MVTec D2S: Densely Segmented Supermarket Dataset

Patrick Follmann[1,2][0000−0001−5400−2384], Tobias Böttger[1,2][0000−0002−5404−8662], Philipp Härtinger[1][0000−0002−7093−6280], Rebecca König[1][0000−0002−4169−6759], and Markus Ulrich[1][0000−0001−8457−5554]

[1] MVTec Software GmbH, 80634 Munich, Germany
{follmann,boettger,haertinger,koenig,ulrich}@mvtec.com
https://www.mvtec.com/research
[2] Technical University of Munich, 80333 Munich, Germany

Abstract. We introduce the Densely Segmented Supermarket (D2S) dataset, a novel benchmark for instance-aware semantic segmentation in an industrial domain. It contains 21 000 high-resolution images with pixel-wise labels of all object instances. The objects comprise groceries and everyday products from 60 categories. The benchmark is designed such that it resembles the real-world setting of an automatic checkout, inventory, or warehouse system. The training images only contain objects of a single class on a homogeneous background, while the validation and test sets are much more complex and diverse. To further benchmark the robustness of instance segmentation methods, the scenes are acquired with different lightings, rotations, and backgrounds. We ensure that there are no ambiguities in the labels and that every instance is labeled comprehensively. The annotations are pixel-precise and allow using crops of single instances for articial data augmentation. The dataset covers several challenges highly relevant in the field, such as a limited amount of training data and a high diversity in the test and validation sets. The evaluation of state-of-the-art object detection and instance segmentation methods on D2S reveals significant room for improvement.

Keywords: instance segmentation dataset, industrial application

1 Introduction

The task of *instance-aware semantic segmentation* (*instance segmentation* for short) can be interpreted as the combination of *semantic segmentation* and *object detection*. While *semantic segmentation* methods predict a semantic category for each pixel [32], *object detection* focuses on generating bounding boxes for all object instances within an image [27]. As a combination of both, *instance segmentation* methods generate pixel-precise masks for all object instances in an image. While solving this task was considered a distant dream a few years ago, the recent advances in computer vision have made instance segmentation a key focus of current research [9,19,32]. This is especially due to the progress in deep convolutional networks [17] and the development of strong baseline frameworks such as Faster R-CNN [27] and Fully Convolutional Networks (FCN) [32].

Related Work. All top-performing methods in common instance segmentation challenges are based on deep learning and require a large amount of annotated training data. Accordingly, the availability of large-scale datasets, such as *ADE20K* [37], *Cityscapes* [2], *ImageNet* [31], *KITTI* [6], *COCO* [22], *Mapillary Vistas* [25], *VOC* [4], *Places* [36], *The Plant Phenotyping Datasets* [24], or *Youtube-8M* [1], is of paramount importance.

Most of the above datasets focus on everyday photography or urban street scenes, which makes them of limited use for many industrial applications. Furthermore, the amount and diversity of labeled training data is usually much lower in industrial settings. To train a visual warehouse system, for instance, the user typically only has a handful of images of each product in a fixed setting. Nevertheless, at runtime, the products need to be robustly detected in very diverse settings.

With the availability of depth sensors a number of dedicated RGBD datasets have been published [15,16,28,29]: In comparison, these datasets are designed for pose estimation and generally have low resolution images. They often contain fewer scenes (e.g. 111 for [29]) that are captured with video [16] resulting in a high number of frames. Some datasets provide no class annotations [29]. [16] shows fewer, but similar categories to D2S, but only single objects are captured and annotated with lower quality segmentations. In [15], some of these objects occur in real scenes but only box-annotations are provided. The most similar to D2S is [28]: CAD-models and object poses are available and could be used to generate ground truth segmentation masks for non-deformable objects. Compared to D2S, the dataset does not show scenes with more than one instance of the same category and objects appear at a much lower resolution.

Only few datasets focus on industry-relevant challenges in the context of warehouses. The Freiburg Groceries Dataset [13], SOIL-47 [14], and the Supermarket Produce Dataset [30] contain images of supermarket products, but only provide class annotations on image level, and hence no segmentation. The Grocery Products Dataset [7] and GroZi-120 [23] include bounding box annotations that can be used for object detection. However, not all object instances in the images are labeled separately. To the best of our knowledge, none of the existing industrial datasets provides pixel-wise annotations on instance level. In this paper, we introduce the *Densely Segmented Supermarket (D2S) dataset*, which satisfies the industrial requirements described above. The training, validation, and test sets are explicitly designed to resemble the real-world applications of automatic checkout, inventory, or warehouse systems.

Contributions. We present a novel instance segmentation dataset with high-resolution images in a real-world, industrial setting. The annotations for the 60 different object categories were obtained in a meticulous labeling process and are of very high quality. Specific care was taken to ensure that every occurring instance is labeled comprehensively. We show that the high-quality region annotations of the training set can easily be used for artificial data augmentation. Using both the original training data and the augmented data leads to a significant improvement of the average precision (AP) on the test set by about 30 percentage points. In contrast to existing datasets, our setup and the choice of the objects ensures that there is no ambiguity in the labels and an AP of 100% is achievable by an algorithm that performs flawlessly. To evaluate the generalizability of methods, the training set is considerably smaller than the validation and test sets and contains mainly images that show instances of a single category on a homogeneous background. Overall, the dataset serves as a demanding benchmark and resembles real-world applications and their challenges. The dataset is publicly available[3].

