这篇论文主要提出了一种倾斜检测的方法，用倾斜的anchor代替之前的anchor，maybe useful in ocr or avp project

Oriented Boxes for Accurate Instance Segmentation

Patrick Follmann
Research
MVTec Software GmbH
Munich, Germany
orcidID 0000-0001-5400-2384

Rebecca König
Research
MVTec Software GmbH
Munich, Germany

***Abstract*—State-of-the-art instance-aware semantic segmentation algorithms use axis-aligned bounding boxes as an intermediate processing step to infer the final instance mask output. This leads to coarse and inaccurate mask proposals due to the following reasons: Axis-aligned boxes have a high background to foreground pixel-ratio, there is a strong variation of mask targets with respect to the underlying box, and neighboring instances frequently reach into the axis-aligned bounding box of the instance mask of interest.**

In this work, we overcome these problems and propose using oriented boxes as the basis to infer instance masks. We show that oriented instance segmentation leads to very accurate mask predictions, especially when objects are diagonally aligned, touching, or overlapping each other. We evaluate our model on the D2S and Screws datasets and show that we can significantly improve the mask accuracy by 7% and 11% mAP (14.9% and 27.5% relative improvement), respectively.

***Index Terms*—instance segmentation, oriented box detection**

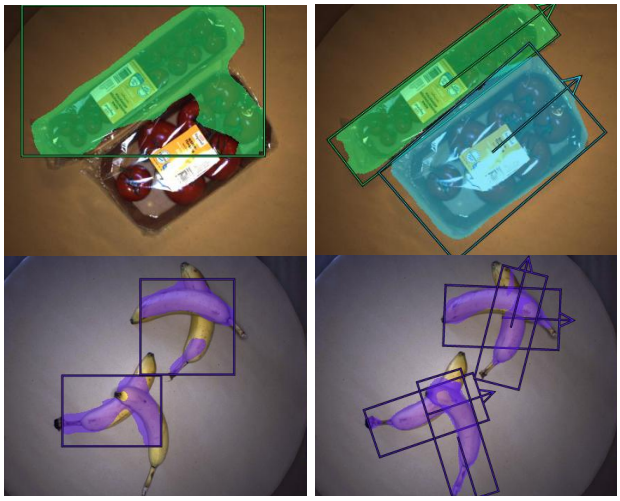
I. Introduction

The precise localization of objects in natural images is fundamental for many industrial tasks, such as bin-picking or object counting. The detection is often done using axis-algined boxes [19], [25]. However, if objects are deformable, articulated, or diagonally oriented, axis-aligned bounding boxes are often only a very coarse approximation of the objects' locations. Instance-aware semantic segmentation (instance segmentation), as introduced in [20], tries to overcome this limitation by predicting a pixel-precise mask for each of the instances. However, recent instance segmentation methods that are at the top of the COCO dataset instance segmentation leaderboard,[1] such as FCIS [17] or the many variants of Mask RCNN [10], rely on axis-aligned box proposals to infer the instance mask per box.

This intermediate axis-aligned box detection step introduces several limitations to the final instance mask output:

Depending on the object's orientation, a majority of the box covers the background or another instance that is not of interest for the mask prediction. As features are pooled with respect to the box, this can lead to false classifications or mask predictions reaching into neighboring objects (e.g. top row example in Fig. 1).

If the object is rotated, the bounding box aspect-ratio can vary significantly. This leads to highly varying mask targets with respect to the bounding box, even for very precise box-predictions.



(a) axis-aligned (b) oriented (*ours*)

Fig. 1: **Benefits of oriented instance segmentation.** (a) result based on axis-aligned boxes, (b) result of our proposed method based on oriented boxes. Oriented boxes contain fewer background pixels and avoid reaching into neighboring instances. Because oriented boxes overlap less, they are not mistakenly filtered out by NMS. Resulting instance masks are much more accurate.

Objects with non-overlapping masks can have significantly overlapping axis-aligned boxes. As a result, one of the candidate boxes may be filtered out by the non-maximum-suppression (NMS) if parameters are not tuned in a very conservative manner.

If the instance is oriented diagonally, the relatively coarse resolution that is used for mask prediction (typically, 14×14 for pooling and 28×28 for prediction) can lead to inaccurate mask boundaries due to interpolation artifacts, especially for large objects. The use of finer grids could resolve this issue to some extent, but comes at the cost of higher computation time and memory consumption.

Böttger *et al.* [2] have shown that the best possible intersection over union (IoU) an oriented bounding box can achieve for an arbitrary mask is typically much higher than the IoU of the axis-aligned bounding box. In this work, we make use of

[1] http://cocodataset.org/#detection-leaderboard

问题：背景信息较多，可能会造成分类错误或分割错误

this fact and propose using oriented as opposed to axis-aligned bounding boxes to infer instance masks since this solves the aforementioned issues:

- Independent of the objects orientation, most of the mask overlaps with the instance of interest. This increases the mask-to-background ratio for mask targets and avoids large overlaps with neighboring objects.
- If the object is rotated, the bounding box aspect ratio remains constant and the mask exhibits significantly smaller variations with respect to the pooling grid such that the training is better conditioned.
- For objects with non-overlapping masks, oriented bounding boxes overlap significantly less than axis-aligned bounding boxes. This avoids erroneously filtering out candidate boxes by NMS and makes it easier to tune NMS hyperparamters.
- Inaccurate mask boundaries, especially for large, elongated and diagonally aligned objects are avoided because long edges are aligned with the oriented box.

