7th International Conference on
Computer Vision & Image Processing (CVIP 2022)
November 04—06, 2022

Paper Title: Expeditious Object Pose Estimation for Autonomous Robotic Grasping

Paper ID: 160

List of Authors & Affiliation:
Sri Aditya Deevi & Deepak Mishra
Indian Institute of Space Science and Technology

Presenter Information: Deepak Mishra

November 6, 2022

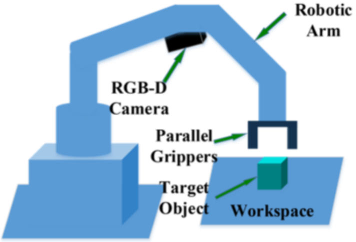
Outline of the Presentation

Problem Statement

Problem Statement

Aim

Create a 6D pose estimation pipeline for pick and place in a robotic simulation environment.



Focus of the Work

Design, development and pipeline-incorporation of DL-based pose estimation models with the qualities :

1. Efficiency : Use of only RGB image and no depth information
2. Speed : Use of no post hoc refinement stages
3. Accuracy : Good performance on relevant metrics

Focus of the Work

Design, development and pipeline-incorporation of DL-based pose estimation models with the qualities :

1. Efficiency : Use of only RGB image and no depth information
2. Speed : Use of no post hoc refinement stages
3. Accuracy : Good performance on relevant metrics

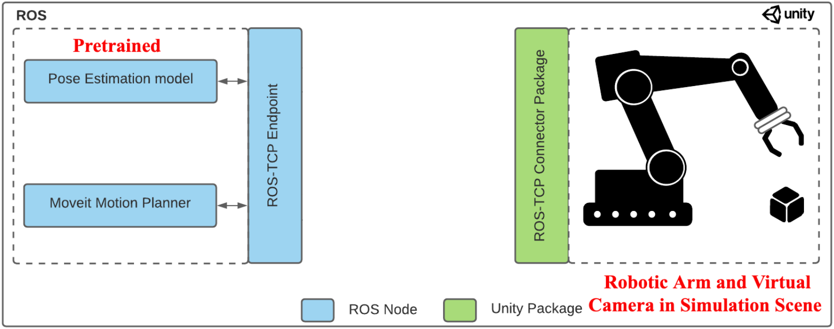
Focus of the Work

Design, development and pipeline-incorporation of DL-based pose estimation models with the qualities :

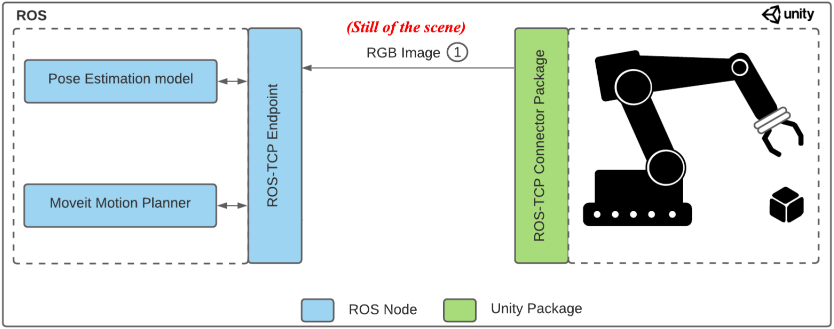
1. Efficiency : Use of only RGB image and no depth information
2. Speed : Use of no post hoc refinement stages
3. Accuracy : Good performance on relevant metrics

