].

Confusion matrices. Fig. 8 shows the confusion matrices for FCN on different levels of the hierarchy presented in Fig. 3. We can see that there is considerable confusion

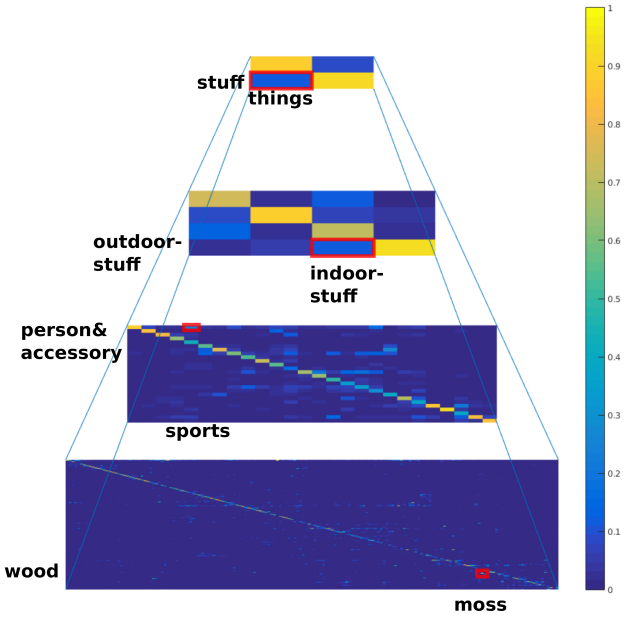


Figure 8. *Confusion matrices for FCN on COCO-Stuff. The lowest level corresponds to the 171 stuff and thing classes labeled in COCO-Stuff (Fig.3, leaf nodes). The higher levels are defined as in Fig. 3 (e.g. the next level contains* furniture, building, water*; the next one* indoor *vs.* outdoor*; and finally* stuff *vs.* things*). For each level we highlight an example of a pair of classes with particularly high confusion.*

on the lowest level, e.g. between *wood* and *moss*, *donut* and *sandwich*. On the higher levels instead, the amount of confusion is lower. This shows that classes that are close in our semantic hierarchy are also close in appearance space.

Is stuff easier than things? Several works show that stuff is easier to segment than things [48, 49, 51, 36, 30]. We argue that this is due to their choice of dataset, rather than a general observation. Despite stuff covering the majority of pixels in an image, most datasets only include a small number of very frequent and high-level stuff classes (Table 1). In contrast, in COCO-Stuff we selected a larger number of relevant stuff labels at a similar level of granularity as the existing thing labels. Therefore we have a similar number of stuff and thing classes, and a similar pixel frequency distribution for both (see Fig. 4).

Revisiting Table 4 we can see that both FCN and DeepLap perform systematically better on thing classes than on stuff. To factor out the influence of different region sizes for stuff and thing classes, we plot in Fig. 9 the global accuracy of FCNs as a function of region size, averaged over the number of regions of that size. We can see that FCNs perform better on things than on stuff for nearly the entire size range (except for very small regions with less than 300 pixels). For medium sized regions (about 30,000 pixels) the performance gap is the largest.

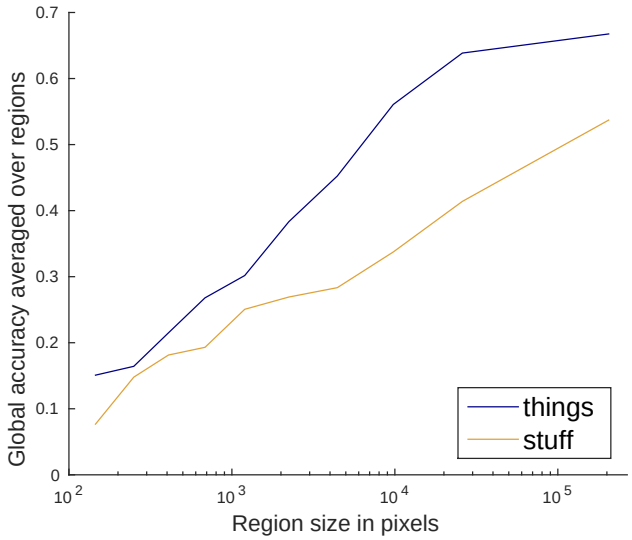


Figure 9. *The influence of region size on semantic segmentation performance, for FCN [37] on COCO-Stuff. Generally, thing classes are easier to segment than stuff classes. The performance depends a lot on the region size. For very small regions the performance on stuff and things is comparable. For medium sized regions the gap between stuff and things is the largest.*

To summarize, we have shown that stuff is harder to segment than things on COCO-Stuff, a dataset where both stuff and things are similarly distributed. Therefore we argue that stuff is not generally easier than things.

5. Conclusion

In this paper we presented the large-scale COCO-Stuff dataset. COCO-Stuff enriches 10,000 COCO images with 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.

Using COCO-Stuff, we have shown that (1) Stuff is important: Stuff classes cover the majority of the image surface and about a third of the nouns in human descriptions of an image. (2) Stuff and thing classes can be discriminated nicely based on simple visual cues. (3) Many classes show frequent patterns of spatial context, and stuff classes appear in more varied contexts than things. (4) Stuff is not generally easier to segment than things.

