

HRPose: Real-Time High-Resolution 6D Pose Estimation Network Using Knowledge Distillation

GUAN Qi^{1,2,3}, SHENG Zihao^{1,2,3}, and XUE Shibe^{1,2,3}

(1. Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China)

(2. Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China)

(3. Shanghai Engineering Research Center of Intelligent Control and Management, Shanghai 200240, China)

Abstract — Real-time six degrees-of-freedom (6D) object pose estimation is essential for many real-world applications, such as robotic grasping and augmented reality. To achieve an accurate object pose estimation from RGB images in real-time, we propose an effective and lightweight model, namely high-resolution 6D pose estimation network (HRPose). We adopt the efficient and small HRNetV2-W18 as a feature extractor to reduce computational burdens while generating accurate 6D poses. With only 33% of the model size and lower computational costs, our HRPose achieves comparable performance compared with state-of-the-art models. Moreover, by transferring knowledge from a large model to our proposed HRPose through output and feature-similarity distillations, the performance of our HRPose is improved in effectiveness and efficiency. Numerical experiments on the widely-used benchmark LINEMOD demonstrate the superiority of our proposed HRPose against state-of-the-art methods.

Key words — 6D pose estimation, Keypoint detection, Knowledge distillation, Lightweight model.

I. Introduction

Object pose estimation aims to obtain the six degrees-of-freedom (6D) pose of an object in a camera coordinate, and its real-time application is crucial for autonomous driving [1]–[4], augmented reality [5], robotic grasping [6], and so forth. For instance, fast and accurate 6D pose estimation is essential in Amazon Picking Challenge [7], where a robot needs to pick objects from a warehouse shelf. Although methods [8], [9] that rely on depth images for this task are more robust,

estimating object poses from RGB images is more attractive for actual scenarios, in terms of hardware cost and availability. This problem is still challenging due to variations in appearance and cluttered environments.

Traditional methods [10], [11] compute object poses by establishing maps between object images and their actual model through feature points or template matching. They rely on hand-crafted features, which are sensitive to image variations and background clutters. Nowadays, with the development of deep learning, deep convolutional neural networks (CNNs) have achieved significant progresses in 6D object pose estimation.

To achieve an efficient pose estimation, existing methods [12]–[15] first use CNNs to detect predefined two-dimensional (2D) keypoints and then recover object poses via a perspective-n-point (PnP) algorithm [16]. Among these methods, Tekin *et al.* [13] employed the object detector YOLOv2 [17] to directly regress 2D locations of keypoints which can achieve almost the fastest speed in pose estimation. However, directly regressing the keypoints coordinates makes the CNN hard to converge, which results in degradation of accuracy. Tremblay *et al.* [18] proposed a multistage architecture to estimate pixel-wise heatmaps of 2D keypoints. Peng *et al.* [15] proposed pixel-wise unit vectors as a representation of keypoints. However, such dense predictions lead to an increase in model size and computational complexity, which restricts them from practical applications.

Recently, neural networks with high accuracies, small model sizes, and low computational costs have at-

tracted much attention for their demands in resource-limited devices, such as embedded systems [19]. For such systems, knowledge distillation [20] has been widely studied for its simplicity and effectiveness. Its main idea is to improve the performance of a small network by transferring the knowledge from a large teacher network. Felix *et al.* [21] proposed an improved knowledge distillation to get a faster version of a 2D keypoints detector based on YOLO6D [13] for object pose estimation, where the output of a trained teacher network is simply transferred. However, the unreliable teacher network introduces noises in training and makes the student network fail to meet requirements on accuracies in real-world applications.

An ideal solution to 6D object pose estimation should satisfy some actual conditions such as textureless appearance, heavy clutter scenes, and environmental variations. Also, it should meet the speed requirement for real-time tasks (e.g. 30 frames per second [17]). To this end, we propose a simple and efficient model, namely high-resolution 6D pose estimation network (HRPose) which predicts keypoints from the high-resolution feature representation in a bottom-up method. HRPose takes the small HRNetV2-W18 [22] as the backbone and is able to retain spatial positions as well as deep semantic information, which leads to more accurate pose estimations.

To further improve the estimation accuracy of HRPose without a performance degradation, we propose a novel method, namely integrated knowledge distillation. We propose to align the output of the teacher and student network at a pixel level, which is named output distillation. We further apply a feature-similarity distillation, which intends to transfer the prior information from the feature maps. The similarity matrix is used to represent the rich semantic information from the feature maps. By minimizing the distance of similarity matrices between the teacher and student networks, the distribution of the feature maps of the student network can approach that of the teacher network.

Our contribution can be summarized as follows:

1) We propose an efficient high-resolution pose estimation network for 6D object pose estimation, which can achieve comparable performance but has about 33% parameters compared with the state-of-the-art methods on the widely-used LINEMOD dataset.

2) To further improve the accuracy of HRPose, we propose an integrated knowledge distillation method, which transfers both the structure information from the outputs and the feature maps of a trained teacher network, for achieving a mean gain of 1.66% on accuracy in the average distance of model points metric.

