Multi-Class Object Detection Using 2D Poses

Christopher Mayershofer, Ala Hammami, Johannes Fottner
Chair of Materials Handling, Material Flow, Logistics
Technical University of Munich
Garching, Germany
{christopher.mayershofer, ala.hammami, j.fottner}@tum.de

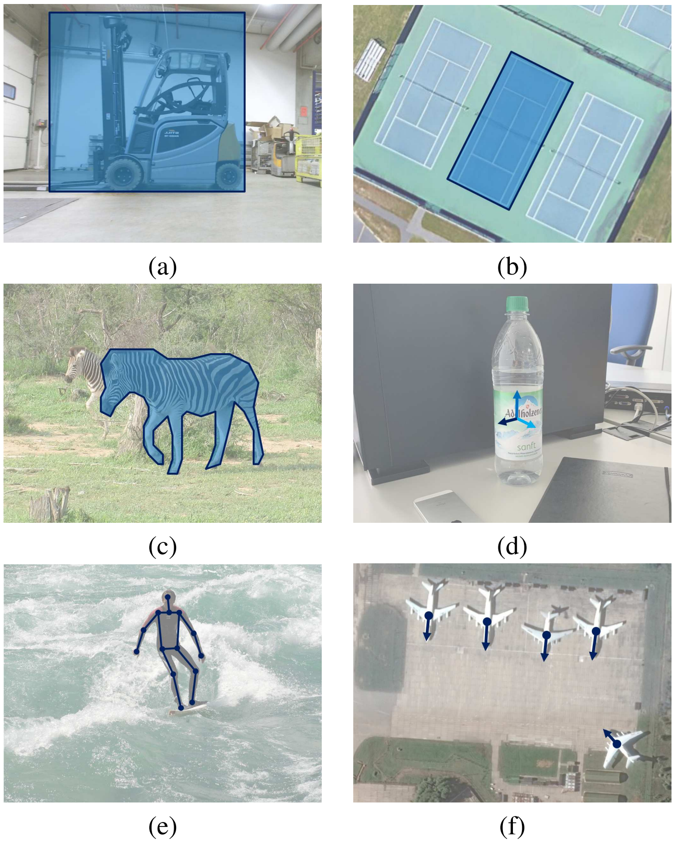
***Abstract*—Object detection (OD) methods are finding application in various fields. The OD problem can be divided into two sub-problems, namely object classification and localization. While the former aims to answer the question what class a given object belongs to, the latter focuses on locating an object within a given image. For localization, both implicit representations, which border the object and its features (e.g. bounding boxes, polygons and masks), and explicit representations, which describe the object's pose in an image (e.g. 6D pose, keypoints), are used. The 2D pose is a simple, yet effective representation that has so far been overlooked. In this paper, we therefore motivate and formulate the use of 2D poses for object localization. Furthermore, we present RetinaNet-2DP, an anchor-based convolutional neural network (CNN) that is capable of detecting objects using 2D poses. To do so, we propose the idea of Anchor Poses and the Gaussian Kernel Distance as a similarity metric between poses. Experiments on the DOTA dataset and two robotics use cases from industry emphasize the performance of the network architecture and more generally demonstrate the potential of the proposed localization representation. Finally, we critically assess our findings and present an outlook of future work.**

***Index Terms*—2D Pose, Object Detection, Computer Vision**

I. Introduction

The emergence of deep neural networks has brought about numerous advances in the field of computer vision that would not be possible with conventional methods. One field that has particularly benefited from this development is object detection (OD). OD is the process of finding objects in a given image ("Where is the object located in the image?") and their assignment to a semantic class ("What class does this object belong to?"). The problem of OD can therefore be divided into two sub-problems: the localization and classification of objects.

