Enhancing 6D Object Pose Estimation

Fabrizio Costa
Politecnico di Torino
Turin, Italy
s337191@studenti.polito.it

Mirko Dinapoli
Politecnico di Torino
Turin, Italy
s346342@studenti.polito.it

Ali Moghadasi
Politecnico di Torino
Turin, Italy
s341002@studenti.polito.it

Faridreza Momtaz Zandi
Politecnico di Torino
Turin, Italy
s339366@studenti.polito.it

Abstract

This paper addresses the problem of 6D object pose estimation from RGB-D images, with a particular focus on how depth information can be effectively exploited within a learning-based pipeline. We first introduce a baseline approach composed of object detection via YOLO, followed by RGB-based feature extraction using ResNet-50 to regress object orientation and an inverse pinhole projection of the central pixel to estimate object translation.
Building upon this baseline, we investigated two complementary strategies to better leverage depth cues. First, we incorporate depth information through a late-fusion mechanism to refine rotation estimation. Second, we replace the conventional center-based projection with an encoder-decoder architecture that learns to weight image pixels by their relevance to translation estimation.
Experiments conducted on the LineMOD dataset demonstrate that the proposed enhancements significantly improve pose accuracy compared to the baseline, highlighting the importance of selectively integrating depth information rather than treating it as a uniform input.

The code for this project is publicly available at https://github.com/fabCosta999/6d-pose-estimation.git.

1. Introduction

The task of 6D object pose estimation, which involves determining the 3D position (translation) and orientation (rotation) of an object in a given coordinate system, is a fundamental challenge in computer vision. This capability is critical for numerous real-world applications, including robotic manipulation, where precise spatial understanding is required for grasping, as well as in augmented reality (AR) and autonomous navigation systems [9, 14]. Despite its importance, estimating the 6D pose remains a complex problem due to factors such as object occlusions, variations in lighting, and the presence of texture-less or symmetric objects within cluttered environments.

Historically, 6D pose estimation techniques have evolved from classical feature-matching methods to sophisticated deep learning architectures. While initial approaches relied heavily on RGB images, the advent of depth-sensing technology has paved the way for RGB-D based methods that leverage geometric information to overcome the limitations of purely visual features [5]. However, processing depth information often introduces significant computational overhead, especially in real-time robotic applications.

In this project, we develop a comprehensive end-to-end pipeline for 6D pose estimation, focusing on the trade-off between accuracy and efficiency. Our methodology follows a multi-phase incremental approach. First, we implement a robust object detection module using the YOLO (You Only Look Once) architecture [7] to localize target objects within the LineMod dataset. Following localization, we establish a baseline pose estimation model utilizing a ResNet-50 backbone [3]. This model is designed to regress the orientation of objects directly in the form of quaternions from RGB features, while the translation is computed projecting the central pixel of the detection.

To further enhance the precision of our predictions, we extend the baseline by incorporating depth information through a multi-modal fusion technique. Inspired by the DenseFusion architecture [16], our extension leverages a simplified 2D-based fusion approach. By processing RGB texture and depth geometric structure separately and fusing them via feature concatenation, we aim to improve robustness without the prohibitive cost of full 3D point cloud processing. The translation prediction is upgraded by ceasing

to consider only the central pixel of the detection, and use a model that learns which pixel to project in 3d space.

2. Related Work

The estimation of 6D object pose has undergone a significant evolution, transitioning from classical template-based methods to sophisticated deep learning architectures that operate across multiple data modalities.

Pose from RGB data Approaches relying exclusively on RGB data represent one of the most persistent challenges in the field due to the inherent loss of explicit spatial information in the 2D projection. Initially, the literature focused on the use of multimodal templates for detecting texture-less objects in heavily cluttered scenes [4] or on the alignment between 2D image projections and large-scale 3D CAD models [1]. Methods such as the one proposed by Rios-Cabrera and Tuytelaars [13] demonstrated the effectiveness of discriminatively trained templates for achieving real-time performance.

With the advent of **deep learning**, RGB-based pose estimation has been addressed either as a direct regression problem or through the detection of keypoints. PoseCNN [17] introduced a seminal framework for quaternion regression and object center localization, while SSD-6D [6] extended 2D detection paradigms to cover the discrete space of 6D poses. To overcome the limitations of direct regression, algorithms like DeepIM [8] introduced iterative refinement processes based on comparing the observation with a rendered model. Other research directions have instead focused on estimating local 3D coordinates or semantic keypoints [10] to solve for the pose using Perspective-n-Point (PnP) algorithms.