Overview of the Approach

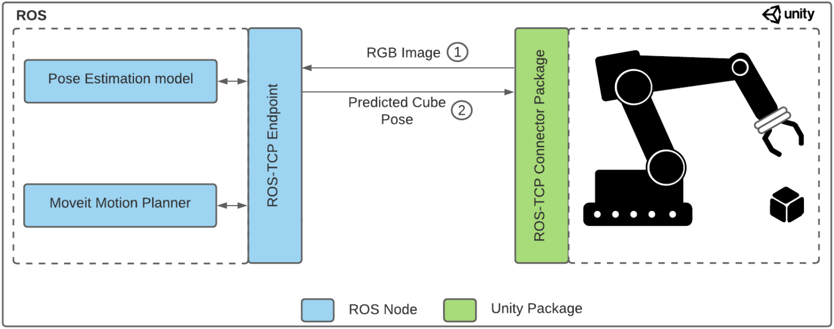
Bird's Eye View



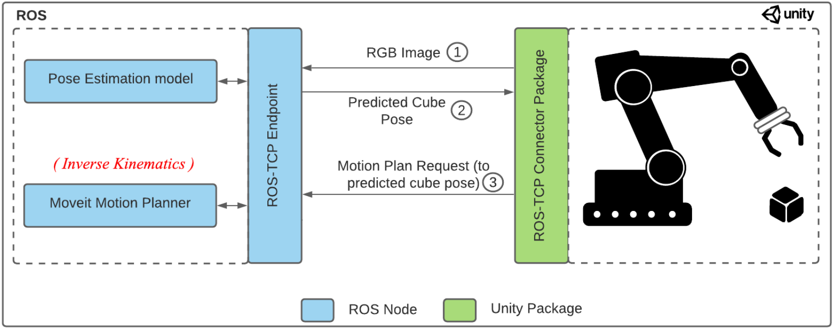
Bird's Eye View



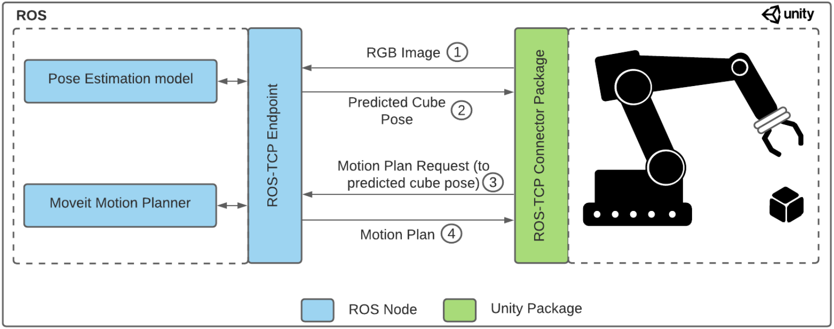
Bird's Eye View



Bird's Eye View

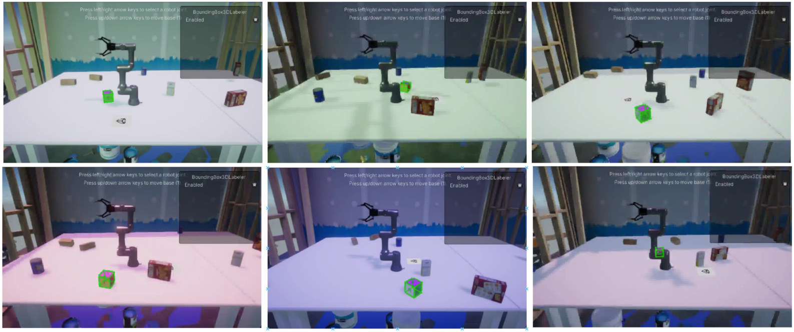


Bird's Eye View



Phases of the Approach

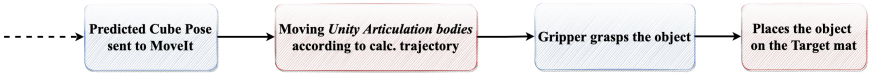
- Training Phase : Collect domain randomized, labelled synthetic data from simulation scene and train the model !



Phases of the Approach

- Training Phase : Collect domain randomized, labelled synthetic data from simulation scene and train the model !

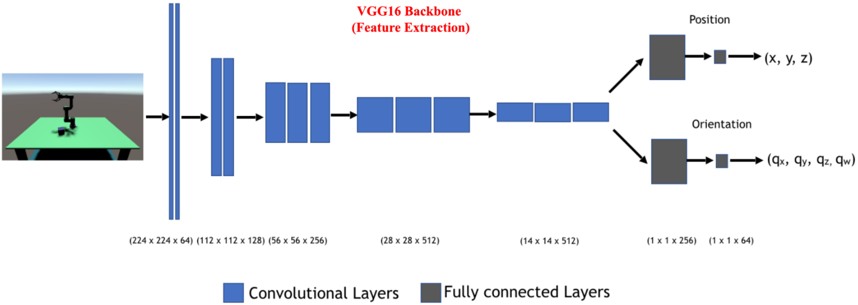
- Test Phase :