The main contributions of this paper are: We propose to predict accurate instance masks based on oriented box detections. Our approach can be easily applied to existing models based on axis-aligned boxes to enhance their performance. We describe the adapted architecture for oriented instance segmentation and explain necessary changes to different parts of the model compared to a baseline axis-aligned instance segmentation model. Our evaluation on two datasets *D2S* [6] and *Screws* [26] shows that the mask accuracy is improved especially for high IoU-thresholds. This leads to a significant increase in overall mAP from 47% to 54% on *D2S* and 40% to 51% on *Screws*, which is a relative improvement of 14.9% and 27.5%, respectively. Moreover, we show that on *Screws* the predicted mask output of our model can be directly used to further refine the oriented box output and improve the box mAP by 2.6%. This is in line with the results on *D2S*, where the smallest axis-aligned boxes of the predicted masks of our model outperforms the baseline that was specifically trained for this task. The improvements are summarized in Fig. 1.

II. Related Work

a) Instance segmentation.: To date, most instance segmentation methods are based on a Faster RCNN [25] two-stage object detector, where the first, region proposal network (RPN) stage proposes axis-aligned class-agnostic boxes at all locations where an object is likely to be found. The second stage pools box-specific features with a region of interest (RoI) pooling and refines them to classify the proposals into one of the target classes or background. A second, parallel branch (sometimes with shared weights) is used to further improve the box coordinates via bounding-box regression. In Mask RCNN, He *et al*. [10] propose to pool features once again, typically with a larger grid size of 14×14, for a third parallel branch that predicts the instance mask. The architecture of the mask prediction branch is similar to the decoder of a fully convolutional network for semantic segmentation [23]. It consists of a number of intermediate convolutions before a transposed convolution upsamples the features by a factor of 2 in each dimension and finally mask probabilites are predicted using a sigmoid activation. PA-Net [22] optimizes the information flow within Mask RCNN by fusing the features of several feature pyramid network (FPN) [18] levels to improve the quality, but at the cost of runtime. Mask Scoring R-CNN [12] learns to predict the quality of the mask predictions, which improves the mAP by recalibrating the predicted scores such that low quality masks also receive a lower score compared to higher quality masks. Among the more recent methods, YOLACT [1] improves the speed of instance segmentation by using a linear combination of prototypes for mask prediction and a fast variant of non-maximum-suppression. Shape priors are also used in ShapeMask [15], mainly with the goal to improve the generalizability of the model and to reduce the amount of necessary annotated training data. In contrast to our method, all of the existing instance segmentation methods rely on axis-aligned bounding boxes.

b) Oriented box detection.: The idea of oriented box detection has been introduced in the context of scene text detection. The existing methods are all based on Faster RCNN [25]. Jiang *et al*. [14] still use an axis-aligned box RPN but infer an oriented final box output by regressing the orientation. In [24], Ma *et al*. extend the RPN to have oriented anchors and the RoI pooling is extended to pool features with respect to oriented boxes. Oriented box detection has been demonstrated to perform well on other domains than in OCR. For example, Ding *et al*. [5] have set a new baseline on the oriented object detection dataset DOTA [27]. In contrast to our work and [24], they do not use oriented anchors but propose a RoI Transformer module before pooling with respect to the oriented box. Another application that benefits from the use of oriented boxes is ship detection in satellite images [28], where the authors propose a dense FPN with oriented anchors.

In this work, we use a similar architecture to Mask RCNN [10], but replace the axis-aligned box RPN by an oriented box RPN as in [24]. We incorporate the idea of [5] to pool features with respect to the oriented box proposals. This ensures that less features are pooled from the background and neighboring objects. In comparison to [10], we use the final boxes instead of the RPN boxes for pooling the mask features, as for oriented boxes it is crucial that the mask prediction fits to the final output box orientation. Moreover, none of the previous works has combined oriented box detection with the prediction of instance masks.

In comparison to previous oriented object detection methods, we explicitly allow to set *classes without orientation*: Since in most applications not all classes have a well-defined orientation we include the option that the model falls back to an axis-aligned box for some user-set classes, which stabilizes the training.