Pose from RGB-D data. The integration of RGB-D data has enabled the mitigation of geometric ambiguities typical of monocular sensors by leveraging explicit depth information. Research based on PointNet [11] and its subsequent evolutions allowed for the direct processing of point clouds, leading to methods such as Frustum PointNets [12], which constrain the geometric search space starting from an initial 2D detection. Sensor fusion has been further refined by architectures like PointFusion [18] and DenseFusion, which extract dense per-pixel features by integrating color information with local geometry. Despite the dominance of deep learning, methods based on geometric descriptors, such as Point Pair Features [2, 15], remain fundamental for their robustness in industrial scenarios involving objects with minimal visual distinctiveness.

3. Model

Our baseline architecture receives as input RGB data and outputs translation $t \in \mathbb{R}^3$ and rotation, expressed as quaternions $q \in S^3$, chosen due to their computational efficiency and lack of gimbal lock issues. Compared to full $SO(3)$ rotation matrices, quaternions provide a more compact parameterization (4 vs 9 parameters), which facilitates the convergence of the regression head. Since our goal is to improve the 6D pose detection accuracy, we introduce some extensions to our baseline that exploit depth data.

3.1. Baseline

Our baseline architecture operates in two sequential stages to decouple the estimation of translation and rotation.

The first stage employs YOLO11s [7] for object localization and classification. It processes a 640×640 RGB image to output bounding box coordinates $[x_{\min}, x_{\max}, y_{\min}, y_{\max}]$ and class label. To get the 3D translation $\mathbf{t} = [X, Y, Z]^T$, we apply the inverse Pinhole camera model. The (X, Y) components are back-projected from the bounding box center (bb_x, bb_y) using the camera intrinsic matrix K:

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

The Z component (depth) is directly sampled from the depth map at the pixel coordinates corresponding to the predicted centroid. The final 3D position is computed as:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = Z \cdot K^{-1} \begin{bmatrix} bb_x \\ bb_y \\ 1 \end{bmatrix} \tag{2}$$

The second stage focuses on rotation regression. Using the bounding box from the previous step, we extract a 224×224 RGB crop which is fed into a ResNet-50 backbone. This network is modified to output a 4-dimensional vector representing the rotation quaternion $\mathbf{q} \in S^3$.
For the rotation regression task we use the Symmetry Aware Geodesic Loss:

$$\mathcal{L}_{rot}(\mathbf{q}_{pred}, \mathbf{q}_{gt}) = \begin{cases} d_{geo}(\mathbf{q}_{pred}, \mathbf{q}_{gt}) \text{ if not sym} \\ \min_{\mathbf{q}_s \in \mathcal{S}_l} d_{geo}(\mathbf{q}_{pred}, \mathbf{q}_{gt} \otimes \mathbf{q}_s) \text{ if sym} \end{cases} \tag{3}$$

where:

- d_{geo} is the geodesic distance, which is the minimum angle (expressed in radians) needed so that the two quaternions represent the same rotation:

$$d_{geo}(\mathbf{q}_1, \mathbf{q}_2) = 2 \arccos(|\langle \mathbf{q}_1, \mathbf{q}_2 \rangle|) \tag{4}$$

- $\min_{\mathbf{q}_s \in \mathcal{S}_l} d_{geo}(\mathbf{q}_{pred}, \mathbf{q}_{gt} \otimes \mathbf{q}_s)$ minimizes over the set of discrete rotational symmetries (S_l) proper to the object category. In this way the loss becomes invariant to equivalent poses, effectively penalizing only the distance to the nearest one.
  In particular ($\mathbf{q}_{gt} \otimes \mathbf{q}_s$) is the Hamiltonian product between the ground truth quaternion and a symmetry transformation $\mathbf{q}_s \in S_l$, thus obtaining another visually equivalent ground truth.

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \begin{bmatrix} w_1 & -x_1 & -y_1 & -z_1 \\ x_1 & w_1 & -z_1 & y_1 \\ y_1 & z_1 & w_1 & -x_1 \\ z_1 & -y_1 & x_1 & w_1 \end{bmatrix} \begin{bmatrix} w_2 \\ x_2 \\ y_2 \\ z_2 \end{bmatrix} \quad (5)$$

  where $\mathbf{q}_1$ and $\mathbf{q}_2$ correspond to $\mathbf{q}_{gt}$ and $\mathbf{q}_s$ respectively.