3) Our approach is highly accurate and fast enough

(33 ms per image) to achieve the speed requirement in real-time tasks.

II. The Related Work

In this section, we review related works on RGB-based 6D object pose estimation and knowledge distillation.

1. 6D object pose estimation

Recently, the estimation of 6D object poses including 3D locations and 3D orientations has been an active topic. Previous methods mainly rely on matching techniques [11] or local feature descriptors [10], [23] which are not robust to variations of appearances and environments.

Similar to other computer vision tasks, learning-based methods have achieved significant progresses. Given an image, some previous works [6], [24] rely on the power of deep neural networks and directly estimate object poses in a single shot. However, the direct regression of 6D poses is still difficult due to the non-linearity of the rotation space [15], which requires a pose-refinement algorithm to get an accurate 6D pose.

Some recent methods [13], [15], [25], [26] first predict 2D keypoints of objects and then compute 6D poses through 2D-3D correspondences with a PnP algorithm. In other words, the problem of 6D pose estimation is transformed into the problem of keypoint detections. In this kind of methods, BB8 [12] detects the objects of interest using segmentation and then predicts 2D keypoint coordinates from detected regions. PVNet [15] is proposed to use pixel-wise unit vectors as a representation of keypoints and use the predicted vectors to vote for keypoint locations through RANSAC [27]. DPOD [25] estimates dense multi-class 2D-3D correspondence maps between an input image and available 3D models. HybridPose [26] utilizes keypoints, edge vectors, and symmetry correspondences as the representation of 6D poses. HybridPose achieves a state-of-the-art performance with an additional refinement sub-module.

Despite the accuracy of CNN on pose estimation increases, this relies on its large scalability and time-consuming computations (e.g., VGG [28] or ResNet [29]) which ignores model efficiencies. A few recent works focus on improving the efficiency. Tekin *et al.* [13] employed a lightweight detector YOLOv2 [17] to this task and achieved almost the fastest speed of estimation. However, this method made predictions based on a low-resolution feature map and was not sufficiently precise to meet the accuracy requirement in practical scenarios.

2. Knowledge distillation

CNNs are expensive in terms of computations and memories. Deeper networks are preferred for accuracy,

while smaller networks are widely used due to their efficiency [30]–[32]. So model compression becomes a focus, which intends to speed up running times while maintaining accuracies. Knowledge distillation [20] is one of the model compression methods, which transfers knowledge from an accurate teacher network to a compact student network. By utilizing extra supervision information of a trained teacher network, the student network can achieve a better performance.

Bucila *et al.* [33] proposed an algorithm to train a single small neural network by mimicking the output of an ensemble of models. Hinton [20] proposed a knowledge distillation (KD) method using softmax outputs of a teacher network as an extra supervision. Since the dimension of both outputs is identical, such an output distillation method can be applied to any pair of networks.

For better utilizing the information contained in the teacher network, some feature distillation methods [34]–[37] have been proposed, which transfer intermediate feature representations. Romero *et al.* [34] proposed a hint learning method that aligns the intermediate feature maps between the teacher and student network. Zagoruyko *et al.* [35] proposed to force the student network to mimic the attention maps of a powerful teacher network. Liu *et al.* [37] proposed to distill the pixel-level and structure information from the teacher network simultaneously. Such feature distillation schemes can be combined with an output distillation to improve the performance of the student network.

Given the effectiveness of knowledge distillation, we propose a novel integrated knowledge distillation that contains output and feature-similarity distillations to enhance the performance of HRPose in this work.

III. High-Resolution 6D Pose Estimation Network Using Knowledge Distillation

Given an RGB image, the task of 6D pose estimation is to detect objects and estimate their 6D poses. The 6D pose can be denoted as a rigid transformation $[\mathbf{R}, \mathbf{t}; 0, 1]$ from the object coordinate to the camera coordinate, where \mathbf{R} is a 3×3 rotation matrix and \mathbf{t} is a 3×1 translation vector.

We propose a framework of HRPose with knowledge distillation for real-time 6D object pose estimation as shown in Fig.1. We first train a large teacher network that shares the same architecture as the proposed HRPose. Then, we train the proposed HRPose with the assistance of knowledge learned from the teacher network. HRPose takes the small HRNetV2-W18 [22] as the backbone, which has fewer convolutional layers than the teacher network. Knowledge distilla-

tion works in this step which transfers both the knowledge of the output and the feature maps from the teacher network to the HRPose.