Pose Estimation Models

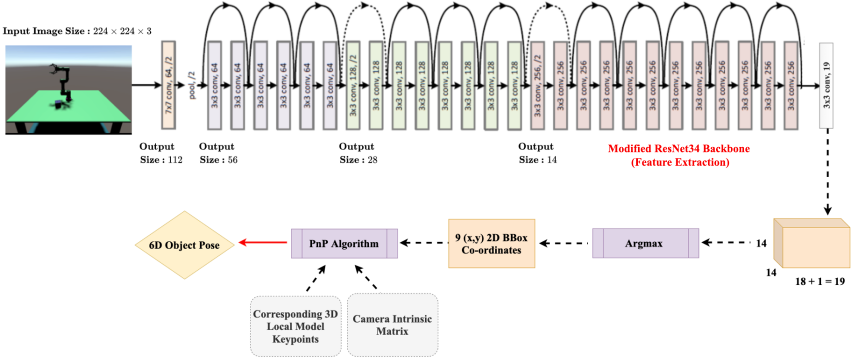
Model-1 : UnityVGG16

- Template based approach that directly regresses the pose information
- Transfer Learning utilized



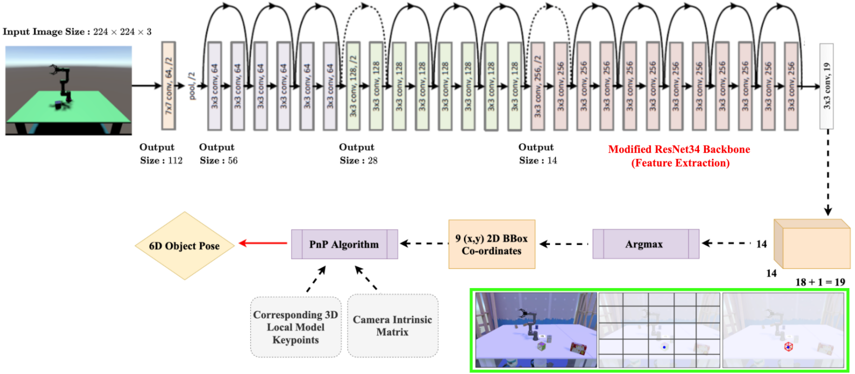
Model-2 : Pose6DSSD

- Correspondence based approach where we first the regress the 2D image coordinates of certain keypoints
- PnP algorithm used to predict the final 6D object pose



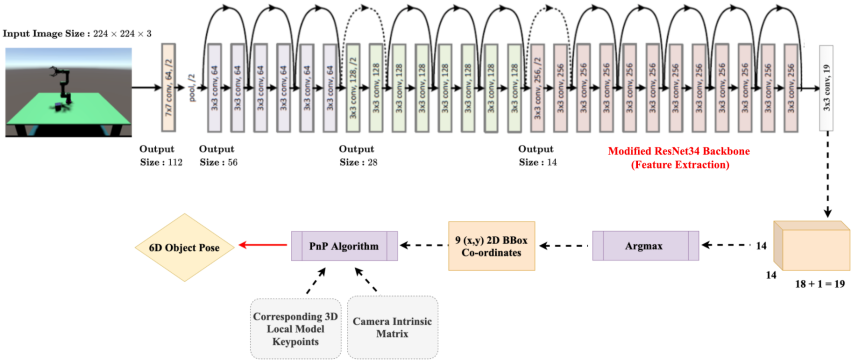
Model-2 : Pose6DSSD

- Correspondence based approach where we first the regress the 2D image coordinates of certain keypoints
- PnP algorithm used to predict the final 6D object pose



Model-2 : Pose6DSSD

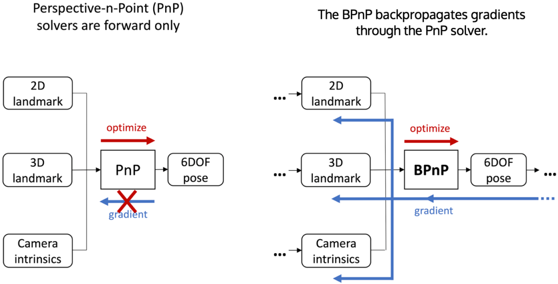
- Correspondence based approach where we first the regress the 2D image coordinates of certain keypoints (No FC Layers)
- PnP algorithm used to predict the final 6D object pose (Not E2E trainable)



Model-3 : DOSSE-6D (v2) [Deep Object Single Stage Estimator]