In this paper, we propose using 2D poses as a means of localization for multi-class object detection. The use of 2D poses for localization purposes is common in robotics tasks such as navigation of autonomous mobile robots or grasping for robotic arms. In the field of object detection, however, only bounding boxes, masks, 6D poses and keypoints are being used. To still make use of the possibilities of object detection for aforementioned use cases, the output representation is being post-processed into a 2D pose representation. Unlike OD using bounding boxes or masks, using a 2D pose representation for multi-class object detection therefore enables end-to-end learning for robotics without the need for post-processing. Moreover, compared to OD using 3D poses, the 2D pose representation is less complex due to the lack of interpretation of the 3D space in a 2D image, which both simplifies the manual annotation of real images and potentially increases the performance and robustness of trained models.



(a) (b)

(c) (d)

(e) (f)

Fig. 1. **Localization techniques for object detection.** Currently, objects are localized using Horizontal Bounding Boxes (HBBs) (a) [1], Oriented Bounding Boxes (OBBs) (b) [2] and Polygons and Masks (c) [3]. Furthermore, explicit methods are used to represent an objects location, using 6D poses (d) and keypoints (e) [3]. We propose using 2D poses (f) [2] a simple, yet effective and common way of localization in the robotics domain. Reference denotes image source. Best viewed in color.

This paper describes, formalizes and evaluates multi-class object detection using 2D poses. Our contributions can be summarized as follows:

- **Object localization using 2D poses.** We propose and formalize the idea of using 2D poses as a means of localization representation for OD systems in images.
- **2D pose multi-class object detection model.** In this

This work was supported by BMW Group.

paper, we develop an anchor-based single-shot detector based on RetinaNet [4]. To achieve this, we propose using Anchor Boxes and a Gaussian Kernel Distance as a similarity measure.

- **Evaluation and application scenarios.** We first ablate our novel concepts on the DOTA dataset. We then demonstrate the generalizability and applicability of our approach using two robotic case studies from industry.

II. Related Work

This section summarizes different localization representations and their corresponding architecture implementations in computer vision research.

Horizontal Bounding Boxes (HBB) are rectangular boxes used to border an object and its features. HBBs are always aligned parallel to the image edges. They are defined either by their center point (x, y), box height h, and width w or by specifying the left, top, right and bottom values of the boxes' corners, x_{min}, x_{max}, y_{min} and y_{max}. Due to the low amount of annotation effort required, HBBs are a common means of localization used in object detection. Convolutional Neural Network (CNN) architectures able to predict HBBs can be classified into two main categories: those that use prior information and those that do not use prior information. Anchor-based models [4], [5] learn to minimize the offset between anchor box and ground truth box, while [6], [7] use keypoints to directly predict objects. Reference [8] takes yet another anchor-free approach, using heatmaps to predict the object's corners.

Oriented Bounding Boxes (OBB) extend the idea of using a box that borders an object by additionally specifying the box's rotation. This kind of localization is often used in top-view images, such as aerial images (e.g. DOTA [2], HRSC16 [9]) and poses various challenges for standard HBB detectors [10]. To overcome these challenges, existing HBB models have been adapted to the OBB task. As before, anchor-based methods [11] as well as anchor-free methods [12] have been proposed.

Polygons and Masks are a more detailed means of localization used in various image understanding tasks. In a similar way to the box approaches referred to above, polygons and masks localize an object by bordering its features. In contrast to the former, they allow a more precise, even pixel-level object localization but require a high annotation effort. Masks are usually represented by a discrete two-dimensional map, whereas polygons are represented by a set of points. Note that each of these representations can be converted to the other respective form. Both, two-stage approaches [13] and single shot methods [14] have been proposed in the literature.

Moving on from implicit localization techniques that border an object's features, explicit localization techniques are used to locate an object by specifying its pose (i.e. position and/or orientation):

6D Poses are an explicit means of localization describing an object's six-degree-of-freedom (6D) pose in space. 6D pose estimation is used in a number of real-world applications, such as robotic grasping [15] and augmented reality [16]. Due to the nature of dimensionality reduction during the image generation process (i.e. 3D space to 2D image), accurate manual annotation of 6D poses in natural images is a difficult endeavor. Therefore, 6D pose estimation relies on simulation environments [17] or difficult ground-truth generation mechanisms to create 6D pose annotations [18]. Again, various deep learning architectures for object detection using HBBs were adopted to enable the estimation of 6D poses, e.g. [19].