III. Oriented Instance Segmentation

In the following, we present the key differences of instance segmentation based on oriented boxes. The changes enable

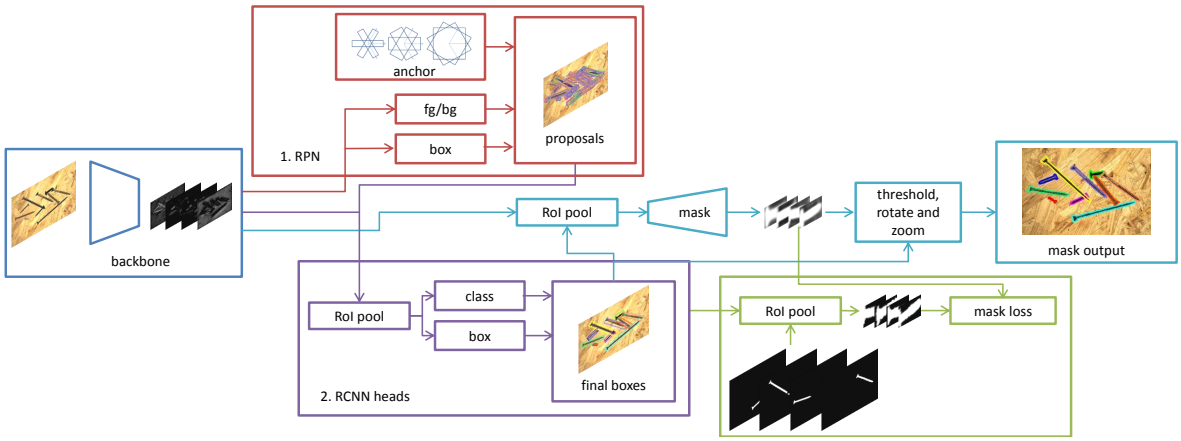


Fig. 2: **Architecture:** (blue, center left) input image and backbone for feature extraction, (red, top) oriented RPN based on oriented anchors. (violet, bottom) features are pooled with respect to oriented proposals to feed FRCNN heads for classification and second box proposal refinement. This results in the final oriented box output. (cyan, middle to top-right) features are pooled with respect to final boxes to feed the branch for mask prediction. Finally, mask probabilites are zoomed, rotated to fit to the oriented box output and thresholded. (green, bottom-right) During training, mask targets are calculated by RoI pooling the ground truth masks.

more accurate mask predictions than instance segmentation based on axis-aligned bounding boxes.

An overview of our oriented instance segmentation (*OIS*) architecture is depicted in Fig. 2. The network is based on Mask RCNN [10] with a feature pyramid network (FPN) [18] and enhances some of the ideas that were developed in [5], [14], [21], [24]. In a first step, the backbone is to applied to the input image to extract features that are used for the three following stages: The first stage is the RPN, which predicts for each of a number of template *anchor boxes* whether it is likely that an object with similar bounding box is present or not (fg/bg branch) and if so, how the anchor should be refined to better match the underlying object (box branch). The second stage are the RCNN heads, where the box proposal outputs of the RPN are used to RoI pool the features for class prediction (class branch) and further box refinement (box branch). The third stage uses the final RCNN head box output to again RoI pool the features to feed the mask branch that outputs a pixel-precise mask for each of the final boxes.

a) Box representation.: We use a five parameter representation for boxes $b = (r, c, l1, l2, \phi)$, where (r, c) are the subpixel-precise center point row and column coordinates, $(l1, l2)$ are the semi-axes lengths of the oriented box and ϕ is the orientation pointing in the direction of the major axis. ϕ is given as an angle in radians between the positive horizontal axis and the $l1$-axis in mathematically positive sense. In contrast to the parametrization of [24], we do not enforce that $l1 \geq l2$ such that sudden flips of the orientation for boxes with aspect ratio close to one are avoided.

We allow two different scenarios: The first scenario is that we are interested in the exact orientation of boxes, i.e., $\phi \in (-\pi, \pi]$. In the second scenario, ϕ is limited to the range $(-\frac{\pi}{2}, \frac{\pi}{2}]$, which is suitable if the actual orientation of the box is not important, but only a tight oriented bounding box of

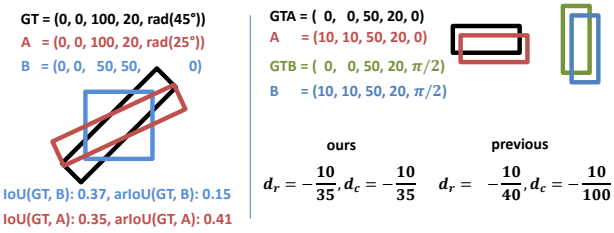


Fig. 3: **arIoU and box target example.** (left): (black) ground truth box, (red, blue) oriented anchors. Although, for the blue anchor the coordinates $l1, l2$, and ϕ differ a lot, the exact IoU is higher than for the red anchor where only ϕ differs by 20°. This issue is solved by the use of the arIoU for anchor assignment. (right) Our box center targets ensure that the targets are consistent for anchors with different orientation but the same center-coordinates.

the instance mask is the goal. In this case, predictions with ϕ outside of the range can be corrected by subtracting or adding π. To differentiate the two scenarios, we use a parameter *IgnoreDirection* that is set to *true* in the latter case. Note that it is important that also the ground truth given in the dataset at hand is annotated in accordance to *IgnoreDirection*.

b) RPN.: Because we want to predict oriented proposal boxes, the region proposal network is fed with oriented anchors with different orientations, aspect ratios, and subscales. Ding *et al.* [5] argue that this leads to a massively higher number of anchors compared to axis-aligned boxes. However, we choose the orientation of the ground truth boxes such that it is aligned with the longer box side-length. Therefore, only aspect ratios smaller or equal to one are necessary, which almost halves the number of anchors. Moreover, if one is

没太明白这个地方讲的是什么?