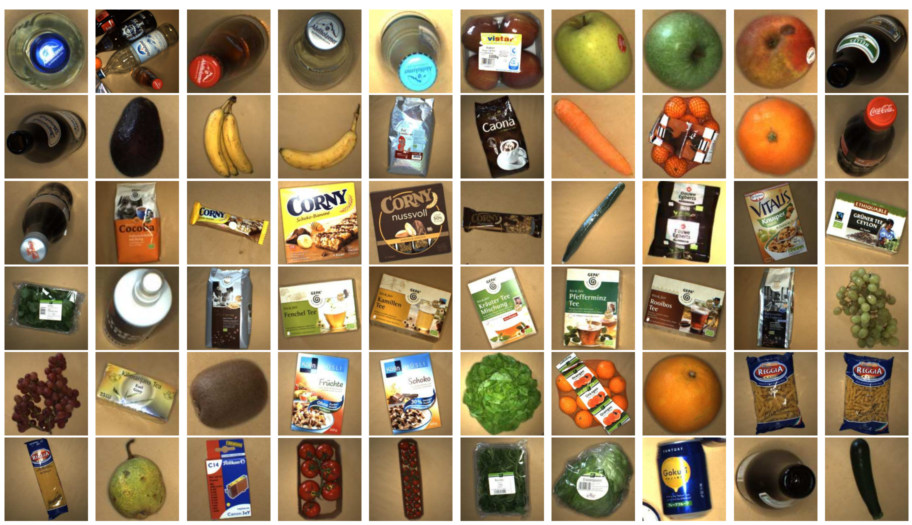

Fig. 1. Overview of the 60 different classes within the *D2S* dataset

2 The Densely Segmented Supermarket Dataset

The overall target of the dataset is to realistically cover the real-world applications of an automatic checkout, inventory, or warehouse system. For example, existing automatic checkout systems in supermarkets identify isolated products

[3] https://www.mvtec.com/research

that are conveyed on a belt through a scanning tunnel [3,12]. Even though such systems often provide a semi-controlled environment, external influences (e.g. lighting changes) cannot be completely avoided. Furthermore, the system's efficiency is higher if non-isolated products can be identified as well. Consequently, methods should be able to segment also partly occluded objects. Also, the background behind the products is not constant in many applications because of different types of storage racks in a warehouse system or because of dirt on the conveyer belt of a checkout system in the supermarket, for example.

We acquired a total of 21 000 images in 700 different scenes with various backgrounds, clutter objects, and occlusion levels. In order to obtain systematic test settings and to reduce the amount of manual work, a part of the image acquisition process was automated. Therefore, each scene was rotated ten times with a fixed angle step and acquired under three different illuminations.

Setup. The image acquisition setup is depicted in Fig. 2. A high-resolution (1920×1440) industrial color camera was mounted above a turntable. The camera was intentionally mounted off-centered with respect to the rotation center of the turntable to introduce more perspective variations in the rotated images.

Objects. An overview of the 60 different classes is shown in Fig. 1. The object categories cover a selection of common, everyday products such as fruits, vegetables, cereal packets, pasta, and bottles. They are embedded into a class hierarchy tree that splits the classes into groups of different packaging. This results in neighboring leafs being visually very similar, while distant nodes are visually more different, even if they are semantically similar products, e.g. single apples in comparison to a bundle of apples in a cardboard tray. The class hierarchy can be used, for instance, for advanced training and evaluation strategies similar to those used in [26]. However, it is not used in the scope of this paper.

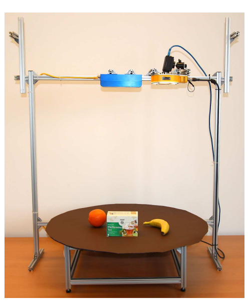

Fig. 2. The *D2S* image acquisition setup. Each scene was rotated ten times using a turntable. For each rotation, three images under different illuminations were acquired

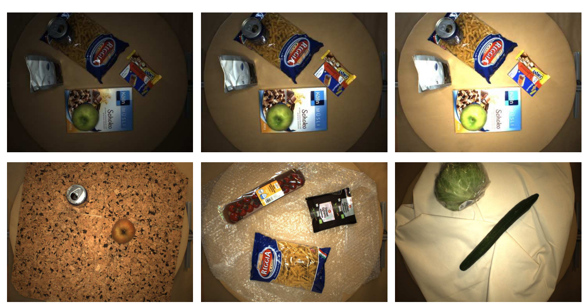

Fig. 3. (*Top*) Each scene was acquired under three different lightings. (*Bottom*) As opposed to the training set (where a single uniform background is used), the test and validation sets include three additional backgrounds. This allows for a detailed evaluation of the robustness of the methods