Keypoints are another explicit spatial localization representation used for marking objects or features that stand out in an image, using single points or a set thereof. Keypoint representations are used in human pose estimation [3] or object keypoint estimation [18] datasets. State-of-the-art approaches for keypoint detection adopt convolutional neural networks to work for the given task of detecting keypoints, e.g. [20], [21].

To sum up, we have shown that both implicit and explicit methods can be used to solve individual application-specific problems. Whilst all of those localization techniques contain a spatial localization component, only some consider the localization orientation. To the best of our knowledge, the use of a 2D pose (i.e. center point (x, y) and orientation θ) for object detection has not yet been considered, which is why we wish to formulate, conceptualize and motivate this localization technique in this paper.

III. Problem Formulation

Let $I \in \mathbb{R}^{W \times H \times C}$ be an input image of width W, height H and channel C, with $C = 3$ for color and $C = 1$ for monochrome images. We aim to detect objects, i.e. to both classify and localize them, using a 2D pose representation. Therefore, we try to find a function F that maps the input image I to a set of object detections D:

$$F(I) = D; D = \{d_1, d_2, ..., d_n\} \tag{1}$$

Here, d_i represents the detection of object i by specifying its location p_i and class c_i:

$$d_i = (c_i, p_i) \tag{2}$$

The object's location p_i is described using a 2D pose representation, $p_i = (x_i, y_i, \Theta_i)$, where (x_i, y_i) describes the object's center point and Θ_i its orientation with respect to the image coordinate frame. Fig. 2 illustrates the problem.

IV. Multi-Class Object Detection using 2D Poses

This section presents our simple, yet effective, method of performing multi-class object detection using 2D poses. We propose an anchor-based single-shot deep convolutional neural network that utilizes 2D poses for object localization. We first describe the intuition behind two general concepts applicable to this type of object localization, namely Anchor Poses and Gaussian Kernel Distance, before detailing our RetinaNet-2DP implementation.

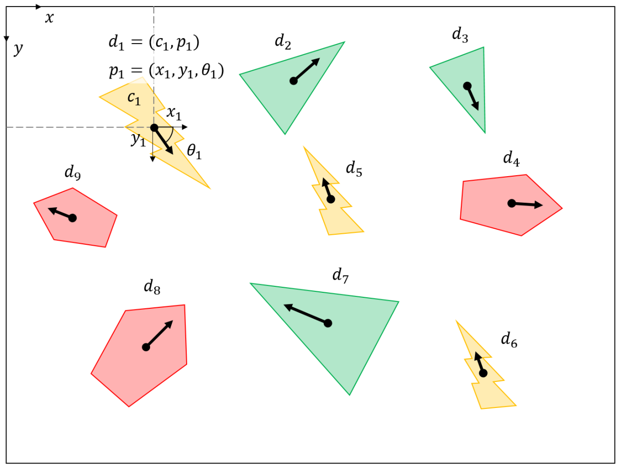


Fig. 2. **Object Detection using 2D Poses.** We are interested in detecting objects (COLORED SHAPES) in an input image I. Therefore, we aim to find a function F that maps an input image I to a set of object detections $D = \{d_1, d_2, ..., d_n\}$. Where d_i describes the ith object's location using a 2D pose representation p_i (ARROWS) and its class c_j(SHAPE GEOMETRY AND COLOR).

A. Anchor Poses

As discussed in Section II, CNNs are commonly infused with prior information in order to improve their performance. We adapt the idea of using anchors as a means of incorporating prior information to work with 2D poses, which we refer to as Anchor Poses. Instead of having to learn to directly predict an object's pose, the network is able to use the anchors as prior information and only needs to predict the offset of those anchors to best match a given object.