也就是只用纵横比小于1的gt来监督

角度范围再砍一半

not interested in the exact orientation of the box in the range $(-\pi, \pi]$, but only wants the angle to be exact modulo π (range $(-\frac{\pi}{2}, \frac{\pi}{2})$), three orientations such as $(-\frac{\pi}{3}, 0, \frac{\pi}{3})$ are sufficient in most situations.

IOU最大不一定是最佳match

As usual, the anchor target assignment is based on the IoU between anchors and ground truth. However, the exact oriented box IoU sometimes leads to inefficient assignments (cf. Fig. 3). Hence, we use the angle-related IoU (arIoU [21]) to judge whether an anchor is assigned to background or foreground:

余弦相似度

$$\text{arIoU}(A, B) = \max(0, cos(\phi_a - \phi_b)) \cdot \text{IoU}(\hat{A}, B), \quad (1)$$

where $\hat{A}$ is the box $(r_A, c_A, l1_A, l2_A, \phi_B)$.

Although the arIoU can fix the problem of wrong anchor assignment, its values can become small for anchors that have only slightly different parameters than the ground truth box. With the default settings of IoU-thresholds in the anchor assignment ($fgPosThresh = 0.5$, $fgNegThresh = 0.4$), this can lead to a very low number of foreground anchors.

跟之前一样如果一个gt没有anchor，则assign一个最大的

Therefore, we change the anchor assignment as follows: If an anchor has an arIoU larger than *fgPosThresh* with a ground truth box, it is assigned to the foreground. If the arIoU is below *fgNegThresh* and there is another anchor that has a higher IoU with this ground truth, we assign it to the background. However, if the arIoU is larger than zero and it is the highest arIoU that was achieved by any anchor for this ground truth box, we assign this anchor as foreground. This is in contrast to the original Faster RCNN anchor assignment as implemented, e.g. in [9], where the anchor is only assigned to foreground if its IoU is higher than *fgNegThresh*. This change leads to a more stable training because enough foreground examples are present to train the box and class branches of the RPN.

也是RPN强制assign，head不用

The presented relaxation of the *fgNegThresh* threshold is only used in the RPN. For the RCNN heads and the mask head, we only want good box proposals to contribute to the training. This is in line with the findings of [3], where the assignment thresholds are increased in several iterative box refinements.

c) Oriented box targets.: The correct definition of training targets is a crucial incredient for oriented box detection, as minor modifications can have a large impact on the training convergence.

In comparison to [24], we use a slightly different box target calculations for the row and column coordinate deltas:

$$d_r = \frac{r^* - r}{\bar{l}}, \; d_c = \frac{c^* - c}{\bar{l}}, \quad (2)$$

where (r^*, c^*) are coordinates of the ground truth box, (r, c) are coordinates and $\bar{l} = (l1 + l2)/2$ is the mean axis length of the box for which the deltas should be calculated. Using $\bar{l}$ for normalization is beneficial because there is no dependency on the orientation of the box. Others, such as [5], [24], use $l2$ to normalize d_r and $l1$ to normalize d_c instead of the mean value $\bar{l}$. This is similar to the original target normalization of [8] for axis-aligned boxes with $\phi = 0$. However, if a box is oriented with $\phi = \frac{\pi}{2}$ the meaning of $l1$ and $l2$ with respect to r and c is flipped. This is avoided by the use of $\bar{l}$. See Fig. 3

用长款的均值normalize，而不是wh分别normalize

右边的是本文的这种mask，包含的背景信息更少，所以28*28的feature更好用。

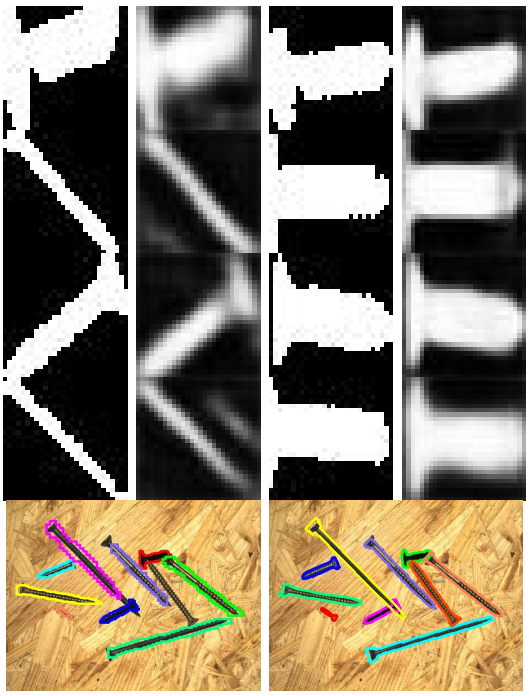


Fig. 4: **Comparison of mask targets.** (top) randomly chosen mask targets and mask probabilities, in the left two columns for axis-aligned boxes, in the right two columns for oriented boxes, respectively. (bottom) Final mask output for axis-aligned boxes and oriented boxes on a generated *Screws* image (left and right). Note that the low resolution (28×28) of mask targets and predictions is used much more efficiently for oriented boxes, where mask targets are more detailed. Moreover, the targets are much more consistent even for totally different instances. This simplifies both the training and the prediction.