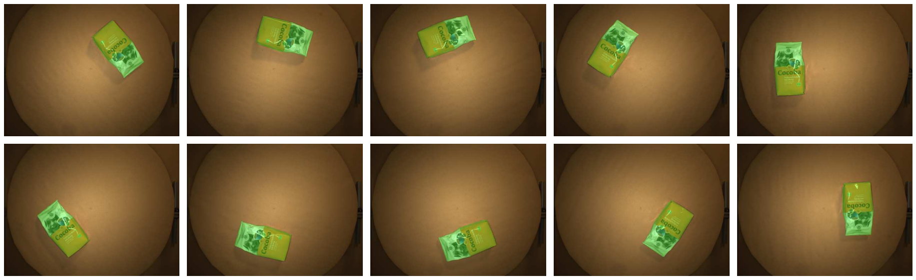

Fig. 4. Each scene was acquired at ten different rotations in steps of 36°. The camera was mounted slightly off-centered in order to introduce more variation in the images

Rotations. To increase the number of different views and to evaluate the invariance of approaches with respect to rotations [5,38], each scene was rotated ten times in increments of 36°. The turntable allowed automation and ensured precise rotation angles. An example of the ten rotations for a scene from the training set is displayed in Fig. 4.

Lighting. To evaluate the robustness of methods to illumination changes and different amounts of reflection, each scene and rotation was acquired under three different lighting settings. For this purpose an LED ring light was attached to the camera. The illumination was set to span a large spectrum of possible lightings, from under- to overexposure (see *top* of Fig. 3).

Background. The validation and test scenes have a variety of different backgrounds that are shown in Fig. 3 (*bottom*). This allows to evaluate the generalizability of approaches. In contrast, the training set is restricted to images with a single homogeneous background. It is kept constant to imitate the settings of a warehouse system, where new products are mostly imaged within a fixed environment and not in the test scenario.

Fig. 5. Objects appear with different amounts of occlusion. These may either be caused by objects of the same class, objects of a different class or by clutter objects not within the training set

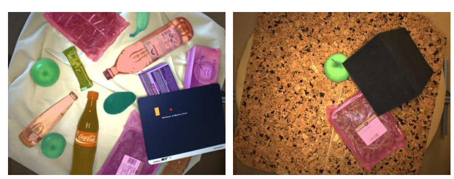

Fig. 6. To test the robustness of approaches to unseen clutter objects, objects not within the training set were added to the validation and test sets (e.g., a mouse pad and a black foam block)

Occlusion and Clutter. As indicated in Fig. 5, occlusions may arise from objects of the same class, objects of a different class, or from clutter objects. Clutter objects have a category that is not present in the training images. They were added explicitly to the validation and test images to evaluate the robustness to novel objects. Examples of the selected clutter objects are shown in Fig. 6.

3 Dataset Splitting

In contrast to existing datasets for instance-aware semantic segmentation, such as *VOC* [4] and *COCO* [22], the *D2S* training set has a different distribution with respect to image and class statistics than the validation and test sets. The complexity of the captured scenes as well as the average number of objects per image are substantially higher in the validation and test sets (see Table 1). The motivation for this choice of split is to follow common industrial requirements, such as: low labelling effort, low complexity of training set acquisition for easy replicability, and the possibility to easily add new classes to the system.

The split is performed on a per-scene basis: all 30 images of a scene, i.e. all combinations of the ten rotations and three lightings, are included in either the training, the validation, or the test set. In the following, we describe the rules for generating the splits.

Training Split. To meet the mentioned industrial requirements, the training scenes are selected to be as simple as possible: They have a homogeneous background, mostly contain only one object and the amount of occlusions is reduced to a minimum. To summarize, we add scenes to the training split that

- contain only objects of one category[4],
- provide new views of an object,
- only contain objects with no or marginal overlap,
- have no clutter and a homogeneous background.

The total number of scenes in the training set is 147, resulting in 4380 images of 6900 objects. The rather small training set should encourage work towards the generation of augmented or synthetic training data, for instance using generative adversarial networks [8,11,18,34,35].

Validation and Test Splits. The remaining scenes are split between the validation and the test set. They consist of scenes with

- single or multiple objects of different classes,
- touching or occluded objects,
- clutter objects and
- varying background.

[4] In order to provide similar views of each object class as they are visible in the validation and test set, four scenes were added to the training set that contain two distinct classes.

Table 1. Split statistics. Due to our splitting strategy, the number of images and the number of instances per image is significantly lower for the training set. The complexity of validation and test scenes is approximately the same

split	`all`	`train`	`val`	`test`
# scenes	700	146	120	434
# images	21000	4380	3600	13020
# objects	72447	6900	15654	49893
# objects/image	3.45	1.58	4.35	3.83
# scenes w. occlusion	393	10	84	299
# scenes w. clutter	86	0	18	68
rotations		✓	✓	✓
lighting variation		✓	✓	✓
background variation			✓	✓
clutter			✓	✓

These scenes were chosen such that the generalization capabilities of approaches can be evaluated. Additionally, current methods struggle with heavy occlusion and novelty detection. These issues are addressed by this choice of splits as well. The split between validation and test set was performed on subgroups of images containing the same number of total and occluded objects. This ensures that both sets have approximately the same distribution. The ratio of the number of scenes in the validation and test set is chosen to be 1:4. The reasons for this decision are twofold: First, the evaluation of the model on a small validation set is faster. Second, we do not want to encourage training on the validation set, but stimulate work on approaches that require little training data or use augmentation techniques. The statistics of the number of images and objects in the splits are visualized in Table 1.