We release our new dataset and trained semantic segmentation models online, hoping to further promote research on stuff and stuff-thing contextual relations.

Acknowledgements. Work supported by the ERC Starting Grant VisCul.

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. 4

[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. 2

[3] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, 2010. 6

[4] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE Trans. on PAMI*, 2012. 6

[5] A. Arnab, S. Jayasumana, S. Zheng, and P. Torr. Higher order conditional random fields in deep neural networks. In *ECCV*, 2016. 5

[6] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. What's the point: Semantic segmentation with point supervision. In *ECCV*, 2016. 3, 6

[7] S. Bell, P. Upchurch, N. Snavely, and K. Bala. OpenSurfaces: A richly annotated catalog of surface appearance. In *SIGGRAPH*, 2013. 2

[8] S. Bell, P. Upchurch, N. Snavely, and K. Bala. Material recognition in the wild with the materials in context database. In *CVPR*, 2015. 2

[9] X. Boix, J. Gonfaus, J. van de Weijer, A. D. Bagdanov, J. Serrat, and J. Gonzàlez. Harmony potentials: Fusing global and local scale for semantic image segmentation. *IJCV*, 2012. 6

[10] P. Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover Publications, 1966. 2

[11] G. 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. 2, 5

[12] H. Caesar, J. Uijlings, and V. Ferrari. Region-based semantic segmentation with end-to-end training. In *ECCV*, 2016. 7

[13] 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. 2

[14] 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. 2, 7

[15] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *CVPR*, 2014. 2

[16] 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, 5

[17] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015. 5

[18] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *CVPR*, 2015. 1, 2, 5

[19] 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. 2

[20] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 7

[21] I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *IEEE Trans. on PAMI*, 36(2):222–234, 2014. 1, 2

[22] M. Everingham, S. Eslami, L. van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes challenge: A retrospective. *IJCV*, 2015. 1, 2, 7

[23] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Trans. on PAMI*, 32(9), 2010. 2

[24] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004. 4

[25] 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. 1, 2

[26] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009. 2

[27] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, 2014. 2

[28] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. 6

[29] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *ECCV*, 2008. 1, 2

[30] A. Ion, J. Carreira, and C. Sminchisescu. Probabilistic joint image segmentation and labeling. In *NIPS*, pages 1827–1835, 2011. 2, 8

[31] 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. 3

[32] A. Khosla, J. Xiao, A. Torralba, and A. Oliva. Memorability of image regions. In *NIPS*, 2012. 6

[33] B. Kim, M. Sun, P. Kohli, and S. Savarese. Relating things and stuff by high-order potential modeling. In *ECCV*, 2012. 2

[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. 2

[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. 1, 2, 3, 5

[36] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *IEEE Trans. on PAMI*, 33(12):2368–2382, 2011. 1, 2, 3, 5, 8

[37] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2, 5, 7, 8

[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. 1, 2, 3, 5

[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. 1, 2

[40] A. Ošep, A. Hermans, F. Engelmann, D. Klostermann, M. Mathias, and B. Leibe. Multi-scale object candidates for generic object tracking in street scenes. In *Proc. Intl. Conf. on Robotics and Automation*, 2016. 2, 5

[41] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *ICCV*, 2007. 2

[42] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004. 3

[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. 1, 2, 7

[44] L. Sharan, R. Rosenholtz, and E. Adelson. Material perception: What can you see in a brief glance? *Journal of Vision*, 14(9), 2014. 2

[45] 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. 2, 5

[46] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 2

[47] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 7

[48] J. Tighe and S. Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. In *ECCV*, 2010. 2, 3, 5, 8

[49] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *CVPR*, 2013. 1, 2, 8

[50] J. Tighe and S. Lazebnik. Superparsing - scalable nonparametric image parsing with superpixels. *IJCV*, 101(2):329–349, 2013. 2, 3, 5

[51] J. Tighe, M. Niethammer, and S. Lazebnik. Scene parsing with object instances and occlusion ordering. In *CVPR*, 2014. 2, 8

[52] L. to predict where humans look. Judd, t. and ehinger, k. and durand, f. and torralba, a. In *ICCV*, 2009. 6

[53] 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. 5

[54] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 2013. 1, 2

[55] A. Vezhnevets, V. Ferrari, and J. M. Buhmann. Weakly supervised semantic segmentation with multi image model. In *ICCV*, 2011. 6

[56] J. Wang and A. Yuille. Semantic part segmentation using compositional model combining shape and appearance. In *CVPR*, 2015. 2

[57] T. Wang, B. Han, and J. Collomosse. Touchcut: Fast image and video segmentation using single-touch interaction. *CVIU*, 2014. 3

[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. 3

[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. 2

[60] J. Xu, A. G. Schwing, and R. Urtasun. Learning to segment under various forms of weak supervision. In *CVPR*, 2015. 3

[61] L. Zhang, M. Tong, T. Marks, H. Shan, and G. Cottrell. SUN: A bayesian framework for saliency using natural statistics. *Journal of Vision*, 8(7):32, 2008. 6

[62] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 5