3.1.1. YOLO11

YOLO11 represents the latest version of YOLO, characterised by high accuracy, while being computationally efficient. Its architecture consists of three main components. First, the **backbone** serves as the primary feature extractor, utilizing convolutional neural networks to transform raw image data into multi-scale feature maps. Second, the **neck** component acts as an intermediate processing stage, employing specialized layers to aggregate and enhance feature representations across different scales. Third, the **head** component functions as the prediction mechanism, generating the final outputs for object localization and classification based on the refined feature maps. YOLOv11's main innovations with respect to past iterations are:

- **C3K2** block (Cross Stage Partial with kernel size 2): it is a more computationally efficient implementation of Cross Stage Partial Bottleneck, employing two smaller convolutions instead of one large one. The "k2" indicates a smaller kernel size, which contributes to faster processing while maintaining performance.
- **C2PSA** (Convolutional block with Parallel Spatial Attention): this mechanism allows the model to focus more effectively on important regions within the image, enabling YOLO11 to concentrate on specific areas of interest potentially improving detection accuracy of varying sizes and positions.

To finetune the network we have frozen its first 10 layers, corresponding to the backbone.

3.1.2. ResNet-50

ResNet-50 is a convolutional network characterised by the presence of residual blocks with skip (shortcut) connections to address the vanishing gradient problem. It has 50 layers organized into four main stages, in each of which the channel width is doubled while the spatial dimensions are halved. Each residual block employs a bottleneck design, consisting of three convolutional layers: a 1×1 convolution to reduce dimensionality, a 3×3 convolution as the bottleneck, and a final 1×1 convolution to restore dimensionality. This bottleneck structure allows for efficient computation and parameter usage in deeper networks. We substitute the final fully connected layer with an MLP which outputs 4 neurons, which are the components of the quaternion, normalized as it must have unit norm.

3.2. RGB-D Network for Rotation Estimation

To overcome the limitations of the RGB-only baseline, we implement an RGB-D network extension, shown in Fig. 1, inspired by the DenseFusion architecture [16]. The core of this extension consists of two parallel convolutional networks whose outputs are fused to ensure robust rotation estimation. While the baseline relied solely on visual features, this multi-modal approach leverages both the visual texture from RGB images and the geometric structure provided by depth maps.

In this architecture, feature extraction is performed through two parallel branches: a ResNet-50 backbone [3] for extracting high-level visual features and a dedicated convolutional neural network (CNN) for processing depth information. These heterogeneous features are subsequently fused through a concatenation step, creating a dense feature representation that captures the spatial relationship between color and geometry.

The fused representation is then given as input to a final fully connected layer, acting as the regression head to predict the object's orientation. Unlike the baseline, this regression head is designed to map the combined RGB-D features into a 4-dimensional unit quaternion representing the rotation. By adopting this parallel fusion structure for rotation, the model achieves higher precision in cluttered

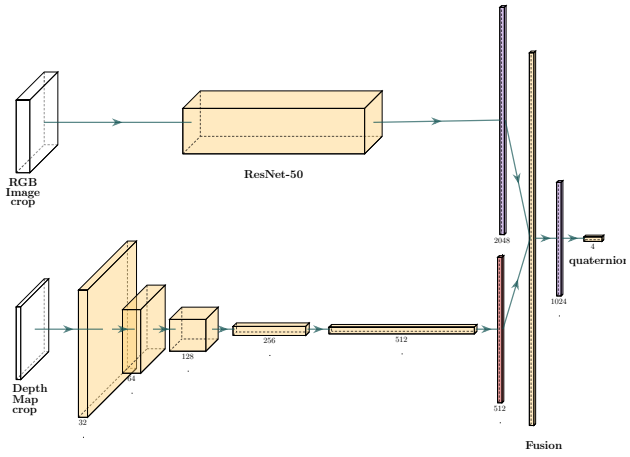


Figure 1. Architecture of the proposed RGB-D extension network. The RGB crop is processed by a ResNet-50 backbone, while the depth crop is processed by a shallow CNN. Features are fused to regress the orientation quaternion.

scenes, as the geometric cues from the depth channel compensate for potential ambiguities in the RGB texture.

3.3. Encoder-Decoder for Translation Estimation

In the baseline approach, object translation is estimated by applying the inverse pinhole projection to the depth value at the center of the detected bounding box. While simple and effective in many cases, this strategy is sensitive to detection inaccuracies and depth artifacts. In particular, if the selected pixel does not belong to an object surface, there wouldn't be a corresponding valid depth value, causing the translation estimate to fail entirely. This situation can arise even with accurate detections, for instance in the presence of object cavities or thin structures.