4 Statistics & Comparison

In this section, we compare our dataset to *VOC* [4] and *COCO* [22]. These datasets have encouraged many researchers to work on instance segmentation and are frequently used to benchmark state-of-the-art methods.

Dataset Statistics. As summarized in Table 2, *D2S* contains significantly more object instances than *VOC*, but fewer than *COCO*. Specifically, although the *D2S* training set is larger than that of *VOC*, the number of training objects is less than 1% of those in *COCO*. This choice was made intentionally, as in many industrial applications it is desired to use as few training images as possible. In contrast, the proportion of validation images is significantly larger for *D2S* in order to enable a thorough evaluation of the generalization capabilities. On average, there are half as many objects per image in *D2S* as in *COCO*.

Table 2. Dataset statistics. Number of images and objects per split, average number of objects per image and number of classes for *D2S* (ours), *VOC 2012*, and *COCO*. *For *VOC 2012* and *COCO*, the object numbers are only available for the training and validation set

Dataset		*VOC*	*COCO*	*D2S*
# images	all	4369	163957	21000
	`train`	1464	118287	4380
	`val`	1449	5000	3600
	`test`	1456	40670	13020
# objects	all	-	-	72447
	`train`	3507	849941	6900
	`val`	3422	36335	15654
	`test`	-	-	49893
# obj/img		2.38*	7.19*	3.45
# classes		20	80	60

Class Statistics. Since the images of *COCO* and *VOC* were taken from flickr[5], the distribution of object classes is not uniform. In both datasets, the class *person* dominates, as visualized in Fig. 7: 31% and 25% of all objects belong to this class for *COCO* and *VOC*, respectively. Moreover, 10% of the classes with the highest number of objects are represented by 51% and 33% of all objects, while only 5.4% and 13.5% of the objects belong to the 25% of classes with the lowest number of objects. This class imbalance is valid since both *COCO* and *VOC* represent the real world where some classes naturally appear more often than others. In the evaluation all classes are weighted uniformly. Therefore, the class imbalance inherently poses a challenge to learn all classes equally well,

[5] https://www.flickr.com

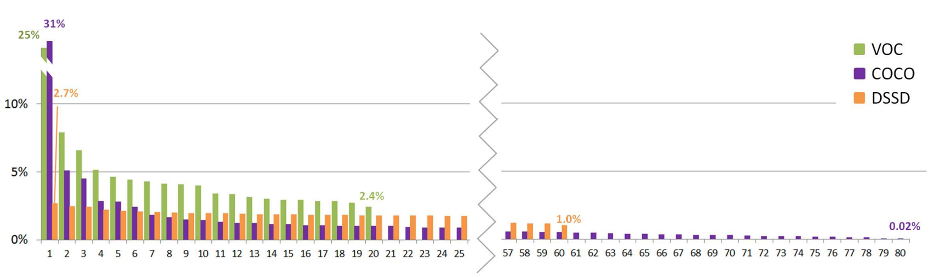

Fig. 7. Ratio of objects per class for *D2S* (*orange*), *VOC* (*green*) and *COCO* (*violet*). In *COCO* and *VOC*, the class *person* is dominant and some classes are underrepresented. In *D2S*, the number of objects per class is uniformly distributed. Note that for *COCO* and *VOC* the diagram was calculated based on train and val splits

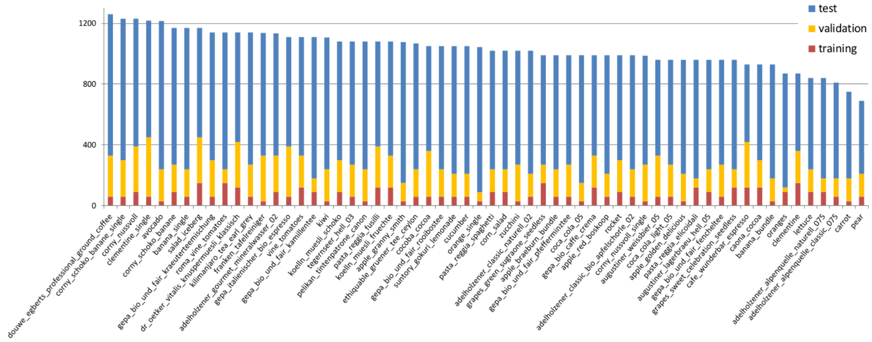

Fig. 8. Number of images per class and split sorted by the total number of images per class for *D2S*. The number of images per class is almost uniformly distributed

independent from the number of training samples. For example, the *COCO 2017* validation set contains nine instances of the class *toaster*, but 10 777 instances of *person*. Nevertheless, both categories are equally weighted in the calculation of the mean average precision, which is the metric used for ranking the methods in the *COCO* segmentation challenge.

