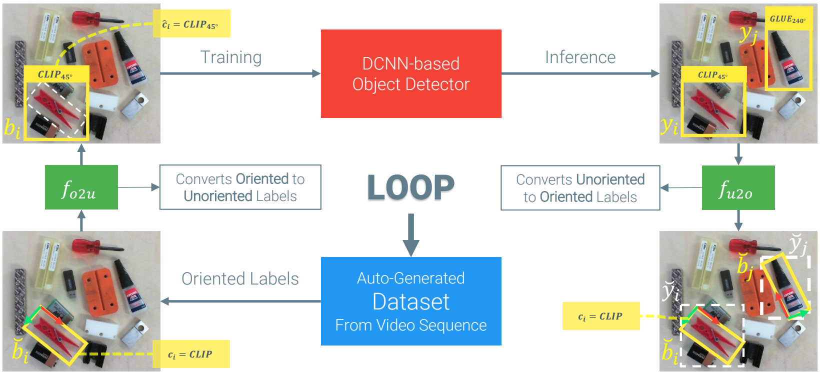
Effective Deployment of CNNs for 3DoF Pose Estimation and Grasping in Industrial Settings

D. De Gregorio[1], R. Zanella[2], G. Palli[2], L. Di Stefano[3],

[1]EYECAN.ai, [2]DEI-UNIBO, [3]DISI-UNIBO

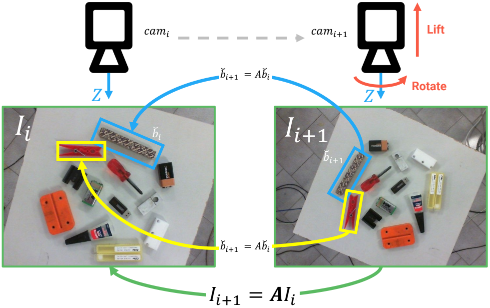
ICPR, 2020

Overall Pipeline of LOOP



Framework that **L**everages on a generic **O**bject detector for **O**rientation **P**rediction.

Auto-generated Dataset from video sequence



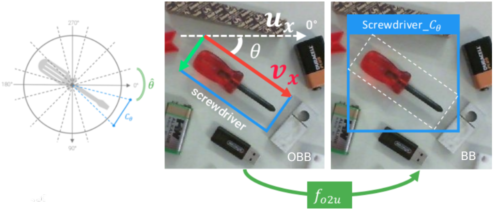
Restricted camera movements leads to a controlled rigid transformation A between the two consecutive images I_i, I_{i+1} such that $I_{i+1} = AI_i$. The same rigid transformation A can be applied to each bounding box $\breve{b}_i$ present in the image I_i so as to obtain a new set of bounding box such that $\breve{b}_{i+1} = A\breve{b}_i$

How does LOOP work?

- Given an RGB image, LOOP produces a set of predictions $\breve{y}_i = \{\breve{b}_i, \theta_i, c_i\}$, where $\breve{b}_i = \{x, y, w, h\} \in \mathbb{R}^4$ represents the coordinates of the **Oriented Bounding Box (OBB)** clockwise-rotated by an angle of θ_i and $c_i \in \mathbb{Z}^+$ is the object class.
- We leverage on a classical object detector, which outputs a set of simpler predictions $y_i = \{b_i, \hat{c}_i\}$, where $b_i = \{x, y, W, H\} \in \mathbb{R}^4$ represents the coordinates of the *unoriented* **Bounding Box (BB)** and $\hat{c}_i \in \mathbb{Z}^+$ encodes both the object class and the orientation information.

Oriented-to-Unoriented Bounding Boxes

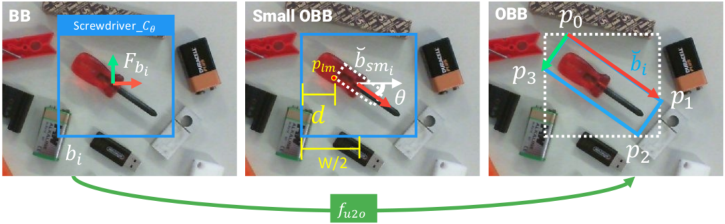
The OOB angle θ_i is estimated as a classification task by simply quantizing the angular range into k bins and by expanding all the C categories into $C' = kC$ new classes.



f_{o2u} is the function used to convert the original object class c_i in the expanded class $\hat{c}_i$ which encodes the object type and its quantized orientation.