In a similar way to anchor boxes, Anchor Poses represent a set of predefined poses of a certain position and orientation, which can be thought of as being tiled across the input image for each image scale of the feature extractor. Due to the use of Anchor Poses, the network prediction task is simplified to only refine those prior poses to optimally localize an object. Before training, the predefined Anchor Poses generally do not overlap with the ground truth poses (i.e. there is a rotational and translational offset between ground truth and anchor). During training, our model learns the offsets to be applied to each Anchor Pose and therefore refines the position and orientation.

In order to come up with good anchors, [22] chose to manually define them, whereas [5] used k-means clustering to find the optimal parameters on a large-scale dataset. As no large-scale dataset with 2D pose annotations is available, we search for the best anchor pose parameters as further discussed in Section V.

B. Gaussian Kernel Distance

Intersection over union (IoU) is a similarity measure used for implicit localization tasks. IoU compares a ground truth shape S_{GT} with a predicted shape S_P by computing their intersection over union:

$$IoU = \frac{area(S_{GT} \cap S_P)}{area(S_{GT} \cup S_P)} \tag{3}$$

In most applications, a prediction is considered correct if the IoU is greater than or equal to a predefined threshold. When using 2D poses as a means of localization, calculating the overlap (and hence the IoU) is no longer feasible.

Therefore, we replace the IoU concept and use a non-linear Gaussian kernel to evaluate the similarity between two poses. The Gaussian Kernel Distance (GKD) maps the distance between predicted and ground truth pose in a value between zero and one. Similar to IoU, GKD tends to one the closer the predicted and ground truth pose are. We calculate the one-dimensional GKD for each element k_x, k_y and k_Θ, before multiplying them to get the final GKD K that replaces the IoU metric:

$$k_x = e^{\frac{-(x-x*)^2}{\sigma_x^2}}; k_y = e^{\frac{-(y-y*)^2}{\sigma_y^2}}; k_\Theta = e^{\frac{-(\Theta-\Theta^*)^2}{\sigma_\Theta^2}} \tag{4}$$

$$K = k_x \times k_y \times k_\Theta \tag{5}$$

Here, x, y, Θ are the center coordinates and the orientation of the predicted pose, x^*,y^*,Θ^* are the center coordinates and the orientation of the ground truth pose, and σ^2 is the variance. This function tends to one if two poses are similar and zero if they are rather different. Hence, instead of equally penalizing all negative locations and not considering them, the function allows the penalty given to the negative predictions within some radius around the ground truth to be reduced.

A prediction is considered correct if the GKD is greater than or equal to 0.5. Fig. 3 illustrates this behavior.

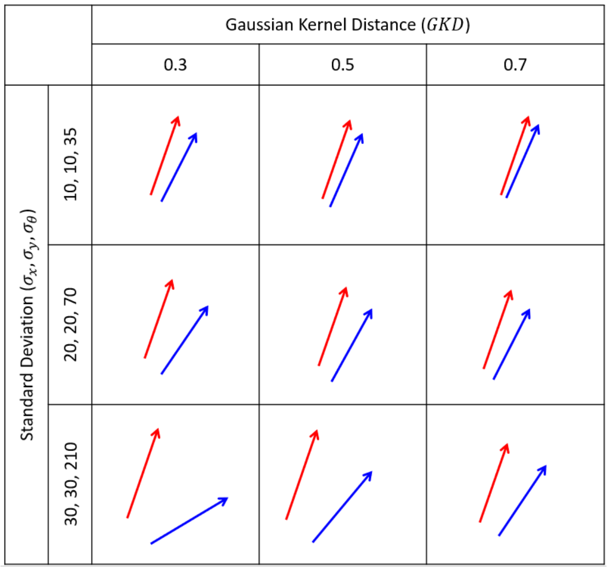

Gaussian Kernel Distance (GKD)
0.3
0.5
0.7
Standard Deviation ($\sigma_x, \sigma_y, \sigma_\theta$)
10, 10, 35
20, 20, 70
30, 30, 210

Fig. 3. **Gaussian Kernel Distance.** In order to evaluate the similarity of two poses, we use the non-linear Gaussian Kernel Distance (GKD). This figure illustrates the GKD under different hyperparameter settings.