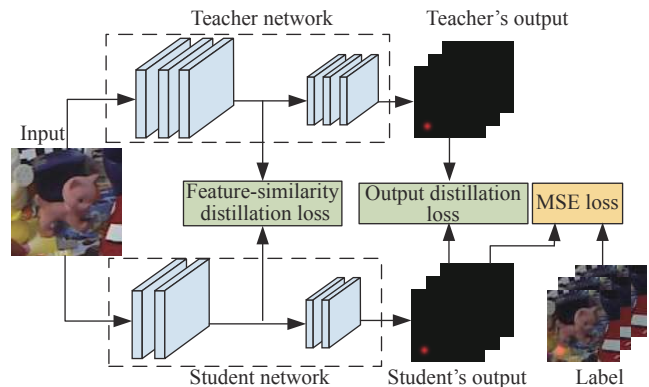


Fig. 1. An overview of the proposed HRPose with knowledge distillation. In the training process, we keep the large teacher network fixed and only optimize the student network. The student network is trained with two distillation terms (feature-similarity loss and output distillation loss) and the pose estimation loss. The trained student network can perform an efficient object pose estimation.

In this section, we start with an introduction to the proposed HRPose and then describe the details of the knowledge distillation.

1. High-resolution pose estimation network

We propose a two-step pipeline for object pose estimation: we first detect 2D object keypoints using CNNs in a bottom-up method as shown in Fig.2 and then calculate 6D poses via a PnP algorithm. We select the 8 vertices of the 3D bounding box and the centroid of an object as keypoints.

Given an RGB image size of $H \times W \times 3$, HRPose processes it using a fully-convolutional architecture and predicts a set of 2D belief maps \mathbf{H} of keypoint locations (Fig.2(c)) and a set of 2D vector fields \mathbf{L} , which represents the degree of correlation between the corners and the centroid (Fig.2(d)) for each object. The set $\mathbf{H} = (\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_I)$ has I belief maps with a size $\frac{H}{4} \times \frac{W}{4}$, where each belief map \mathbf{H}_i represents the location confidence of the i -th keypoint. The set $\mathbf{L} = (\mathbf{L}_1, \mathbf{L}_2, \dots, \mathbf{L}_J)$ has J vector fields with a size $\frac{H}{4} \times \frac{W}{4} \times 2$. Each pair of vertex and the centroid generate a vector field, where each image location in \mathbf{L}_j denotes a 2D vector. Finally, the belief maps and the vector fields are parsed using a post-processing algorithm to output the locations of 2D keypoints.

The network of HRPose, shown in Fig.2 (a), consists of three parts: a stem composed of two 3×3 strided convolutions decreasing the resolution to $\frac{1}{4}$, a backbone network that extracts semantic features, and a regressor using the feature of the backbone to estim-

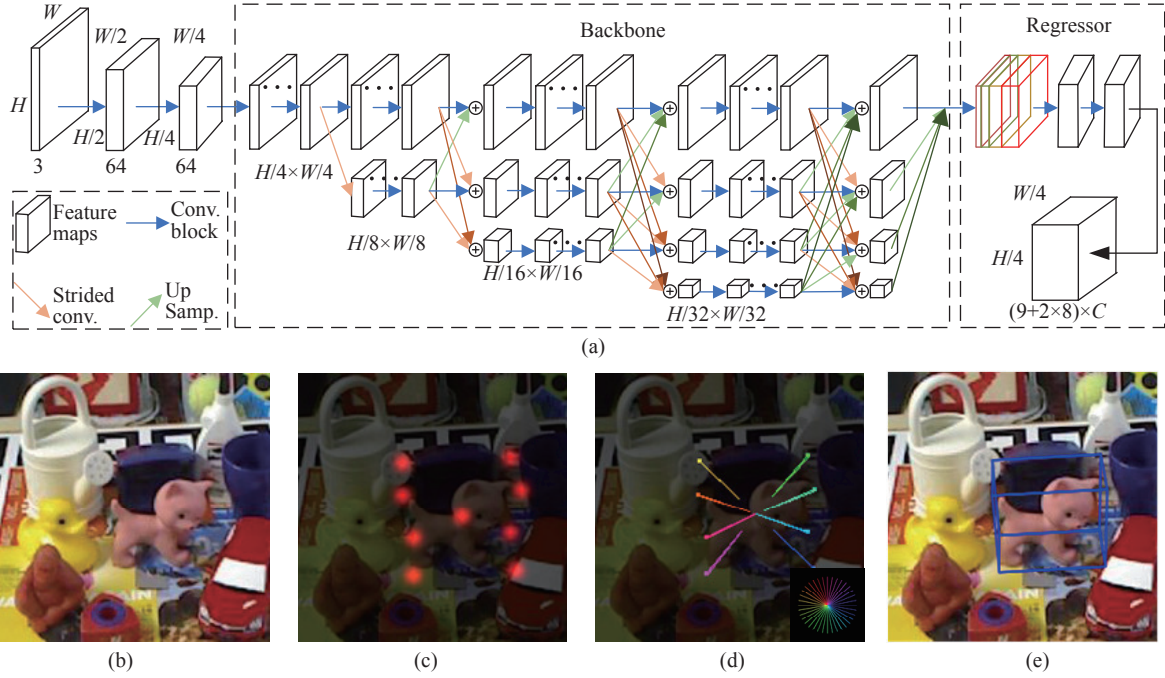


Fig. 2. Overview: (a) The architecture of HRPose. In the output of the network, the first nine heatmaps represent the predicted belief maps for keypoints and the latter 16 heatmaps represent the predicted vector fields; (b) An image in the LINEMOD dataset; (c) Belief maps; (d) Vector fields; (e) Predictions of the 2D location of the corners of the projected 3D bounding boxes in the image.