Moreover, in the calculation of the target delta for the orientation d_ϕ, we avoid large regression targets by ensuring that d_ϕ is in the range $(-\frac{\pi}{2}, \frac{\pi}{2})$ due to the use of the arIoU during the assignment phase. In practice, this worked best if anchor orientations are chosen such that intermediate angles to the ground truth are not larger than $30°$.

d) Oriented mask prediction.: As mentioned above, and in contrast to the original Mask RCNN implementation [10], we use the final oriented box predictions of the RCNN heads to pool features once again from the backbone and feed them into the mask branch. This ensures that the mask predictions are consistent with the box outputs, such that a simple rotation and zooming with respect to the box parameters is sufficient to fit them into the input image.

cascade

The assignment of final oriented boxes to the ground truth instances is done in the same way as the assignment for proposal targets based on the arIoU. In this separate assignment step, we increase the IoU thresholds to $fgNegThres = 0.5$ and $fgPosThresh = 0.7$ such that only good final boxes contribute to the training of the mask branch. If a final box is not assigned to any ground truth instance, the corresponding weights in the

mask prediction loss are set to zero such that the predicted mask is ignored.

We incorporate the mask target generation directly into the model such that it can be done efficiently on the GPU during training. Therefore, we create a binary image where we paint the ground truth masks into zero-initialized channels one-by-one (cf. Fig. 2). The number of channels can be pre-calculated as the maximum number of instances per image. To obtain the cropped mask targets, another oriented single-channel RoI pooling operation on the ground mask image is done. We adapt the RoI pooling operation to pool from the assigned single channel of the ground truth mask image for each instance.

As in [10], a final grid size of 28×28 pixels for mask targets and prediction is used. This low resolution limits the detailedness of the output masks. However, as the oriented boxes generally contain much less background than the axis-aligned boxes, the given resolution is used much more efficiently. This is visualized in Fig. 4. The batch-size of the mask branch equals the input batch size times the maximal number of predictions. Therefore, a higher mask resolution comes at the cost of much higher computation time and memory consumption and should be avoided.

Mask prediction is done class-agnostically since it has already been shown in [10] that a class-specific mask prediction does not improve the results significantly.

e) Classes without orientation.: In most applications, not only objects with a clearly defined orientation are of interest. If we use the smallest oriented bounding box, the orientation of ground truth instances is varying without a hint for the model why this is the case. This can lead to a severe destabilization of the training. Therefore, we propose to assign these classes to a group of *classes without orientation*.

For these categories, we annotate the ground truth boxes as the smallest axis-aligned bounding boxes of the ground truth instance masks with $\phi = 0$.

IV. Experiments

In the following subsections, we evaluate our *OIS* model on two different datasets: namely the *D2S* [6] and *Screws* [26] datasets. Both datasets have high-quality annotations such that a potentially improved mask accuracy can be measured. Additionally, they contain oriented categories as well as *categories without orientation*, such that it can be shown that *OIS* can handle both types at the same time.

To measure the potential benefit of using oriented instance segmentation compared to axis-aligned instance segmentation, we compute the IoU of the smallest oriented bounding box and ground truth mask as well as the IoU of the smallest axis-aligned bounding box and ground truth mask for each instance. The mBMIoU averaged per class mean summarizes this for the whole dataset:

$$\text{mBMIoU} = \frac{1}{C} \sum_{c=1}^{C} \frac{1}{N_c} \sum_{i=1}^{N_c} \text{IoU}(B_i, M_i), \tag{3}$$

where C is the number of classes, N_c is the number of instances per class and $\text{IoU}(B_i, M_i)$ is the intersection over union of the bounding box B_i with the ground truth mask M_i of instance i. The comparison of mBMIoU values in Table I indicates how tighter the oriented boxes are around the instances on average. For both datasets, the difference is significant and in the following we show that this potential can indeed be utilized.

dataset	axis-aligned	oriented
D2S	62%	81%
Screws	46%	58%

TABLE I: **Mean box mask IoU mBMIoU.** Mean IoU of smallest bounding box with mask computed for each instance of a class and averaged over all classes (3).

A. D2S

The densely segmented supermarket dataset (*D2S*) [6] contains 60 different categories of supermarket products lying on a turntable and captured in a top-down view with various orientations. There are elongated objects such as lying bottles, boxes, certain vegetables like carrot, cucumber, or zucchini, but also round or highly deformable objects such as standing bottles, apples, or nets filled with oranges.

Because the instances are touching or overlapping in many cases, the dataset is very challenging. Moreover, difficult scenes are only contained in the validation and test splits, which requires data augmentation to enhance the training set.