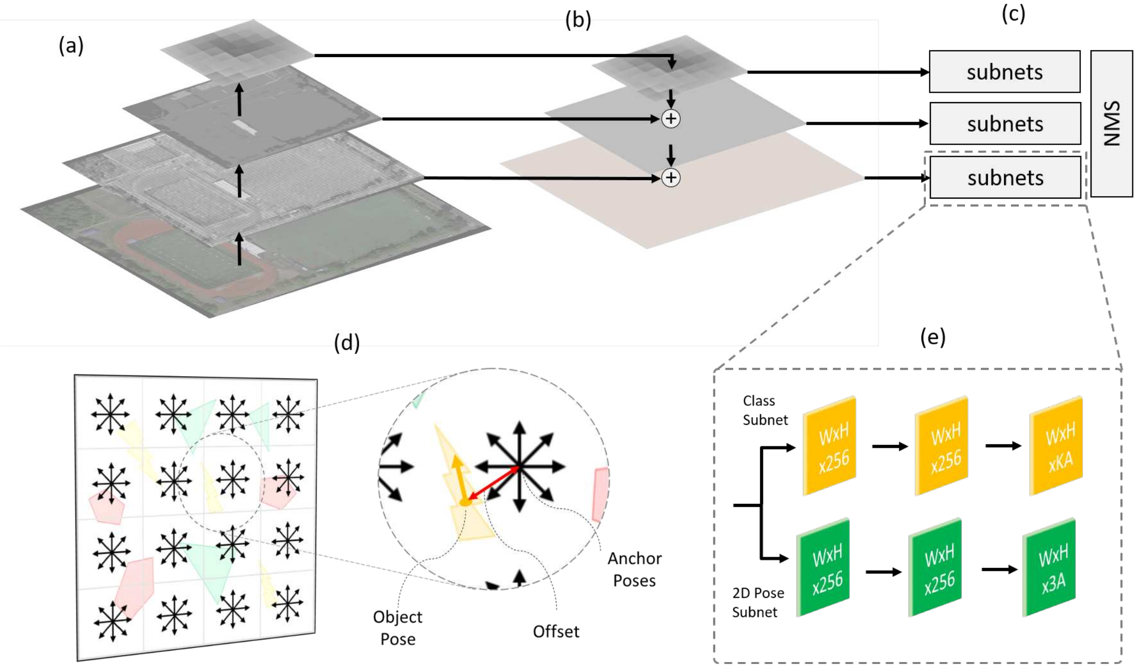


Fig. 4. **RetinaNet-2DP Architecture.** The proposed architecture builds on the modularity of RetinaNet [4]. Semantically rich feature maps are extracted by using residual networks (ResNet) [23] (a) and a feature pyramid [24] (b). Feature maps from different pyramid scales get fed into task-specific subnets for classification and localization (c). Those two parallel, task-specific branches predict class and 2D pose offset for an object using fully-convolutional layers (e). Instead of directly coming up with an objects pose, we deploy Anchor Poses and learn to predict the offset (RED) between anchor pose (BLACK) and object pose (YELLOW) (d).

C. RetinaNet-2DP

Previously discussed concepts are generally applicable to anchor-based object detection systems. Due to its modularity and strong foundation in the research community, we chose to modify the RetinaNet architecture [4] to incorporate those concepts. We call this architecture RetinaNet-2DP. The model comprises a feature extractor and two task-specific networks corresponding to the subtasks (localization and classification) that make up object detection (see Fig. 4). The class network is responsible for classification, whereas the novel pose subnet is responsible for predicting the Anchor Pose offset.