ate the belief maps and the vector fields. The regressor consists of three 1×1 convolutions, and outputs a $\frac{H}{4} \times \frac{W}{4} \times (K \times C)$ tensor representing belief maps and a $\frac{H}{4} \times \frac{W}{4} \times ((K-1) \times 2 \times C)$ tensor representing vector fields. Here, C and K denote the number of the classes of objects and keypoints for each object, respectively. Considering the positive impact of a high-resolution representation on the keypoint detection, we adopt the small HRNetV2-W18 [22] as the backbone, which can maintain a high-resolution representation to achieve a precise keypoint location.

An individual belief maps \mathbf{H}_i is generated by centering a Gaussian kernel around the labeled position. The value at a location $\mathbf{p} \in \mathbb{R}^2$ in \mathbf{H}_i is defined as,

$$\mathbf{H}_i(\mathbf{p}) = \exp\left(\frac{-\|\mathbf{p} - \mathbf{x}_i\|_2^2}{2\sigma^2}\right) \quad (1)$$

where $\mathbf{x}_i \in \mathbb{R}^2$ denotes the ground-truth position of the i -th keypoint in the image and σ denotes the standard deviation.

We use a unit vector $\mathbf{v}_j(\mathbf{p})$ to represent the direction from the j -th vertex to the centroid of the corresponding object. Let \mathbf{x}_j , \mathbf{x}_{cen} denote the ground-truth position of the j -th vertex of the 3D bounding box and the centroid in the image, respectively. If a point \mathbf{p} lies around the j -th vertex, the value at $\mathbf{L}_j(\mathbf{p})$ is a unit vector that points from the vertex to the centroid; for all other points, the vector is zero-valued. Therefore, the

value at the location \mathbf{p} in the ground-truth vector field \mathbf{L}_j is defined as,

$$\mathbf{L}_j(\mathbf{p}) = \begin{cases} \mathbf{v}_j(\mathbf{p}), & \text{if } \mathbf{p} \in \mathcal{N}(\mathbf{x}_j) \text{ or } \mathbf{p} = \mathbf{x}_j \\ \mathbf{0}, & \text{otherwise} \end{cases} \quad (2)$$

Here, $\mathbf{v}_j(\mathbf{p}) = (\mathbf{x}_j - \mathbf{x}_{\text{cen}}) / \|\mathbf{x}_j - \mathbf{x}_{\text{cen}}\|_2$ is the unit vector. $\mathcal{N}(\mathbf{x}_j)$ denotes the local neighborhood containing pixels within a 3-pixel radius of the ground-truth vertex \mathbf{x}_j .

We use the mean squared error for learning the belief maps and the vector fields. The overall objective loss function is defined as,

$$\mathcal{L}_{\text{mse}} = \frac{1}{I} \sum_{i=1}^I \|\mathbf{H}_i - \tilde{\mathbf{H}}_i\|_2^2 + \frac{1}{J} \sum_{j=1}^J \|\mathbf{L}_j - \tilde{\mathbf{L}}_j\|_2^2 \quad (3)$$

where \mathbf{H}_i and $\tilde{\mathbf{H}}_i$ are the i -th ground-truth and the predicted belief map, respectively. \mathbf{L}_j and $\tilde{\mathbf{L}}_j$ are the j -th ground-truth and predicted vector fields, respectively.

After processing an input image with the proposed network, we can extract 2D keypoint positions from the estimated belief maps using a greedy inference algorithm [18]. Since each belief map represents the keypoints of an unknown number of instances of the same type, it is necessary to assemble the detected keypoints to form individual objects according to the predicted vector fields. We take the local peaks from the predicted belief maps above a threshold as keypoints and

then group the keypoints into object instances according to the predicted vector fields. For each vertex, we then compare the predicted vector field with the direction from the vertex to the object centroid and assign the detected vertex to the closest object centroid within a certain angular threshold.

When the vertices of each object instance are detected, a PnP [16] algorithm can use the camera intrinsic, the 3D keypoints, and the corresponding projected vertices to compute the final 6D pose.

2. Integrated knowledge distillation

To further improve the pose estimation accuracy without a model performance degradation, we introduce and integrate the knowledge distillation technique, named integrated knowledge distillation. A brief outline of the training method is shown in Fig.1. We expect the student network to learn not only the information provided by the ground-truth labels, but also the finer structure knowledge encoded by the teacher network. Let T and S denote a teacher network and a student network, respectively.