Since depth measurements are only defined on object pixels, we explicitly identify valid depth locations and define $M(p)$ as the valid depth mask, where:

$$M(p) = \begin{cases} 1 & \text{if } p \text{ has a valid depth} \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

Depth maps provide spatially distributed geometric information, where different pixels may contribute unequally to a reliable translation estimate. Motivated by this observation, we replace the single-point projection with a learned pixel-wise weighting mechanism that aggregates depth information over the object surface.

The proposed encoder-decoder network takes as input the cropped RGB image, the corresponding depth map, and a 2D coordinate grid encoding pixel locations within the crop. The network outputs a single-channel logit map $L \in \mathbb{R}^{H \times W}$, representing the unnormalized importance of each pixel for translation estimation.

The encoder progressively downsamples the input through a sequence of convolutional layers in order to capture global geometric and contextual patterns. The decoder mirrors this process by gradually restoring spatial resolution using transposed convolutions. To preserve fine-grained spatial information that would otherwise be lost during downsampling, skip connections are introduced between corresponding encoder and decoder stages. These connections allow high-resolution local features to be directly propagated to deeper layers, which is crucial for accurate pixel-level weight prediction.

An overview of the proposed encoder-decoder architecture for translation estimation is shown in Fig. 2.

To obtain a normalized weighting distribution, a spatial softmax operation is applied to the predicted logits, restricted to pixels with valid depth measurements:

$$w(p) = \frac{\exp\left(\frac{L(p)}{\tau}\right) M(p)}{\sum_q \exp\left(\frac{L(q)}{\tau}\right) M(q)}, \tag{7}$$

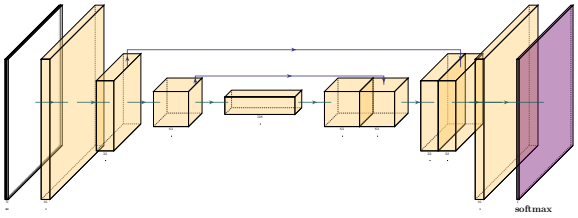


Figure 2. Architecture of the proposed encoder–decoder network for translation estimation. The network predicts a pixel-wise importance map that is later normalized and used to aggregate depth information.

where τ is the temperature parameter. We selected a low temperature value to encourage sparse distributions, forcing the network to concentrate the probability mass on a limited set of informative pixels.

Each valid depth pixel is projected into 3D space using the inverse pinhole camera model, obtaining the projected points $X(p)$.

The final translation vector is computed as a weighted average of the reconstructed 3D points:

$$\hat{\mathbf{t}} = \sum_p w(p)\, \mathbf{X}(p). \tag{8}$$

The network is trained by minimizing a translation loss defined as

$$\mathcal{L}_{\text{trans}} = \left\| \hat{\mathbf{t}} - \mathbf{t}_{\text{gt}} \right\|_1, \tag{9}$$

where $\mathbf{t}_{\text{gt}}$ denotes the ground-truth translation vector.

To prevent degenerate solutions where the network assigns uniform weights across the image, we introduce an entropy-based regularization term. This term is computed on a globally normalized version of the weight map, without restricting to valid depth pixels:

$$\tilde{w}(p) = \frac{\exp\left(\frac{L(p)}{\tau_g}\right)}{\sum_q \exp\left(\frac{L(q)}{\tau_g}\right)}. \tag{10}$$

The corresponding entropy loss is defined as

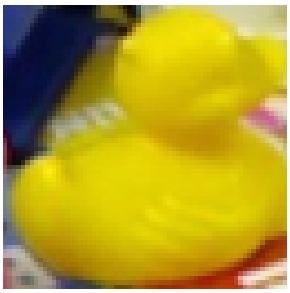
$$\mathcal{L}_{\text{ent}} = -\sum_p \tilde{w}(p) \log \tilde{w}(p). \tag{11}$$

The total training loss is given by

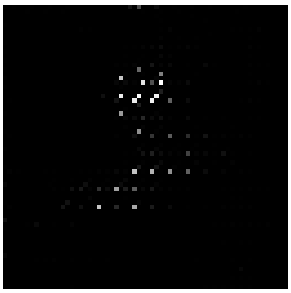
$$\mathcal{L} = \mathcal{L}_{\text{trans}} + \lambda \mathcal{L}_{\text{ent}}, \tag{12}$$

where λ is a small weighting factor ensuring that translation accuracy remains the primary optimization objective.