Correspondence based approach similar to the Pose6DSSD, additional elements are :

- Backpropagatable PnP (BPnP) Module [2] → Define a stationary constraint and Implicit Derivative

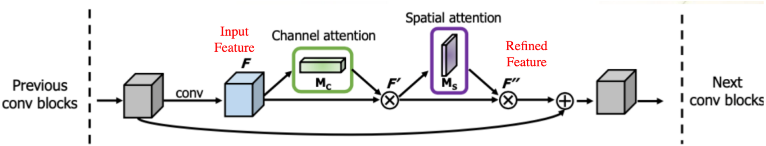


- Attention Module

Model-3 : DOSSE-6D (v2)

Correspondence based approach similar to the Pose6DSSD, additional elements are :

- Backpropagatable PnP (BPnP) Module
- Attention Module [10][11]→ Convolution based efficient channel and spatial attention, using MaxPool & AvgPool features



ResBlock + Attention Module

Model-4 : AHR-DOSSE-6D [AHR – Attention High Resolution]

Best performing model due to the following elements :

1. Single Stage E2E trainable, correspondence approach without post-refinement stages $\rightarrow$ BPnP Module
2. Use of attention module $\rightarrow$ Channel + Spatial
3. Maintain **High-Resolution** feature representations throughout the backbone [7] $\rightarrow$ AHRNet Backbone
4. Increased input image resolution $\rightarrow$ Parameter efficiency maintained
5. Use of more geometrical details $\rightarrow$ Farthest Point Sampling

Model-4 : AHR-DOSSE-6D

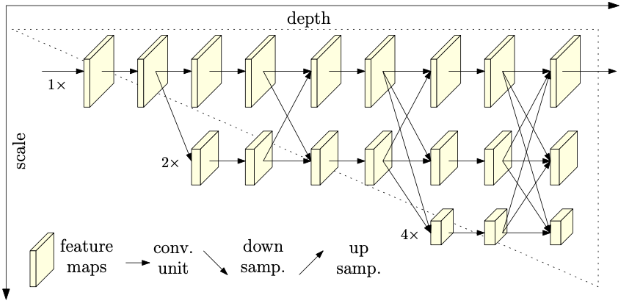
Best performing model due to the following elements :

1. Single Stage E2E trainable, correspondence approach without post-refinement stages $\rightarrow$ BPnP Module
2. Use of attention module $\rightarrow$ Channel + Spatial
3. Maintain **High-Resolution** feature representations throughout the backbone [7] $\rightarrow$ AHRNet Backbone
4. Increased input image resolution $\rightarrow$ Parameter efficiency maintained
5. Use of more geometrical details $\rightarrow$ Farthest Point Sampling

Model-4 : AHR-DOSSE-6D

Best performing model due to the following elements :

1. Single Stage E2E trainable, correspondence approach without post-refinement stages → BPnP Module
2. Use of attention module → Channel + Spatial
3. Maintain **High-Resolution** feature representations throughout the backbone [7] → AHRNet Backbone



Model-4 : AHR-DOSSE-6D

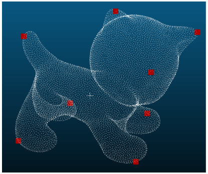
Best performing model due to the following elements :

1. Single Stage E2E trainable, correspondence approach without post-refinement stages → BPnP Module
2. Use of attention module → Channel + Spatial
3. Maintain **High-Resolution** feature representations throughout the backbone [7] → AHRNet Backbone
4. Increased input image resolution → Parameter efficiency maintained
5. Use of more geometrical details → Farthest Point Sampling

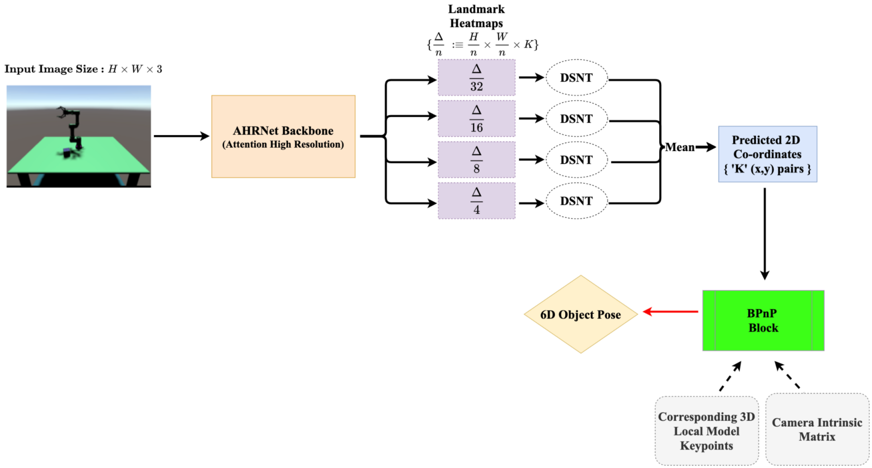
Model-4 : AHR-DOSSE-6D

Best performing model due to the following elements :