We adopt the output distillation and the feature-similarity distillation to help the training of HRPose jointly. The purpose of the output distillation is intuitive: if the output of a student is similar to that of the teacher, the performance of the student should be similar to the teacher. Transferring the knowledge from the output layer forces the student network to produce a similar output as that of the teacher which is useful to improve the performance of the student network. Here, mean squared error (MSE) is used as a loss function to measure the divergence between the teacher and student outputs. Therefore, the output distillation loss function is formulated as:

$$\mathcal{L}_{\text{od}} = \frac{1}{I} \sum_{i=1}^I \|\mathbf{H}_i^s - \mathbf{H}_i^t\|_2^2 + \frac{1}{J} \sum_{j=1}^J \|\mathbf{L}_j^s - \mathbf{L}_j^t\|_2^2 \quad (4)$$

Here, \mathbf{H}_i^s and \mathbf{H}_i^t denote the belief maps for the i -th keypoint predicted by the pre-trained teacher model and the in-training student model, respectively. Similarly, \mathbf{L}_j^s and \mathbf{L}_j^t denote the vector fields for the j -th keypoint predicted by the teacher and the student models, respectively.

The feature-similarity distillation aims to transfer more structured information from the teacher network to the student network. Generally, features in certain regions share the same properties related to the task. The trained teacher network has extracted certain features related to the object pose estimation task from the original input. The information from the feature maps of the teacher network is valuable for the student network since it provides the student network with

guidance on the keypoint detection. Therefore, we apply the feature-similarity distillation to make the student feature maps similar to that of the teacher.

We use $\mathbf{F} \in \mathbb{R}^{C \times H \times W}$ to denote the output feature map of a layer in the CNN where C is the total number of channels and $H \times W$ is the spatial dimensions. Let \mathbf{F}^t and \mathbf{F}^s denote the feature maps from certain layers of the teacher and student networks, respectively. In our method, we assume that the spatial dimensions of \mathbf{F}^t and \mathbf{F}^s are identical. The function loss in the feature-similarity distillation is written as

$$\mathcal{L}_{fs} = \|\mathbf{G}_s - \mathbf{G}_t\|_F^2 \quad (5)$$

Here, $\mathbf{G} \in \mathbb{R}^{HW \times HW}$ is the similarity matrix, with each entry g_{ij} defined as,

$$g_{ij} = \frac{\mathbf{f}_i^T \mathbf{f}_j}{\|\mathbf{f}_i\|^2 \|\mathbf{f}_j\|^2} \quad (6)$$

where $\mathbf{f}_i \in \mathbb{R}^C$ denotes a feature vector extracted from the i -th ($i = 1, 2, \dots, H \times W$) spatial location of the feature map \mathbf{F} . Each item g_{ij} represents the similarity between the i -th feature vector and the j -th feature vector. With the help of feature-similarity distillation, the student is trained to minimize the divergence between the student and teacher feature maps. Feature-similarity distillation provides more supervision information for student models. In our experiments, we choose to align the feature maps extracted from the backbone because the abstract semantics makes more sense for keypoint detection.

Therefore, the student network is trained to optimize the following loss function:

$$\mathcal{L} = \mathcal{L}_{\text{mse}} + \lambda_1 \mathcal{L}_{\text{od}} + \lambda_2 \mathcal{L}_{fs} \quad (7)$$

Here, λ_1 and λ_2 are tunable parameters to balance the standard MSE loss and the distillation loss.

Fig.1 summarizes the training of the knowledge transfer framework. The backbone of the teacher network and the student network are HRNetV2-18 and small HRNetV2-18, respectively. We first train a teacher network to optimize (3) without any extra loss. Then, we train a target student network to minimize (7), with the knowledge distillation from the teacher network to the target network being conducted throughout the entire training process. At a test time, we only use the efficient and cost-effective HRPose while throwing away the large teacher network, since the target network already extracts the teacher's knowledge.

IV. Experimental Results and Analysis

1. Dataset and training strategy

To validate the proposed method, we perform ex-

periments on the LINEMOD dataset [38], which is a standard benchmark for 6D object pose estimation. It provides about 15000 actual images with annotated 6D poses of 13 texture-less objects in heavily cluttered scenes. The precise 3D models of the corresponding objects are also available. We follow prior works [13] to use around 15% of the LINEMOD examples for training and 85% for testing. To prevent overfitting, we add synthetic images to the training set following [15]. We render 10000 images for each object and synthesize another 10000 images by the “Cut and paste” strategy as shown in Fig.3. The background of all synthetic images is randomly sampled from the SUN397 dataset [39]. Besides, we perform online data augmentation including random blur, color jittering, and rotation (± 30 degrees) during training.

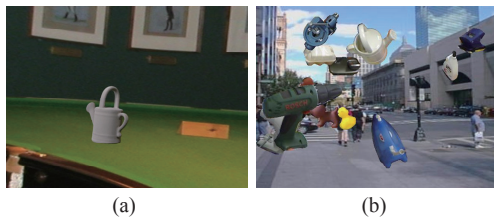


Fig. 3. An illustration of the synthetic images. (a) The rendered image whose pose is uniformly sampled; (b) The synthetic image using “Cut and paste.”