A residual network (ResNet) [23] in combination with a feature pyramid (FPN) [24] is used to extract multi-scale, semantically strong feature maps from a single-resolution input image. These feature maps are used as an input to both task specific branches. The classification branch predicts the probability of an object being present at every pyramid level for each Anchor Pose A and class K. In order to predict a 2D pose from these feature maps, we modify the localization subnet to work with 2D pose offsets as regression targets. The pose subnet is a fully convolutional network that receives extracted feature maps on different scales as input and predicts the translational and rotational offset from a given Anchor Pose (as detailed in Sec. IV-A). Due to its multi-scale and anchor-based nature, the network returns multiple possible poses for a single object. In order to determine the single most reliable pose, we use non-maximum suppression that internally builds on top of the GKD metric as introduced above.

V. Experiments

This section presents the ablation study results and illustrates two general industrial use cases that showcase the benefits of our robot-intuitive, end-to-end learning approach. All experiments used the RetinaNet-2DP implementation, as described in Section IV. The training was conducted in parallel on two Nvidia GeForce GTX 1080Ti GPUs using the Adam optimizer in Pytorch.

All models were assessed using the mean average precision metric, in which we replace the intersection over union calculation by the Gaussian Kernel Distance as discussed in Section IV-B. Note that due to this change, the results are no longer absolutely comparable to other models' performance. However, it still allows ablation of previously discussed features. The quantitative results are accompanied by qualitative visuals (see Fig. 5).

A. Ablation study

This section covers the influence of the Gaussian Kernel Distance as well as the anchor pose configuration on detection performance. For this purpose, we utilize the DOTA dataset [2]. The dataset consists of images collected from Google Earth and remote sensing satellites. It contains 2806 RGB images and comes with horizontal and oriented bounding box annotations for 15 different classes (e.g. plane, ship, storage tank, tennis and basketball court, etc.). We use the annotated OBBs and automatically convert them to 2D poses.

Influence of the standard deviation on model performance: First, we studied the influence of the standard deviation on the model performance. Since standard deviation plays a role in both loss and performance evaluation, we fixed the evaluation standard deviation to (20, 20, 70) and searched for the best loss standard deviation by training with different configurations. The results are shown in Table I. The loss standard deviation has a significant impact on the model's performance. Choosing a relatively big or small standard deviation during training (i.e. loss calculation) causes a drop in mAP. We found that choosing a standard deviation of (10, 10, 35) for the loss calculation resulted in the best model performance. Additional experiments were all conducted using this configuration for the loss standard deviation.

TABLE I

RetinaNet-2DP ablation results for standard deviation σ and anchor pose configuration on DOTA. Anchor poses are specified using set-builder notation: $\{Ax \,|\, x \in \mathbb{N}_0, 0 \leq x \leq B\}$. Model: R101-FPN-2DP

	Anchor pose configuration		Model performance mAP@0.5 GKD
Std. deviation σ for loss calculation	A	B	with $\sigma = (20, 20, 70)$
(5, 5, 35)	45	2	32.8%
(10, 10, 35)	**45**	**2**	**41.8%**
(10, 10, 70)	45	2	38.2%
(20, 20, 70)	45	2	21.8%
(10, 10, 35)	45	1	6.8%
(10, 10, 35)	45	4	36.9%
(10, 10, 35)	45	8	37.8%

Influence of good anchors on model performance: Next, we evaluated the impact of different anchor pose configurations. Table I shows the effect of the anchor configuration on the accuracy of the network. The experiment showed that an anchor configuration with the angles of (0, 45, 90) degree gives the best accuracy. Both adding and removing additional anchors causes a drop in accuracy.