$$f_{o2u}(c_i, \theta, \hat{\theta}) = c_i k + C_\theta = c_i k + \left\lfloor \theta / \hat{\theta} \right\rceil = \hat{c}_i$$

Unoriented-to-Oriented Bounding Boxes



We retrieve the oriented prediction from the unoreinted one generated by the Object Detector using the decoding function

$$f_{u2o}(\hat{c}_i, \hat{\theta}) = \begin{cases} c_i = \left\lfloor \frac{\hat{c}_i}{k} \right\rfloor \\ \theta_i = \hat{\theta}_i \cdot (\hat{c} \bmod k) \end{cases}$$

Experimental Dataset

We created a *Real* and a *Synthetic* dataset, with:

- 7155 *auto-labeled* images
- 12 objects divided into *Textured* and *Untextured*
- 15 tabletop scenes (3 on *homogeneous* background; 3 on *wood*; 3 on *black* and 5 on an high-clutter backgrounds)

Training and Test

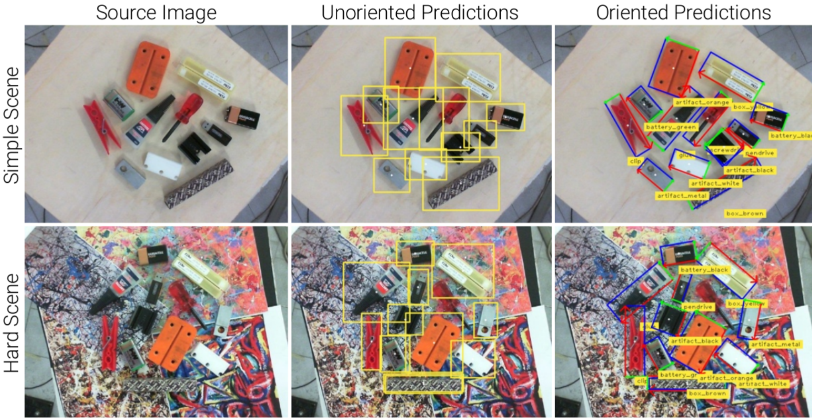
- The Object Detector used in our tests is YOLOv3
- Several models $LOOP_{\hat{\theta}}$ are trained on different discretization angles $\hat{\theta}$
- A model $LOOP^{S}_{10}$ was also trained with the synthetic dataset
- The models are tested on two scenes never seen in training, that we call *Simple Scene* (477 images) and *Hard Scene* (477 images)

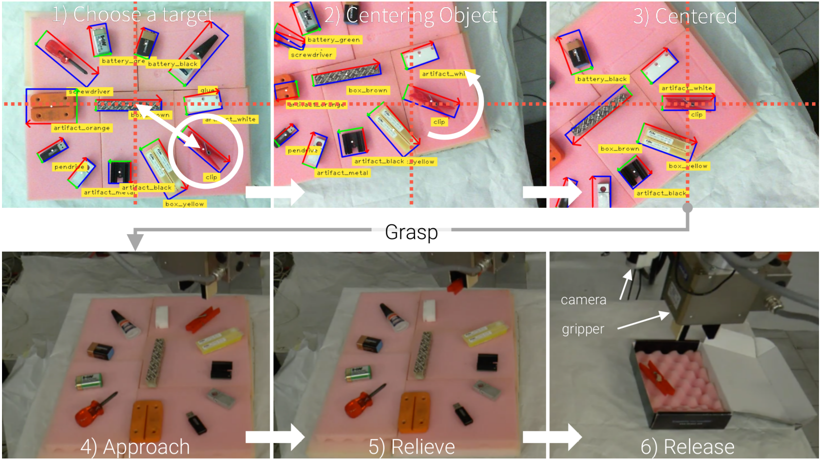
Results

The mean average precision (mAP) computed for each model, across all objects, for the *Simple Scene*, *Hard Scene* and both.

Model	Simple	Hard	Overall
$LOOP_5$	0.96	0.96	0.96
$LOOP_{10}$	**0.99**	**0.98**	**0.99**
$LOOP_{10}^S$	0.95	0.86	0.92
$LOOP_{20}$	0.95	0.97	0.96
$LOOP_{30}$	0.90	0.88	0.89
$LOOP_{45}$	0.69	0.70	0.69

Qualitative Evaluation




1) Choose a target
2) Centering Object
3) Centered
Grasp
4) Approach
5) Relieve
6) Release
camera
gripper

Supplementary Material Online

- Several runs of experiment are shown in the Video from the on-board and off-board cameras
- An open source implementation of the proposed method is available on GitHub

Thanks!