In training, we adopt the ADAM optimizer [40] with a mini-batch size of 32. The initial learning rate is set to 0.0001 and halves every 20 epochs. All models are trained for 120 epochs. Our implementation is based on PyTorch with TITAN XP GPU. In our experiment, we set λ_1 and λ_2 to be 0.5 and 0.00005. For simplicity, the proposed teacher network is named “Teacher,” and the student networks with and without knowledge distillation are named “HRPose+KD” and “HRPose,” respectively.

2. Evaluation metrics

We use two common metrics for evaluation: the average distance of model points (ADD) metric [38] and 2D projection metric [41]. The ADD metric is defined as an average distance between the transformed 3D model points using the ground-truth pose and the estimated pose. For the ADD metric, we identify a pose to be correct if the average distance is less than 10% of the object’s diameter. The 2D projection metric computes the mean distance between the 2D projections of the object’s 3D mesh vertices using the estimated and the ground truth pose. A pose is identified to be correct if the distance is less than 5 pixels when using the 2D projection metric.

Given the ground-truth rotation \mathbf{R} and the translation \mathbf{t} , the predicted rotation $\tilde{\mathbf{R}}$ and the translation $\tilde{\mathbf{t}}$, the ADD metric is calculated as,

$$\text{ADD} = \frac{1}{m} \sum_{\mathbf{x} \in \mathcal{M}} \|(\mathbf{R}\mathbf{x} + \mathbf{t}) - (\tilde{\mathbf{R}}\mathbf{x} + \tilde{\mathbf{t}})\| \quad (8)$$

where \mathcal{M} represents the set of 3D model points and m is the number of points. For symmetric objects, the average closest point distance (ADD-S) [6] is used to evaluate the performance of 6D pose estimation. The accuracy of pose estimation is defined as the percentage of correct pose estimations. Besides, the number of model parameters and the FLOPs (floating point operations) are adopted to evaluate the model efficiency.

3. Comparison with state-of-the-art methods

We compare the proposed method with the state-of-the-art RGB only methods without any refinement using both ADD metric (shown in Table 1 [13], [15], [25], [26], [42], [43]) and 2D projection error (shown in Table 2 [13], [15], [43]). Since some methods do not report their 2D projection accuracy, we do not include them in Table 2.

Table 1. Quantitative evaluation of 6D pose using ADD(-S) metric on the LINEMOD dataset (The boldface numbers denote the best overall methods; Objects with * are symmetric)

	Tekin [13]	DPOD [25]	PVNet [15]	CDPN [42]	HybridPose [26]	GDR-Net [43]	Teacher	HRPose	HRPose+KD
Ape	21.62	53.28	43.62	64.38	63.10	76.29	68.26	61.21	65.36(+4.15)
Benchvise	81.80	95.34	99.90	97.77	99.90	97.96	99.42	95.53	97.38(+1.85)
Cam	36.57	90.36	86.86	91.67	90.40	95.29	89.78	84.89	85.98(+1.09)
Can	68.80	94.10	95.47	95.87	98.50	98.03	98.62	93.60	94.88(+1.28)
Cat	41.82	60.38	79.34	83.83	89.40	93.21	90.02	86.03	87.33(+1.30)
Driller	63.51	97.72	96.43	96.23	98.50	97.72	98.91	96.23	96.73(+0.50)
Duck	27.23	66.01	52.58	66.76	65.00	80.28	72.72	67.95	71.52(+3..57)
Eggbox*	69.58	99.72	99.15	99.72	100.00	99.53	100.00	98.97	99.06(+0.09)
Glue*	80.02	93.83	95.66	99.61	98.80	98.94	98.65	97.00	97.49(+0.49)
Holepuncher	42.63	65.83	81.92	85.82	89.70	91.15	84.64	78.10	80.55(+2.45)
Iron	74.97	99.80	98.88	97.85	100.00	98.06	98.98	95.50	95.90(+0.40)
Lamp	71.11	88.11	99.33	97.89	99.50	99.14	99.42	96.64	97.70(+1.06)
Phone	47.74	74.24	92.41	90.75	94.90	92.35	91.93	86.65	89.91(+3.26)
Average	55.95	82.98	86.27	89.86	91.36	93.69	91.64	87.55	89.21(+1.66)

Table 2. Quantitative evaluation of 6D pose using 2D projection metric on the LINEMOD dataset