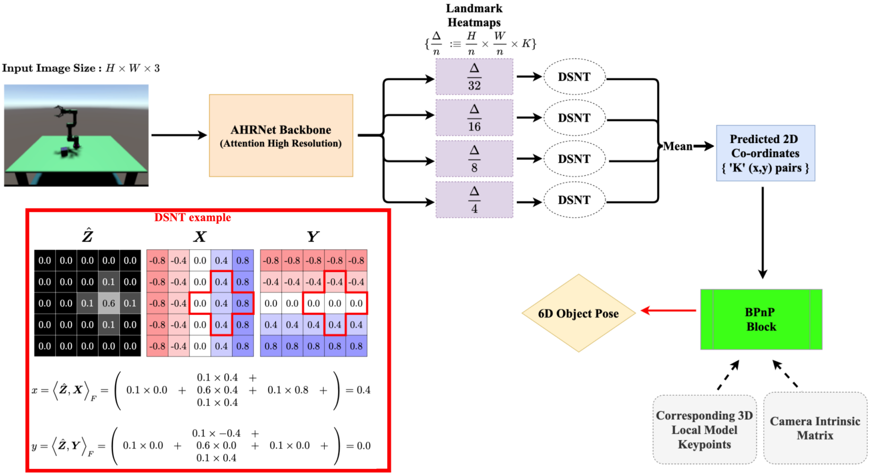
1. Single Stage E2E trainable, correspondence approach without post-refinement stages $\rightarrow$ BPnP Module
2. Use of attention module $\rightarrow$ Channel + Spatial
3. Maintain **High-Resolution** feature representations throughout the backbone [7] $\rightarrow$ AHRNet Backbone
4. Increased input image resolution $\rightarrow$ Parameter efficiency maintained
5. Use of more geometrical details $\rightarrow$ Farthest Point Sampling



AHR-DOSSE-6D : High Level Block Diagram



AHR-DOSSE-6D : High Level Block Diagram

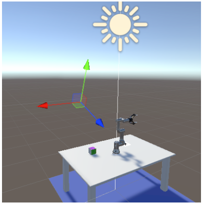


0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.1	0.0
0.0	0.0	0.1	0.6	0.1
0.0	0.0	0.0	0.1	0.0
0.0	0.0	0.0	0.0	0.0

-0.8	-0.4	0.0	0.4	0.8
-0.8	-0.4	0.0	0.4	0.8
-0.8	-0.4	0.0	0.4	0.8
-0.8	-0.4	0.0	0.4	0.8
-0.8	-0.4	0.0	0.4	0.8

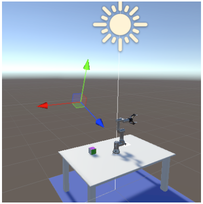
-0.8	-0.8	-0.8	-0.8	-0.8
-0.4	-0.4	-0.4	-0.4	-0.4
0.0	0.0	0.0	0.0	0.0
0.4	0.4	0.4	0.4	0.4
0.8	0.8	0.8	0.8	0.8

Experimental Configuration



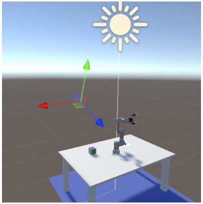
- Two simulation scenes – Simple and Cluttered

Experimental Configuration



- Two simulation scenes – Simple and Cluttered
- Experiments – Same Environment and Cross Environment cases

Experimental Configuration



- Two simulation scenes – Simple and Cluttered
- Experiments – Same Environment and Cross Environment cases