There is no such class imbalance in *D2S*. In the controlled environment of the supermarket scenario, all classes have the same probability to appear in an image. The class with the highest number of objects is represented by only 2.7% of all objects. Only 14% of the objects represent the 10% of classes with the highest number of objects, while 19% of the objects are from the 25% of classes with the lowest number of objects. The class distribution of *D2S* is visualized in Fig. 8, where the number of images per class is shown in total and for each split. As mentioned above, the number of images for each class is rather low in the training split, especially for classes that have a similar appearance for different views, such as *kiwi* and *orange_single*. Note that, although the split choice between validation and test set is not made on the class level, each class is well represented in both sets. The key challenge of the *D2S* dataset is thus not the handling of underrepresented classes, but the low amount of training data.

Label Consistency. It is difficult to ensure that all object instances in large real-world datasets are labeled consistently. On the one hand, it is hard to establish a reliable review process for the labeling of large datasets, e.g. to avoid unlabeled objects. On the other hand, some labels are ambiguous by nature, for instance a painting of a person. Fig. 9 shows examples for label inconsistencies from *ADE20K* [37], *VOC* and *COCO*.

In *D2S*, the object classes are unambiguous and have been labeled by six expert annotators. All present objects are annotated with high quality labels. A perfect algorithm, which flawlessly detects and segments every object in all images of the *D2S* dataset, will achieve an AP of 100%. This is not the case for *COCO*, *VOC*, and *ADE20K*. In these datasets, if an algorithm correctly

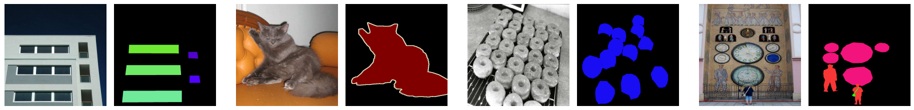

Fig. 9. Large real-world datasets are extremely difficult to label consistently. In the examples from *ADE20K*, *VOC* and *COCO*, some labels are missing (*from left to right*): a window, the sofa, some donuts, and the painting of a person

detects one of the objects that is not labeled, the missing ground truth leads to a false positive. Furthermore, if such an object is not found by an algorithm, the resulting false negative is not accounted for. As algorithms improve, this might prevent better algorithms from obtaining higher scores in the benchmarks. In *COCO*, this problem is addressed using *crowd annotations*, i.e. regions containing many objects of the same class that are ignored in the evaluation. However, crowd annotations are not present in all cases.

5 Benchmark

In this section, we provide first benchmark results for our dataset. We evaluate the performance of state-of-the-art methods for object detection [21,27] and instance segmentation [9,19]. We experiment with various training sets, which differ in the number of rotations and the availability of under- and overexposed images. Furthermore, we evaluate a simple approach for augmenting the training data artificially.

5.1 Evaluated Methods

Object Detection. For the object detection task, we evaluate the performance of Faster R-CNN [27] and RetinaNet [21]. We use the official implementations of both methods, which are provided in the Detectron[6] framework. Both methods use a ResNet-101 [10] backbone with Feature Pyramid Network [20].

Instance Segmentation. For the instance segmentation task, we evaluate the performance of Mask R-CNN [9] and FCIS [19]. We use the official implementation of Mask R-CNN in the Detectron framework and the official implementation of FCIS provided by the authors[7]. Mask R-CNN uses a ResNet-101 with Feature Pyramid Network as backbone, while FCIS uses a plain ResNet-101. Since both methods output boxes in addition to the segmentation masks, we also include them in the object detection evaluation.

[6] https://github.com/facebookresearch/Detectron

[7] https://github.com/msracver/FCIS

Training. All methods are trained end-to-end. The network weights are initialized with the COCO-pretrained models provided by the respective authors. The input images are resized to have a shorter side of 800 pixels (600 pixels for FCIS, respectively). All methods use horizonal flipping of the images at training time. FCIS uses *online hard example mining* [33] during training.

5.2 Evaluation Metric

The standard metric used for object detection and instance segmentation is *mean average precision* (mAP) [4]. It is used, for instance, for the ranking of state-of-the-art methods in the *COCO* segmentation challenge [22]. We compute the mAP exactly as in the official COCO evaluation tool[8] and give its value in percentage points. The basic average precision (AP) is the area under the precision-recall curve, computed for a specific intersection over union (IoU) threshold. In order to reward algorithms with better localization, the AP is usually averaged over multiple IoU thresholds, typically the interval $[0.5, 0.95]$ in steps of 0.05. The mAP is the mean over APs of all classes in the dataset. In the following, we just use the abbreviation AP for the value averaged over IoUs and classes. When referring to class-averaged AP for a specific IoU threshold, e.g. 0.5, we write AP_{50}.

5.3 Data Augmentation

In order to keep the labeling effort low and still achieve good results, it is crucial to artificially augment the existing training set such that it can be used to train deep neural networks. Hence, we experiment with a simple data augmentation technique, which serves as baseline for more sophisticated approaches. In particular, we simulate the distribution of validation and test set using only the annotations of the training set. For this purpose, we assemble 10 000 new artificial images that contain one to fifteen objects randomly picked from the training split. We denote the augmented data as `aug` in Table 3. For each generated image, we randomly sample the lighting and number of object instances. For each instance, we randomly sample its class, the orientation, and the location in the image. The background of these images is the plain turntable. We make sure that the instances' region centers lie on the turntable and that occluded objects have a visible area larger than 5000 pixels. Fig. 10 shows example images of the artificially augmented dataset for all three different lightings. Due to the high-quality annotations without margins around the object border, the artificially assembled images have an appearance that is very similar to the original test and validation images.