In *D2S*, each image is captured with three different lightings (normal, bright, and dark), but our focus is to show that *OIS* improves the mask quality compared to the axis-aligned instance segmentation baseline (*AAIS*). Hence, we only use the normal lighting images. Further, we scale the images to a relatively low resolution of 512×384 to speed up the training and evaluation and to reduce the energy consumption of our experiments.

a) Data preparation and model settings.: To have a fair comparison, we use exactly the same training data for all models. As mentioned, for *D2S* it is necessary to generate additional training data by utilizing the existing annotations of the training set. We follow the approach of [7] and crop random instances from the training set using their ground truth masks before pasting them onto random backgrounds that look similar to the ones of the validation and test set. This way, the training set can be enlarged at almost no additional cost. 2000 additional augmented training images are generated with random backgrounds and randomly positioned and rotated instances. The dataset consists of 9000 images in total: 2000 augmented images plus 1960 original training images, 1200 validation images and 4340 test images. Examples for generated images are given in the supplementary material. For *D2S*, we assign the following classes to *classes without orientation*: *apple_golden_delicious, apple_granny_smith, apple_red_boskoop, clementine, clementine_single, orange_single, oranges, lettuce, salad_iceberg*. By their nature, these categories have no well-defined orientation. We set *IgnoreDirection* to *true* for *D2S* experiments.

Fig. 5: ***D2S* results.** *(from left to right)* input image with ground truth masks and generated ground truth oriented boxes, results based on axis-aligned boxes, results based on oriented boxes. The results based on oriented boxes are much more precise, especially for elongated, diagonally oriented objects. More results can be found in the supplementary material.

For the training, validation, and test splits, the box ground truth for *OIS* is generated as follows: We use the smallest oriented bounding box of the instance mask by default, except for *classes without orientation*, where the smallest axis-aligned bounding box is used. Moreover, for instances, where the IoU of the smallest axis-aligned bounding box with the smallest oriented bounding box is higher than 0.95 and at the same time the roundness (cf. appendix) of the mask is higher than 0.95, we annotate the instance with the smallest axis-aligned bounding box. The same applies to the augmented images. However, because instances are frequently occluded, we use the orientation of the smallest oriented bounding box of the amodal mask [16] and fit the smallest bounding box with that orientation to the final, possibly occluded mask. This helps to obtain more consistent orientations and stabilizes the training.

We use the same network architecture for *AAIS* and *OIS*. For *AAIS*, we replace oriented anchors with axis-aligned anchors and use the default axis-aligned RoI pooling operations. Therefore, the intermediate box proposals generated by the RPN and the final box outputs generated by the RCNN heads are axis-aligned for *AAIS*. All other model hyperparameters are the same as for *OIS*. Due to the high complexity of the dataset, we use a ResNet-50 [11] backbone that was pretrained on ImageNet [4]. Note that we train all other weights of the RPN, the RCNN, and mask heads from scratch. We train the model for 50 epochs and evaluate after each epoch on the validation set (using box mAP). The results are always shown for the best model regarding the validation set mAP (early stopping). All other training and model parameters can be found in the appendix.

model	AABB	OBB	mask	mask agn
AAIS	49%	-	47%	55%
OIS	**55%***	46%	**54%**	**64%**

TABLE II: ***D2S* quantitative results.** mAP values on the test set. AABB: axis-aligned box, OBB: oriented box, mask: instance mask, mask agn: same as mask, but without evaluating the class prediction. *OIS* clearly improves the mask mAP. (*) axis-aligned boxes are predicted from masks.

b) Analysis.: Fig. 5 shows qualitative results on *D2S* test images. In comparison to *AAIS*, the *OIS* mask predictions are much more accurate, especially for diagonally oriented objects. Due to the less precise mask targets in *AAIS*, also the mask probabilities become inexact. In combination with the upscaling to the final box size and thresholding, the mask output is often not exactly in correspondence with the instance boundaries. In case of overlapping or touching objects, *OIS* clearly shows better results as in those cases the box overlap is lower. In many cases for *AAIS*, the mask is extended into the background or onto neighboring objects.

Generally, if the box is predicted correctly, the mask is very precise for *OIS*. This is a clear improvement to *AAIS*, which also reflects in mAP-values shown in Table II and Fig. 6: Especially for high IoU-thresholds, the mAP can be significantly improved by *OIS*: E.g., from 58% to 65% @IoU0.75 and from 23% to 43% @IoU0.85.

Since *AAIS* predicts axis-aligned boxes, we can only evaluate the box mAP for the axis-aligned ground truth boxes. Interestingly, if we use the axis-aligned smallest bounding boxes of the instance masks predicted by *OIS*, the mAP of 55% clearly outperforms the axis-aligned box mAP of *AAIS*. Moreover, in contrast to *AAIS*, for *OIS* the mask mAP is even higher than the oriented box mAP, which shows that the mask prediction is very precise. Since our focus is to improve mask accuracy, we also show results where the predicted class is not taken into account (class-agnostic evaluation). Also in this case, the mask mAP can be improved substantially.