- Mixture Loss function considered : *[For AHR-DOSSE-6D]*

$$\mathcal{L} = \overbrace{\lambda_{heat}\mathcal{L}_{heat} + \lambda_{reproj}\mathcal{L}_{reproj}}^{\text{Indirect Supervision}} + \overbrace{\lambda_{add}\mathcal{L}_{add}}^{\text{Direct Supervision}}$$

$$\mathcal{L}_{\text{heat}} = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{K}\sum_{k=1}^{K}\left\|\mathbf{H}_k^{s,pred} - \mathbf{H}_k^{s,\text{true}}\right\|_F^2 \quad ; \quad \mathcal{L}_{reproj} = \frac{1}{K}\sum_{i=1}^{K}\|\mathbf{x_i} - \boldsymbol{\pi_i}\|^2$$

$$\mathcal{L}_{\text{add}} = \frac{1}{m}\sum_{\mathbf{x}\in\mathcal{M}}\|(\mathbf{Rx}+\mathbf{T}) - (\tilde{\mathbf{R}}\mathbf{x}+\tilde{\mathbf{T}})\|^2$$

Results

Results on Unity Synthetic Data [Cube Object, TEST SPLIT : 3000 RGB images]

$$\text{ADD} = \frac{1}{m} \sum_{\mathbf{x} \in \mathcal{M}} \|(\mathbf{R}\mathbf{x} + \mathbf{T}) - (\tilde{\mathbf{R}}\mathbf{x} + \tilde{\mathbf{T}})\|$$

(Lower is better !)

S.No.	Approach \ Expt. Config.	Train-Clutter + Test-Clutter	Train-Clutter + Test-Simple	Train-Simple + Test-Simple	Train-Simple + Test-Clutter
1.	**UnityVGG16**	1.6801	16.5287	2.0248	53.7345
2.	**Pose6DSSD**	1.3976	9.0066	1.0054	39.0549
3.	**DOSSE-6D_v1**	1.2150	3.9213	0.9789	58.1505
4.	**DOSSE-6D_v2**	0.8836	10.5477	0.7604	41.8551
5.	**DOSSE-6D_v3**	0.9540	30.3129	1.0083	48.6070
6.	**AHR-DOSSE-6D**	0.4192	22.6130	0.4685	92.2395

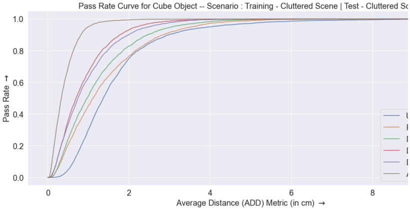
Table: Table displaying the average ADD metric values (in cm)

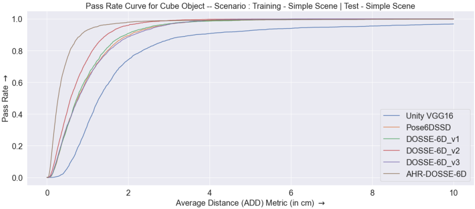
Results on Unity Synthetic Data [Cube Object, TEST SPLIT : 3000 RGB images]

$$\text{ADD} = \frac{1}{m} \sum_{\mathbf{x} \in \mathcal{M}} \|(\mathbf{Rx} + \mathbf{T}) - (\tilde{\mathbf{R}}\mathbf{x} + \tilde{\mathbf{T}})\|$$ *(Lower is better !)*

S.No.	Expt. Config. / Approach	Train-Clutter + Test-Clutter	Train-Clutter + Test-Simple	Train-Simple + Test-Simple	Train-Simple + Test-Clutter
1.	**UnityVGG16**	1.6801	16.5287	2.0248	53.7345
2.	**Pose6DSSD**	1.3976	9.0066	1.0054	39.0549
3.	**DOSSE-6D_v1**	1.2150	3.9213	0.9789	58.1505
4.	**DOSSE-6D_v2**	0.8836	10.5477	0.7604	41.8551
5.	**DOSSE-6D_v3**	0.9540	30.3129	1.0083	48.6070
6.	**AHR-DOSSE-6D**	0.4192	22.6130	0.4685	92.2395

Table: Table displaying the average ADD metric values (in cm)





Results on LINEMOD Benchmark [Data Augmentation used]

$$\text{ADD} = \frac{1}{m} \sum_{\mathbf{x} \in \mathcal{M}} \|(\mathbf{R}\mathbf{x} + \mathbf{T}) - (\tilde{\mathbf{R}}\mathbf{x} + \tilde{\mathbf{T}})\|$$ *(Lower is better !)*