	YOLO6D [13]	PVNet [15]	GDR-Net [43]	Teacher	HRPose	HRPose+KD
Ape	92.10	99.23	98.29	98.86	97.99	98.47(+0.48)
Benchvise	95.06	99.81	99.32	99.03	98.35	99.13(+0.78)
Cam	93.24	99.21	99.41	99.41	99.31	99.51(+0.20)
Can	97.44	99.90	99.51	99.70	98.33	99.02(+0.69)
Cat	97.41	99.30	99.60	99.30	99.20	99.30(+0.10)
Driller	79.41	96.92	98.22	98.41	97.32	98.32(+1.00)
Duck	94.65	98.02	98.97	98.21	98.21	98.68(+0.47)
Eggbox*	90.33	99.34	98.87	99.53	99.15	99.15(+0.00)
Glue*	96.53	98.45	99.42	99.23	99.23	99.42(+0.19)
Holepuncher	92.86	100.00	99.62	99.81	97.14	97.43(+0.29)
Iron	82.94	99.18	97.62	99.28	96.93	97.44(+0.51)
Lamp	76.87	98.27	96.64	98.66	96.64	97.79(+1.15)
Phone	86.07	99.42	97.92	99.33	97.89	98.85(+0.96)
Average	90.38	99.00	98.72	99.14	98.13	98.67(+0.46)

Table 1 indicates that the proposed simple “HRPose” achieves 87.55% pose estimation accuracy without any extra information on average and outperforms PVNet [15]. With the help of knowledge distillation, the overall pose estimation accuracy of “HRPose” raises from 87.55% to 89.21% in the ADD metric and

from 98.13% to 98.67% in the 2D projection metric. Although “HRPose+KD” does not outperform HybridPose [26], it still achieves comparable pose estimation accuracy with merely 33% parameters and 20.6% FLOPs of HybridPose [26], as shown in Table 3, in which M and G stands for 10^6 and 10^9 , respectively.

Table 3. Comparison of different methods in the backbone, model size (the number of model parameters), and computational cost (FLOPs)

Methods	Backbone	#Params	FLOPs
YOLO6D [13]	YOLOv2	50.5M	26.1G
PVNet [15]	ResNet-18	12.9M	72.7G
HybridPose [26]	ResNet-18	12.9M	75.2G
GDR-Net [43]	ResNet-34	33.5M	–
Teacher	HRNetV2-W18	9.7M	23.2G
HRPose	Small HRNetV2-W18	4.2M	15.5G

From both Table 1 and Table 2, we can observe that the distilled small network achieves a better 6D pose estimation performance than its corresponding baselines using both ADD metric and 2D projection metric. Especially, the “HRPose+KD” outperforms the baseline model by a significant margin of 4.15% on “Ape” using the ADD metric.

Fig.4 provides some qualitative results on the LINEMOD dataset. It can be observed that HRPose can achieve robust and reliable pose estimation with various background clutters.

We calculate the number of network parameters (#Params) and the sum of float point operations to measure the model efficiency. The resolution of the input RGB image is 640×480 . As shown in Table 3, HRPose has the minimal model size and the lowest computational complexity. Note that, GDR-Net [43] needs a detector to obtain the object region and we do not count the model size of the detector. The distilled HRPose model achieves comparable even better results

with only about 8.3% of the model size of YOLO6D, 33% of the model size of HybridPose [26]. Although the accuracy of our model is slightly lower than GDR-Net, it is still a comparable result (89.21% in the ADD metric with 98.67% in the 2D projection metric) with almost 13%

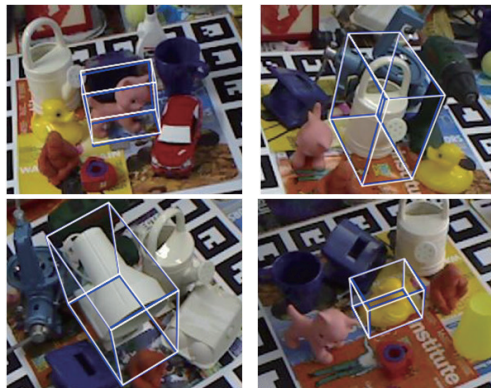


Fig. 4. Visualization of results on the LINEMOD dataset. White and blue bounding boxes represent the ground-truth and estimated poses respectively.

of the model size of GDR-Net. This means our model achieves relative comparable results with a cheaper deployment cost. Also, the proposed method can run at 33 fps on a GTX 2080 GPU which can satisfy the requirement for real-time object pose estimation.

4. Ablation study

To investigate the effectiveness of different components of our distillation scheme, we conduct an ablation study on the object “Cat” from LINEMOD dataset. From Table 4, we can observe that: i) With the

output-distillation (\mathcal{L}_{od}) and the feature-similarity distillation (\mathcal{L}_{fs}), it achieves 1.02% and 0.79% improvements in term of ADD metric, respectively. ii) With the combination of the output-distillation and the feature-similarity, the proposed model achieves an improvement of 1.30% (from 86.03% to 87.33%) accuracy. These observations indicate that the two distillation schemes that we present can improve the accuracy of the network, which can be combined to help the student network to obtain a better performance.