B. *Screws*

The *Screws* [26] dataset contains 9 different types of screws and 4 different types of nuts on a wooden background. The categories differ in size, length, and color, but some are only distinguishable by a different thread. Typical example images are shown in Fig. 7.

The dataset consists of 384 images, of which 70% belong to the train set, 15% to the validation set, and 15% to the test set. Ground truth annotations are given as oriented

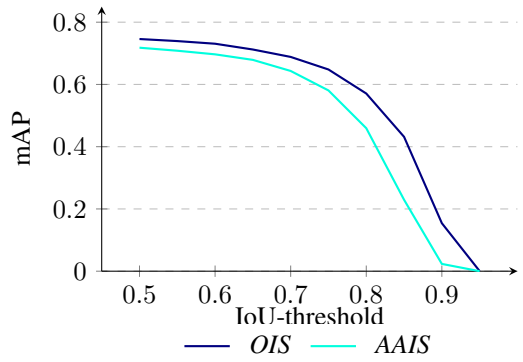


Fig. 6: **mAP on *D2S*.** Mask mAP-values for IoU-thresholds from 0.5 to 0.95. Consistently and especially for high IoU-thresholds *OIS* improves the mAP significantly.

model	AABB	OBB	mask
generated data			
AAIS	52.3%	-	41.4%
OIS	-	56.0%	**50.7%**
real data			
AAIS box from mask	-	35.3%	-
OIS box from mask	-	44.7%	-
OIS	-	50.2%	-
OIS or from mask	-	**52.8%**	-

TABLE III: ***Screws* quantitative results.** mAP values on the test sets. AABB: axis-aligned box result, OBB: oriented box result, mask: instance mask result. For generated data *OIS* clearly improves the mask mAP. On the real dataset, the or box mAP results show that the generated masks are a lot more consistent with the instances for *OIS*.

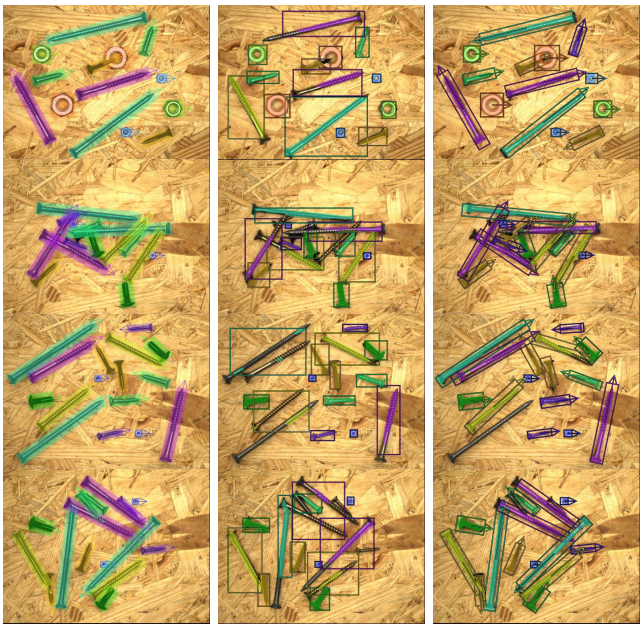


Fig. 7: ***Screws* results.** *(from left to right)* real input with ground truth boxes, results of *AAIS*, results of *OIS*. *OIS* clearly improves the results, especially for close-packed objects. Models were trained only on generated data.

boxes. As the nut-classes are symmetric and the orientation is not well-defined, these four classes are assigned to *classes without orientation*. In this dataset for screw classes, the exact orientation pointing from the screws head to its tail is of interest. Therefore, *IgnoreDirection* is set to *false* such that box orientations are in the range $(-\pi, \pi]$.

a) Generated training data.: Because the pixel-precise annotation of instance masks is tedious and time-consuming, we follow the approach of [26] to generate artificial training images. Therefore, each category is captured on a homogeneous white background where a relatively precise instance mask can be obtained with by thresholding. In this weakly-supervised setting, we use a single template image per category and generate 1300 images (700 train, 300 validation, 300 test) by cropping and pasting random instances onto empty wooden backgrounds similar to those of the original dataset. A generated example image is shown in Fig. 4.

To train and evaluate *AAIS*, we obtain the box ground truth annotations as the smallest axis-aligned bounding box of instance masks. The oriented box ground truth for categories with orientation is generated as follows: The orientation of the box is calculated based on the second moments of the instance mask (cf. appendix). To get the orientation pointing from the screws head to its tail, the orientation is corrected by adding or subtracting π if the orientation from the center of gravity to the most distant point on the mask boundary is pointing into the opposite direction.

b) Model settings.: For the evaluation, we train all models only on the generated training set. Because *Screws* has less intra-class variations and less variations due to changes in perspective, the dataset is less complex than *D2S*. Therefore, in *Screws* experiments we use a SqueezeNet [13] backbone, which speeds up the training process. This also shows that the proposed *OIS* model is superior independent of the backbone or the many other hyperparameters. Details on the remaining parameters are given in the appendix.