5.4 Results

When trained on the full training set `train` and evaluated on the `test` set, the instance segmentation methods provide solid baseline APs of 49.5% (Mask

[8] https://github.com/cocodataset/cocoapi

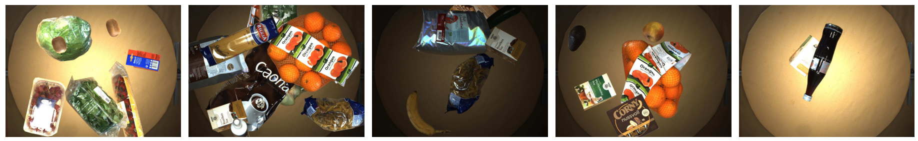

Fig. 10. The artificial augmented training set is generated by randomly assembling objects from the basic training set

R-CNN) and 45.6% (FCIS). The object detection results are on a similar level, with APs of 46.5% (Mask R-CNN), 44.0% (FCIS), 46.1% (Faster R-CNN), and 51.0% (RetinaNet). Tables 3 and 4 show the results in full detail.

Ablation Study. As aforementioned, the *D2S* splits are based on scenes, i.e. all rotations and lightings for one placement of objects are included in the same split. To evaluate the importance of these variations and the ability of methods to learn invariance with respect to rotations and illumination, we perform an ablation study. For this purpose, we create three subsets of the full training set `train`. The `train_rot0` set contains all three lightings, but only the first rotation of each scene. The `train_light0` set contains only the default lighting, but all ten rotations of each scene. The `train_rot0_light0` set contains only the default lighting and the first rotation for each scene.

The resulting AP values of the instance segmentation methods Mask R-CNN and FCIS are summarized in Table 3 (top). As expected, we obtain the best results when training on the full `train` set. Training only on the first rotation reduced the AP on the test set by 15.7% and 9.1% for Mask R-CNN and FCIS, respectively. Training only with default lighting reduced the AP slightly by 3.4% for Mask R-CNN and increased the AP by a neglible 0.4% for FCIS. Training on `train_rot0_light0` reduced the AP by 13.2% and 12.9%, respectively. Overall, the results indicate that the models are more invariant to changes in lighting than to rotations of the objects.

Data Augmentation. As shown in Table 3, training on the augmented dataset `aug` boosts the AP on the test set to 76.1% and 69.8% for Mask R-CNN and FCIS, respectively. This is significantly higher than the 49.5% and 45.6% achieved by training on the original `train` set. Combining the sets `train` and `aug` to `train+aug` further improves the AP by 8.3% and 2.7%, respectively.

Object Detection. We conduct the same ablation study for the task of object detection. The resulting AP values for all training splits of the methods Faster R-CNN and RetinaNet, as well as the results of instance segmentation methods Mask R-CNN and FCIS evaluated on bounding box level, are summarized in Table 4. It is interesting to note, that these AP values are not always better than the AP values obtained for the more difficult task of instance segmentation. On the one hand, we believe that the AP values for object detection and

Table 3. Instance segmentation benchmark results on the test set. Mean average precision values for models trained on different training sets. (*Top*) Training on different subsets of the `train` set. (*Bottom*) Training on augmented data yields the highest AP values

	Mask R-CNN			FCIS		
	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
`train`	49.5	57.6	51.3	45.6	58.3	51.3
`train_rot0`	33.8	41.6	35.6	36.5	47.5	41.8
`train_light0`	46.1	54.8	48.0	46.0	59.3	52.0
`train_rot0_light0`	36.3	45.1	38.6	32.7	43.4	38.1
`aug`	71.6	86.9	81.7	69.8	87.6	82.4
`train+aug`	**79.9**	**89.1**	**85.3**	**72.5**	**88.1**	**83.5**

Table 4. Object detection benchmark results on the test set. Mean average precision values for models trained on different training sets

	Mask R-CNN			FCIS			Faster R-CNN			RetinaNet		
	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
`train`	46.5	58.3	53.5	44.0	59.4	51.7	46.1	55.2	49.7	51.0	61.0	52.8
`train_rot0`	34.1	42.5	38.3	34.6	48.2	41.3	36.7	46.9	41.5	32.9	39.8	34.5
`train_light0`	45.5	55.7	49.5	44.0	60.3	51.9	43.7	53.9	47.8	51.7	62.0	53.6
`train_rot0_light0`	35.7	46.0	40.5	29.9	43.9	35.4	34.3	44.3	39.0	31.6	38.9	33.2
`aug`	72.9	87.9	82.0	**69.9**	88.1	80.7	73.5	88.4	82.2	74.2	86.9	81.4
`train+aug`	**78.3**	**89.8**	**84.9**	68.3	**88.5**	**80.9**	**78.0**	**90.3**	**84.8**	**80.1**	**89.6**	**84.5**

instance segmentation are generally very similar because current instance segmentation methods are based on object detection methods like Faster R-CNN. On the other hand, instance segmentation methods can even outperform object detection methods since a nearly perfect mask can still be generated from a too large underlying box proposal. It is also the case that the box-IoU is a lot more sensitive to the four box coordinates than the final segmentation. A third possible explanation is that the gradients of the mask branch help to learn even more descriptive features. For all methods the overall performance is very similar. Reducing the training set to only one rotation or only one lighting per scene results in worse performance. Analogously, augmenting the dataset by generating artificial training images results in a strong improvement.