B. Case Studies

In addition to the experiments on the DOTA dataset, we present two case studies that illustrate our method's application in industry. Both studies refer to robots operating in an industrial environment. Case study A describes the use of the presented method for automated container handling by a robotic arm with a vacuum gripping system. Container classification and localization using a 2D pose are performed with RetinaNet-2DP and an RGB-D camera mounted on the robot arm. For this purpose, we first created and annotated a dataset. The dataset consists of 588 RGB images of two container types. A threefold split was applied: training set (70 %), validation set (20 %), and testing set (10 %). The dataset contains two different class types. Both training and validation set were annotated using 2D poses representing the object's pose in 2D space. With a relatively simple system we achieve a maximum performance of 91.59 % mAP using (0, 45, 90, 135) as Anchor Poses. We further conducted experiments on the performance impact when using ResNeXt for feature extraction and different network depths (see Table II for further results). In case study B, we consider the application of autonomous mobile robots for the transportation of goods in industry. To enable fine positioning, the system has to classify and locate the pallets to be transported. Again, we use RetinaNet-2DP to perform object detection. In contrast to the previous case study, however, we do not rely on camera images for detection, but data from a 2D laser rangefinder. This type of scanner is commonly used in industrial mobile robots to ensure human safety. We use the publicly available pallet scan dataset from [25]. The dataset consists of 446 monochrome images of pallets. As the pallet dataset came with horizontal bounding boxes only, we manually created 2D pose annotations for it. A threefold split was applied: training set (70 %), validation set (15 %), and testing set (15 %). We achieve a maximum performance of 89.29 % mAP using ResNet-101 and (0, 45, 90, 135) as Anchor Poses. Again, we tried using ResNeXt and two different network depths as documented in Tables II.

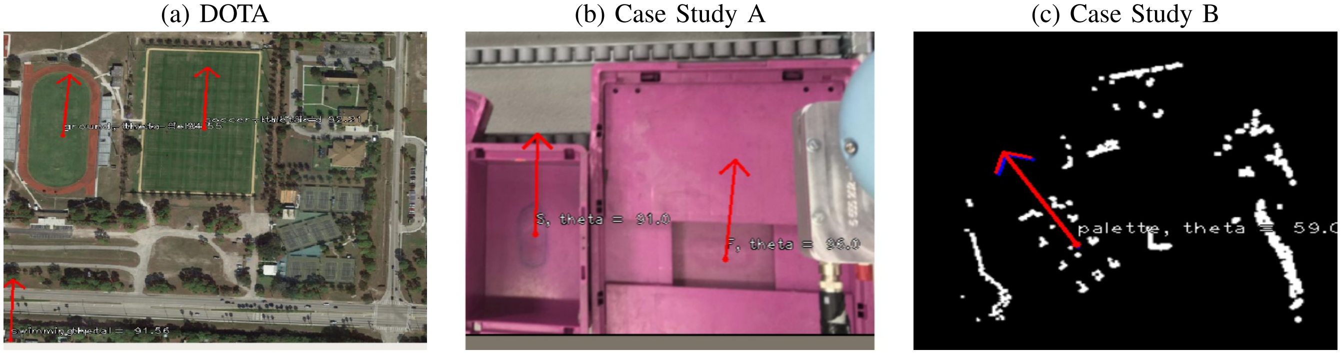
TABLE II

RetinaNet-2DP performance on two industrial case studies. Anchor poses are specified using set-builder notation: $\{Ax \,|\, x \in \mathbb{N}_0, 0 \leq x \leq B\}$. Model declaration: R: ResNet, RX: ResNext, FPN: Feature Pyramid Network

	Anchor pose configuration		Performance mAP@0.5 GKD with $\sigma = (20, 20, 70)$	
Model	A	B	Case Study A	Case Study B
R101-FPN-2DP	45	2	88.3 %	88.0%
R101-FPN-2DP	**45**	**3**	**91.6%**	**89.3%**
R101-FPN-2DP	45	4	88.1%	85.0%
R101-FPN-2DP	45	8	90.2%	88.0%
RX50-FPN-2DP	**45**	**3**	86.5%	84.0%
R50-FPN-2DP	**45**	**3**	89.2%	87.2%
RX101-FPN-2DP	**45**	**3**	89.8%	87.7%

VI. Discussion and Future Work

We have presented a simple, yet effective, localization paradigm using 2D poses and illustrated its applicability in two industrial case studies. In addition to its end-to-end learning ability, the robot-intuitive localization highlights the advantages of the localization paradigm introduced. The end-to-end learning approach enables cost-effective application and helps to overcome current challenges in industry (e.g. dynamic environment, flexibility, adaptability). Furthermore, the intuitive representation offers the potential to reduce development efforts, as post-processing procedures are omitted. Nevertheless, we see challenges that need to be overcome in future research. The GKD similarity metric introduces three additional hyperparameters (i.e. σ_x, σ_y and σ_θ) that need to be specified. Depending on the parameter configuration, the model performance using mean average precision as a key performance indicator can be misleading. Future work needs to either test other similarity metrics or define the above