c) Evaluation on generated data.: First, we evaluate *AAIS* and *OIS* on the generated test set. The quantitative results are shown in Table III. For the generated test set the box mAP is very similar for *AAIS* and *OIS* (52.3% vs. 56.0%). The reason why *OIS* has a higher box mAP is that for oriented detection generally the bounding boxes of overlapping instances do not overlap as much as for axis-aligned detection. Thus, on the one hand, the pooled features contain less influences from other instances. On the other hand, fewer of the correct predictions are filtered out due to NMS. Also here, the clear improvement of mask mAP (41.4% *AAIS* to 50.7% *OIS*) shows that the masks can be predicted much more precise for *OIS*.

d) Evaluation on real data.: Because the original *Screws* dataset only contains oriented box annotations, we predict instance masks and use them to calculate oriented boxes in the same way as was done for the ground truth of the generated dataset (*box from mask*). By this, we measure how well the

predicted masks are aligned with the instance. For *OIS*, we can also evaluate the predicted oriented boxes. The result of *OIS or from mask* is using the predicted box from *OIS*, where ϕ is changed to the orientation calculated from the predicted mask.

Table III shows that for *AAIS* the predicted masks do not coincide with the instances, as the box mAP of 35.3% is rather low. For *OIS*, this result is increased by a large margin to 44.7%. The result is still slightly below the box result directly predicted from *OIS*. However, if predicted masks are used to refine the orientation as in *OIS or from mask* the box results can be further improved by 2.6%.

Qualitative results are shown in Fig. 7.

V. Conclusion

In this work, we have presented an instance segmentation model that predicts instance masks based on oriented box predictions. The use of oriented boxes leads to a more precise approximation of the instance masks and results in very accurate mask predictions. We have shown that the overall mask mAP on *D2S* and *Screws* can be improved from 47% to 54% and from 41.4% to 50.7%, respectively. Moreover, we have shown that predicted instance masks can be used to improve the axis-aligned box detection on *D2S* and the oriented box detection on *Screws*.

References

[1] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. YOLACT: real-time instance segmentation. *CoRR*, abs/1904.02689, 2019. 2

[2] Tobias Böttger, Patrick Follmann, and Michael Fauser. Measuring the accuracy of object detectors and trackers. In *German Conference on Pattern Recognition*, pages 415–426. Springer, 2017. 1

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6154–6162, 2018. 4

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 6

[5] Jian Ding, Nan Xue, Yang Long, Gui-Song Xia, and Qikai Lu. Learning RoI transformer for oriented object detection in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2849–2858, 2019. 2, 3, 4

[6] Patrick Follmann, Tobias Böttger, Philipp Härtinger, Rebecca König, and Markus Ulrich. MVTec D2S: Densely Segmented Supermarket Dataset. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 581–597, Cham, 2018. Springer International Publishing. 2, 5

[7] Patrick Follmann, Bertram Drost, and Tobias Böttger. Acquire, augment, segment and enjoy: Weakly supervised instance segmentation of supermarket products. In Thomas Brox, Andrés Bruhn, and Mario Fritz, editors, *Pattern Recognition*, pages 363–376, Cham, 2019. Springer International Publishing. 5

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014. 4

[9] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. https://github.com/facebookresearch/detectron, 2018. 4

[10] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1059–1067, 2017. 1, 2, 3, 4, 5

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[12] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring R-CNN. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6409–6418, 2019. 2

[13] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and $<$ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 7

[14] Yingying Jiang, Xiangyu Zhu, Xiaobing Wang, Shuli Yang, Wei Li, Hua Wang, Pei Fu, and Zhenbo Luo. R2CNN: Rotational region CNN for orientation robust scene text detection. *arXiv preprint arXiv:1706.09579*, 2017. 2, 3

[15] Weicheng Kuo, Anelia Angelova, Jitendra Malik, and Tsung-Yi Lin. Shapemask: Learning to segment novel objects by refining shape priors. *arXiv preprint arXiv:1904.03239*, 2019. 2

[16] Ke Li and Jitendra Malik. Amodal instance segmentation. In *European Conference on Computer Vision*, pages 677–693. Springer, 2016. 6

[17] Yi Li, Haozhi Qi, Jifeng Da, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2359–2367, 2017. 1

[18] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3

[19] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 1

[20] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *European Conference on Computer Vision*, pages 740–755, 2014. 1

[21] Lei Liu, Zongxu Pan, and Bin Lei. Learning a rotation invariant detector with rotatable bounding box. *arXiv preprint arXiv:1711.09405*, 2017. 3, 4

[22] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8759–8768, 2018. 2

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 2

[24] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*, 20(11):3111–3122, 2018. 2, 3, 4

[25] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 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. 1, 2

[26] Markus Ulrich, Patrick Follmann, and Jan-Hendrik Neudeck. A comparison of shape-based matching with deep-learning-based object detection. *tm-Technisches Messen*, 86(11):685–698, 2019. 2, 5, 6, 7

[27] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3974–3983, 2018. 2

[28] Xue Yang, Hao Sun, Kun Fu, Jirui Yang, Xian Sun, Menglong Yan, and Zhi Guo. Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks. *Remote Sensing*, 10(1):132, 2018. 2