This learned weighting strategy enables a data-driven selection of geometrically reliable pixels, providing a robust alternative to center-based projection. Figure 3 qualitatively illustrates the proposed pixel-wise weighting mechanism.



(a) RGB crop



(b) Predicted weights

Figure 3. Visualization of the proposed pixel-wise weighting strategy for translation estimation. Whiteness represents a higher weight assigned to that pixel.

As shown in Fig. 3, the network learns to focus on stable object surfaces while suppressing background regions and noisy depth measurements. This behavior contrasts with center-based projection, which relies on a single potentially unreliable pixel.

4. Experiments

4.1. Dataset

We evaluate our approach on the LineMOD dataset [5], a standard benchmark for 6D object pose estimation. The dataset consists of RGB-D image sequences depicting objects under varying viewpoints. Each sequence focuses on a single object instance, with ground-truth 6D pose annotations provided for every frame.

We conduct experiments on all 13 objects available: *ape, benchvise, camera, can, cat, driller, duck, eggbox, glue, holepuncher, iron, lamp, phone*. For each object, we use a train/test random split with 80% of the data used for training.

Two objects, *eggbox* and *glue*, exhibit rotational symmetries. For these classes, we employ symmetry-aware evaluation metrics as described below.

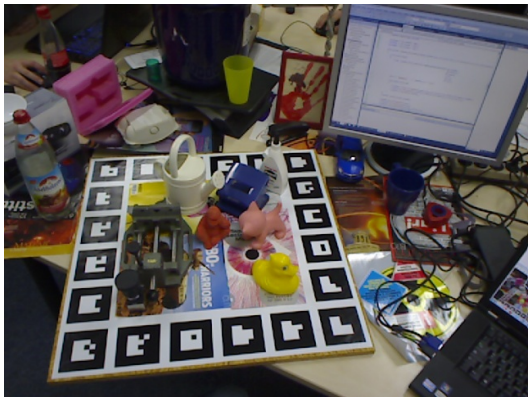


Figure 4. A sample RGB image from LineMod dataset

4.2. Evaluation Metrics

The object detector YOLO, rotation, and translation networks are trained independently, and then the full pipeline is tested and evalueted as follows.

We evaluate pose estimation accuracy using the Average Distance of Model Points (ADD) metric [5]. Given a predicted pose $(\hat{R}, \hat{t})$ and the ground-truth pose (R, t), ADD is defined as the average Euclidean distance between model points transformed by the two poses.

Formally, given a 3D object model $\mathcal{M}$ with $|\mathcal{M}|$ points, the ADD metric is defined as:

$$\text{ADD} = \frac{1}{|\mathcal{M}|} \sum_{\mathbf{x} \in \mathcal{M}} \left\| R\mathbf{x} + \mathbf{t} - \left(\hat{R}\mathbf{x} + \hat{\mathbf{t}} \right) \right\|_2 . \tag{13}$$

For symmetric objects, namely *eggbox* and *glue*, we use the ADD-S variant, which computes the distance to the closest corresponding model point, making the metric invariant to indistinguishable rotations.

$$\text{ADD-S} = \frac{1}{|\mathcal{M}|} \sum_{\mathbf{x} \in \mathcal{M}} \min_{\mathbf{y} \in \mathcal{M}} \left\| R\mathbf{x} + \mathbf{t} - \left(\hat{R}\mathbf{y} + \hat{\mathbf{t}} \right) \right\|_2 , \tag{14}$$

which accounts for indistinguishable rotations by considering the closest model point correspondence.

Out of the 3,166 test images, the baseline fails to produce any pose prediction in 61 cases, as the depth value at the center of the detected bounding box is invalid. In all these cases, the proposed extension successfully predicts the object translation by aggregating information from multiple valid depth pixels. Both methods fail only when the object detector does not produce a bounding box, which occurs in 2 cases.

We report the percentage of samples with ADD below 10% of the object diameter for both the baseline and the proposed extension. We additionally report DenseFusion results on LineMOD as a reference point. Due to differences in data splits and evaluation protocols, this comparison should be interpreted as indicative rather than a direct quantitative comparison.

Table 1 reports the quantitative results obtained on the LineMOD dataset, comparing the proposed extension with the baseline. Results are shown both per object and averaged over all objects. The proposed method consistently improves pose accuracy across all classes.