(a) DOTA

(b) Case Study A

(c) Case Study B

Fig. 5. Qualitative evaluation results using RetinaNet-2DP. We train and evaluate on three different tasks the remote sensing dataset DOTA (a), industrial robotic manipulation (b) and object detection in lidar data (c). Red arrows: predictions, label: class and twist angle.

mentioned hyperparameters to best match the intersection over union metric.

VII. Conclusion

In this paper, we explore the idea of using 2D poses for object detection. Within this context, we introduce RetinaNet-2DP, which implements this paradigm. To perform the implementation, we propose the concept of Anchor Poses and deploy the Gaussian Kernel Distance as a similarity metric between poses. Beyond our use in RetinaNet-2DP, these concepts are generally applicable to anchor-based 2D pose networks. We use the DOTA dataset to determine hyperparameters and present two case studies to highlight the model's applicability for robotic automation in industry. We show that the proposed architecture is able to learn to localize objects using 2D poses and achieves a mAP (using GKD) of greater than 89% on both industrial case studies.

References

[1] C. Mayershofer, D.-M. Holm, B. Molter, and J. Fottner, "LOCO: Logistics Objects in Context," in *Proc. of 2020 IEEE International Conference on Machine Learning and Applications (ICMLA2020)*.

[2] G. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[3] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *Proc. of 2014 European Conference on Computer Vision (ECCV)*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds.

[4] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollr, "Focal loss for dense object detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2999–3007.

[5] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[6] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2018.

[7] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proc. of 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[8] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proc. of 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9626–9635.

[9] Z. Liu, H. Wang, L. Weng, and Y. Yang, "Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 8, pp. 1074–1078, 2016.

[10] J. Ding, N. Xue, Y. Long, G. Xia, and Q. Lu, "Learning roi transformer for oriented object detection in aerial images," in *Proc. of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[11] Q. Q. Liu and J. B. Li, "Orientation robust object detection in aerial images based on r-nms," *Procedia Computer Science*, vol. 154, pp. 650–656, 2019.

[12] Z. Xiao, L. Qian, W. Shao, X. Tan, and K. Wang, "Axis learning for orientated objects detection in aerial images," *Remote Sensing*, vol. 12, no. 6, pp. 908–916, 2020.

[13] A. Kirillov, Y. Wu, K. He, and R. Girshick, "Pointrend: Image segmentation as rendering," in *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[14] L. Chen and G. Papandreou and I. Kokkinos et al., "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[15] C. Wang, D. Xu, Y. Zhu, R. Martn-Martn, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," in *Proc. of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[16] Huang, S.-C. and Huang, W.-L. et al., "Efficient recognition and 6d pose tracking of markerless objects with rgb-d and motion sensors on mobile devices." in *Proc. of International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2019.

[17] C. Mayershofer, T. Ge, and J. Fottner, "Towards Fully-Synthetic Training for Industrial Applications," in *Proc. of 2020 IEEE International Conference on Logistics, Informatics and Service Sciences (LISS2020)*.

[18] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond pascal: A benchmark for 3d object detection in the wild," in *Proc. of IEEE Winter Conference on Applications of Computer Vision*, 2014, pp. 75–82.

[19] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again," in *Proc. of the IEEE International Conference on Computer Vision*, 2017.

[20] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proc. of 2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[21] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proc. of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[24] T. Lin, P. Dollr, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.

[25] I. S. Mohamed, A. Capitanelli, F. Mastrogiovanni, S. Rovetta, and R. Zaccaria, "A 2d laser rangefinder scans dataset of standard eur pallets," *Data in Brief*, p. 103837, 2019.