Qualitative results. We show qualitative results of the best-performing method Mask R-CNN in Fig. 11. Furthermore, Fig. 12 shows typical failure cases we observed for Mask R-CNN and FCIS on the *D2S* dataset. More qualitative results are provided in the supplementary material.

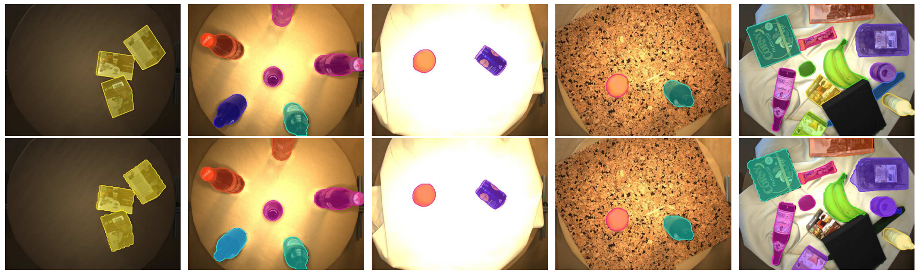

Fig. 11. (*Top*) Ground truth annotations from the *D2S* `val` and `test` sets. (*Bottom*) Results of Mask R-CNN trained on the `train` set. The classes are indicated by different colors

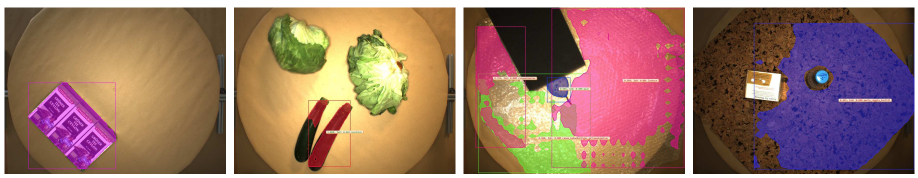

Fig. 12. Typical failure cases of of Mask R-CNN and FCIS on *D2S*. (*From left to right*) (*1*) Nearby objects are detected as a single instance. (*2*) Segmentation mask spans to neighboring objects. (*3 and 4*) Background is falsely detected as object

6 Conclusion

We have introduced *D2S*, a novel dataset for instance-aware semantic segmentation that focuses on real-world industrial applications. The dataset addresses several highly relevant challenges, such as dealing with very limited training data. The training set is intentionally small and simple, while the validation and test sets are much more complex and diverse. As opposed to existing datasets, *D2S* has a very uniform distribution of the samples per class. Furthermore, the fixed acquisition setup prevents ambiguities in the labels, which in turn allows flawless algorithms to achieve an AP of 100%. We showed how the high-quality annotations can easily be utilized for artificial data augmentation to significantly boost the performance of the evaluated methods from an AP of 49.5% and 45.6% to 79.9% and 72.5%, respectively. Overall, the benchmark results indicate a significant room for improvement of the current state-of-the-art. We believe the dataset will help to boost research on instance-aware segmentation and leverage new approaches for artificial data augmentation.

Acknowledgements

We want to thank the students Clarissa Siegfarth, Bela Jugel, Thomas Beraneck, Johannes Köhne, Christoph Ziegler and Bernie Stöffler for their help to acquire and annotate the dataset.

References

1. Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., Vijayanarasimhan, S.: Youtube-8m: A large-scale video classification benchmark. CoRR **abs/1609.08675** (2016), https://arxiv.org/abs/1609.08675 2
2. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3213–3223 (2016). https://doi.org/10.1109/CVPR.2016.350 2
3. ECRS: RAPTOR. https://www.ecrs.com/products/point-of-sale-pos/accelerated-checkout/, accessed 2018-03-07 4
4. Everingham, M., Eslami, S.M.A., Gool, L.J.V., Williams, C.K.I., Winn, J.M., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision **111**(1), 98–136 (2015). https://doi.org/10.1007/s11263-014-0733-5 2, 6, 7, 11
5. Follmann, P., Böttger, T.: A rotationally-invariant convolution module by feature map back-rotation. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 784–792 (2018). https://doi.org/10.1109/WACV.2018.00091 5
6. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The KITTI dataset. International Journal of Robotics Research **32**(11), 1231–1237 (2013). https://doi.org/10.1177/0278364913491297 2
7. George, M., Floerkemeier, C.: Recognizing products: A per-exemplar multi-label image classification approach. In: European Conference of Computer Vision (ECCV). pp. 440–455. Springer (2014). https://doi.org/10.1007/978-3-319-10605-2_29 2
8. Gurumurthy, S., Kiran Sarvadevabhatla, R., Venkatesh Babu, R.: DeLiGAN: Generative adversarial networks for diverse and limited data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 166–174 (2017). https://doi.org/10.1109/CVPR.2017.525 6
9. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask R-CNN. In: IEEE International Conference on Computer Vision (ICCV). pp. 1059–1067 (2017). https://doi.org/10.1109/ICCV.2017.322 2, 10
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90 10
11. Huang, X., Li, Y., Poursaeed, O., Hopcroft, J., Belongie, S.: Stacked generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5077–5086 (2017). https://doi.org/10.1109/CVPR.2017.202 6
12. ITAB: HyperFLOW. https://itab.com/en/itab/checkout/self-checkouts/, accessed 2018-03-07 4
13. Jund, P., Abdo, N., Eitel, A., Burgard, W.: The freiburg groceries dataset. CoRR **abs/1611.05799** (2016), https://arxiv.org/abs/1611.05799 2
14. Koubaroulis, D., Matas, J., Kittler, J.: Evaluating colour-based object recognition algorithms using the SOIL-47 database. In: Asian Conference on Computer Vision. p. 2 (2002) 2
15. Lai, K., Bo, L., Fox, D.: Unsupervised feature learning for 3d scene labeling. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). pp. 3050–3057. IEEE (2014). https://doi.org/10.1109/ICRA.2014.6907298 2

16. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view RGB-D object dataset. In: 2011 IEEE International Conference on Robotics and Automation (ICRA). pp. 1817–1824. IEEE (2011). https://doi.org/10.1109/ICRA.2011.5980382 2
17. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015). https://doi.org/10.1038/nature14539 2
18. Li, J., Liang, X., Wei, Y., Xu, T., Feng, J., Yan, S.: Perceptual generative adversarial networks for small object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1222–1230 (2017). https://doi.org/10.1109/CVPR.2017.211 6
19. Li, Y., Qi, H., Da, J., Ji, X., Wei, Y.: Fully convolutional instance-aware semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2359–2367 (2017). https://doi.org/10.1109/CVPR.2017.472 2, 10
20. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017). https://doi.org/10.1109/CVPR.2017.106 10
21. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal Loss for Dense Object Detection. In: IEEE International Conference on Computer Vision (ICCV) (2017). https://doi.org/10.1109/ICCV.2017.324 10
22. Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: European Conference of Computer Vision (ECCV). pp. 740–755 (2014). https://doi.org/10.1007/978-3-319-10602-1_48 2, 6, 7, 11
23. Merler, M., Galleguillos, C., Belongie, S.: Recognizing groceries in situ using in vitro training data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–8 (2007). https://doi.org/10.1109/CVPR.2007.383486 2
24. Minervini, M., Fischbach, A., Scharr, H., Tsaftaris, S.A.: Finely-grained annotated datasets for image-based plant phenotyping. Pattern Recognition Letters **81**, 80–89 (2016). https://doi.org/10.1016/j.patrec.2015.10.013 2
25. Neuhold, G., Ollmann, T., Rota Bulo, S., Kontschieder, P.: The mapillary vistas dataset for semantic understanding of street scenes. In: IEEE International Conference on Computer Vision (ICCV). pp. 4990–4999 (2017). https://doi.org/10.1109/ICCV.2017.534 2
26. Redmon, J., Farhadi, A.: Yolo9000: Better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017). https://doi.org/10.1109/CVPR.2017.690 4
27. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(6), 62–66 (2017). https://doi.org/10.1109/TPAMI.2016.2577031 2, 10
28. Rennie, C., Shome, R., Bekris, K.E., De Souza, A.F.: A dataset for improved RGBD-based object detection and pose estimation for warehouse pick-and-place. IEEE Robotics and Automation Letters **1**(2), 1179–1185 (2016). https://doi.org/10.1109/LRA.2016.2532924 2
29. Richtsfeld, A., Mörwald, T., Prankl, J., Zillich, M., Vincze, M.: Segmentation of unknown objects in indoor environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4791–4796. IEEE (2012). https://doi.org/10.1109/IROS.2012.6385661 2

30. Rocha, A., Hauagge, D.C., Wainer, J., Goldenstein, S.: Automatic fruit and vegetable classification from images. Computers and Electronics in Agriculture **70**(1) (2010). https://doi.org/10.1016/j.compag.2009.09.002 2
31. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Li, F.: ImageNet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y 2
32. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**, 640–651 (2015). https://doi.org/10.1109/TPAMI.2016.2572683 1, 2
33. Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 761–769 (2016). https://doi.org/10.1109/CVPR.2016.89 11
34. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learning from simulated and unsupervised images through adversarial training. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2107–2116 (2017). https://doi.org/10.1109/CVPR.2017.241 6
35. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: IEEE International Conference on Computer Vision (ICCV). pp. 5907–5915 (2017). https://doi.org/10.1109/ICCV.2017.629 6
36. Zhou, B., Khosla, A., Lapedriza, À., Torralba, A., Oliva, A.: Places: An image database for deep scene understanding. CoRR **abs/1610.02055** (2016), `http://arxiv.org/abs/1610.02055` 2
37. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ADE20K dataset. CoRR **abs/1608.05442** (2016), `http://arxiv.org/abs/1608.05442` 2, 9
38. Zhou, Y., Ye, Q., Qiu, Q., Jiao, J.: Oriented response networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4961–4970 (2017). https://doi.org/10.1109/CVPR.2017.527 5