Figure 5 shows the ADD accuracy as a function of the error threshold, computed over all objects. The curve represents the percentage of test samples whose ADD error falls below a given threshold. Compared to the baseline, the proposed extension achieves substantially higher accuracy across the entire range of thresholds, confirming the effectiveness of selectively integrating depth information for pose estimation.

Table 1. Comparison between the baseline method and the proposed extension on the LineMOD dataset. For symmetric objects (eggbox and glue), ADD-S is reported. DenseFusion† results are reported for reference and taken from the original paper, as they were obtained under a different training and evaluation protocol.

object	baseline	extension	DenseFusion † (per-pixel)	DenseFusion † (iterative refinement)
Ape	13.4	**94.3**	79.5	92.3
Benchvise	55.3	**99.2**	84.2	93.2
Camera	13.6	**96.3**	76.5	94.4
Can	0.84	**99.6**	86.6	93.1
Cat	27.5	**97.9**	88.8	96.5
Driller	37.2	**97.5**	77.7	87.0
Duck	11.2	**95.6**	76.3	92.3
Eggbox (S)	0.0	97.6	**99.9**	99.8
Glue (S)	29.6	97.5	99.4	**100.0**
Holepuncher	0.0	**95.6**	79.0	92.1
Iron	39.4	**98.7**	92.1	97.0
Lamp	45.9	**99.2**	92.3	95.3
Phone	11.2	**96.0**	88.0	92.8
ALL	21.6	**97.3**	86.2	94.3

5. Conclusion

We presented an enhanced framework for 6D pose estimation that leverages depth data to refine translation and rotation regression. By incorporating a symmetry-aware mechanism directly into the manifold of rotations ($SO(3)$), we addressed the inherent ambiguities of symmetric objects. Our framework demonstrates a substantial performance gain over the RGB-only baseline, matching the accuracy of more complex state-of-the-art networks while maintaining a lightweight architecture. This suggests that explicit depth informations are crucial for robust pose estimation.

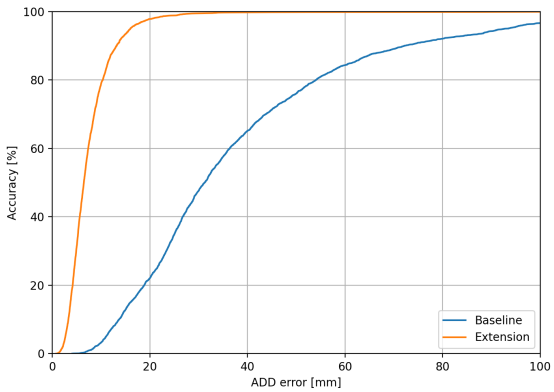


Figure 5. ADD accuracy curves on the LineMOD dataset. The plot reports the percentage of test samples whose ADD error is below a given threshold.

References

[1] Mathieu Aubry, Daniel Maturana, Alexei A. Efros, Bryan C. Russell, and Josef Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 3762–3769, 2014. 2

[2] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 998–1005, 2010. 2

[3] 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*, pages 770–778, 2016. 1, 3

[4] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 858–865, 2011. 2

[5] 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 *Asian conference on computer vision*, pages 548–562, 2012. 1, 5

[6] 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 (ICCV)*, pages 22–29, 2017. 2

[7] Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements, 2024. 1, 2

[8] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. *arXiv preprint arXiv:1804.00175*, 2018. 2

[9] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: a hands-on survey. *IEEE transactions on visualization and computer graphics*, 22(12):2633–2651, 2016. 1

[10] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G. Derpanis, and Kostas Daniilidis. 6-dof object pose from semantic keypoints. *arXiv preprint arXiv:1703.04670*, 2017. 2

[11] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016. 2

[12] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. *arXiv preprint arXiv:1711.08488*, 2017. 2

[13] Reyes Rios-Cabrera and Tinne Tuytelaars. Discriminatively trained templates for 3d object detection: A real time scalable approach. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2048–2055, 2013. 2

[14] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint arXiv:1809.10790*, 2018. 1

[15] Joel Vidal, Chyi-Yeu Lin, and Robert Martí. 6d pose estimation using an improved method based on point pair features. In *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*, pages 405–409, 2018. 2

[16] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3343–3352, 2019. 1, 3

[17] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. 2

[18] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. *arXiv preprint arXiv:1711.10871*, 2017. 2