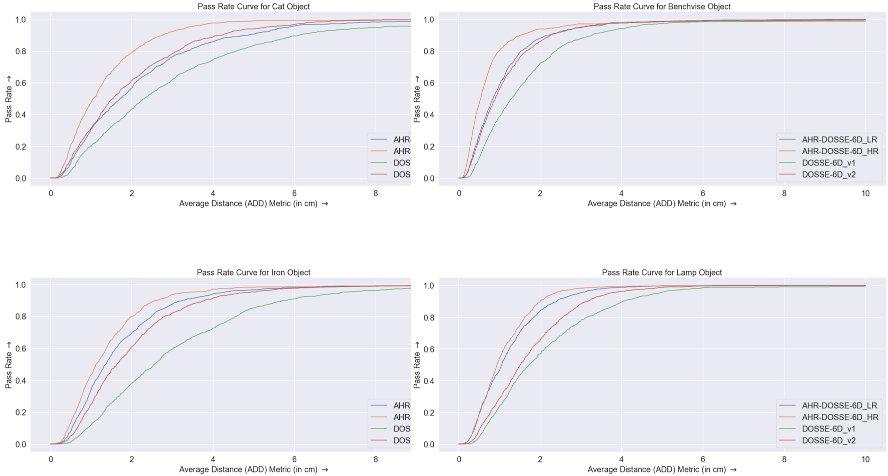
S.No.	Object / Approach	Cat ($d = 15.50$ cm)	Benchvise ($d = 28.69$ cm)	Lamp ($d = 28.52$ cm)	Can ($d = 20.20$ cm)	Iron ($d = 30.32$ cm)
1.	**SSD-6D** [5]	0.51	0.18	8.20	1.35	8.86
2.	**Tekin et al.** [8]	41.82	81.80	71.11	68.80	74.97
3.	**DOSSE-6D_v1**	33.45	86.77	74.94	60.19	60.22
4.	**DOSSE-6D_v2**	50.23	94.53	85.55	78.01	82.45
5.	**AHR-DOSSE-6D_LR**	45.89	94.30	94.36	84.14	88.10
6.	**AHR-DOSSE-6D_HR**	68.31	96.69	97.86	95.02	93.63

Table: Table displaying the ADD metric pass rates (in %).

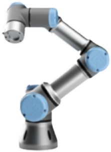
Input Image sizes :

- DOSSE-6D_v1, AHR-DOSSE-6D_LR → $224 \times 224 \times 3$
- DOSSE-6D_v2, AHR-DOSSE-6D_HR → $448 \times 448 \times 3$

Results on LINEMOD Benchmark [Graphical]



Thankyou!



References I

[1] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. "Benchmarking in Manipulation Research: Using the Yale-CMU-Berkeley Object and Model Set". In: *IEEE Robotics Automation Magazine* 22.3 (2015), pages 36–52. DOI: 10.1109/MRA.2015.2448951.

[2] Bo Chen, Alvaro Parra, Jiewei Cao, Nan Li, and Tat-Jun Chin. "End-to-end learnable geometric vision by backpropagating PnP optimization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pages 8100–8109.

[3] Guoguang Du, Kai Wang, Shiguo Lian, and Kaiyong Zhao. "Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review". In: *Artificial Intelligence Review* 54.3 (2021), pages 1677–1734.

[4] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. "Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes". In: *Computer Vision – ACCV 2012*. Edited by Kyoung Mu Lee, Yasuyuki Matsushita, James M. Rehg, and Zhanyi Hu. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pages 548–562. ISBN: 978-3-642-37331-2.

[5] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. "Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pages 1521–1529.

[6] Aiden Nibali, Zhen He, Stuart Morgan, and Luke Prendergast. "Numerical coordinate regression with convolutional neural networks". In: *arXiv preprint arXiv:1801.07372* (2018).

[7] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. "Deep high-resolution representation learning for human pose estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pages 5693–5703.

References II

[8] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. “Real-time seamless single shot 6d object pose prediction”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pages 292–301.

[9] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. “Domain randomization for transferring deep neural networks from simulation to the real world”. In: *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2017, pages 23–30.

[10] Qilong Wang, Banggu Wu, Pengfei Zhu, P. Li, Wangmeng Zuo, and Qinghua Hu. “ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pages 11531–11539.

[11] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. “Cbam: Convolutional block attention module”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pages 3–19.

Questions ?