Table 4. Ablation study of different components of the loss in the proposed method (\mathcal{L}_{od} : output distillation; \mathcal{L}_{fs} : feature-similarity distillation)

	W/O distillation	\mathcal{L}_{od}	\mathcal{L}_{fs}	$\mathcal{L}_{od}+\mathcal{L}_{fs}$
ADD	86.03	87.05(+1.02)	86.83(+0.80)	87.33(+1.30)

Besides, we perform an ablation study on the setting of the hyperparameters λ_1 and λ_2 . For simplicity, we fix λ_1 to be 0.5. Table 5 reports the impact of the hyperparameters on the training process using ADD metric, where λ_2 increases from 0.00005 to 0.001. Then we fix λ_2 to be 0.00005 and let λ_1 vary from 0.05 to 1. It can be observed that the proposed HRPose achieves

a higher accuracy varying from 0.00005 to 0.0005 compared with the ADD accuracy obtained with $\lambda_2=0$. If $\lambda_1=0.5, \lambda_2=0.00005$, the proposed model achieves the highest accuracy. However, if the hyperparameters is set too large (e.g. $\lambda_1=1$ or $\lambda_2=0.001$), knowledge distillation will disrupt the training of the student network which leads to the failure of convergence.

Table 5. Ablation study on the selection of hyperparameters λ

$\lambda_1 = 0.5$	λ_2				
	0	0.00005	0.0001	0.0005	0.001
ADD	87.05	87.33	87.23	87.14	85.73
$\lambda_2 = 0.00005$	λ_1				
	0	0.05	0.1	0.5	1
ADD	86.83	87.23	87.14	87.33	86.03

V. Conclusions

In this paper, we have proposed a simple and lightweight high-resolution 6D pose estimation network (HRPose) by adopting the small HRNet as a feature extractor. This design is helpful to reduce computation burdens as well as to guarantee a high accuracy of pose estimation. With about 33% parameters of the state-of-the-art models, our HRPose can achieve a comparable performance on the widely-used benchmark LINEMOD dataset. To enhance the performance of HRPose, we have also proposed a novel knowledge distillation technique that transfers the structure knowledge from a large and complex network to the proposed HRPose. With the help of the proposed knowledge distillation method, the performance of the proposed HRPose can be further improved for 6D object pose estimation. Our method is highly accurate and fast enough (33 frames per second) to satisfy the real-time requirement.

References

- [1] D. Xu, D. Anguelov, and A. Jain, “PointFusion: Deep sensor fusion for 3D bounding box estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp.244–253, 2018.
- [2] Z. Sheng, S. Xue, Y. Xu, *et al.*, “Real-time queue length estimation with trajectory reconstruction using surveillance data,” in *Proceedings of 2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Shenzhen, China, pp.124–129, 2020.
- [3] Z. Sheng, L. Liu, S. Xue, *et al.*, “A cooperation-aware lane change method for autonomous vehicles,” *arXiv preprint*, arXiv: 2201.10746, 2022.
- [4] Z. Sheng, Y. Xu, S. Xue, *et al.*, “Graph-based spatial-temporal convolutional network for vehicle trajectory prediction in autonomous driving,” *IEEE Transactions on Intelligent Transportation Systems*, early access, DOI: [10.1109/TITS.2022.3155749](https://doi.org/10.1109/TITS.2022.3155749), 2022.
- [5] E. Marchand, H. Uchiyama, and F. Spindler, “Pose estimation for augmented reality: A hands-on survey,” *IEEE Trans. on Visualization and Computer Graphics*, vol.22, no.12, pp.2633–2651, 2015.
- [6] Y. Xiang, T. Schmidt, V. Narayanan, *et al.*, “PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes,” in *Proceedings of 2018 Robotics: Science and Systems Conference*, Pittsburgh, Pennsylvania, USA, arXiv:1711.00199, 2018.
- [7] N. Correll, K. E. Bekris, D. Berenson, *et al.*, “Analysis and

- observations from the first amazon picking challenge,” *IEEE Transactions on Automation Science and Engineering*, vol.15, no.1, pp.172–188, 2018.
- [8] C. Wang, D. Xu, Y. Zhu, *et al.*, “DenseFusion: 6D object pose estimation by iterative dense fusion,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp.3343–3352, 2019.
- [9] Y. He, W. Sun, H. Huang, *et al.*, “PVN3D: A deep point-wise 3D keypoints voting network for 6DoF pose estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp.11632–11641, 2020.
- [10] David G Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the IEEE International Conference on Computer Vision*, Kerkyra, Greece, vol.2, pp.1150–1157, 1999.
- [11] S. Hinterstoisser, C. Cagniart, S. Ilic, *et al.*, “Gradient response maps for real-time detection of textureless objects,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.34, no.5, pp.876–888, 2011.
- [12] M. Rad and V. Lepetit, “BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth,” in *Proceedings of 2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, pp.3848–3856, 2017.
- [13] B. Tekin, S. N. Sinha, and P. Fua, “Real-time seamless single shot 6D object pose prediction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp.292–301, 2018.
- [14] M. Oberweger, M. Rad, and V. Lepetit, “Making deep heatmaps robust to partial occlusions for 3D object pose estimation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, pp.119–134, 2018.
- [15] S. Peng, Y. Liu, Q. Huang, *et al.*, “PVNet: Pixel-wise voting network for 6DoF pose estimation,” in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp.4556–4565, 2019.
- [16] V. Lepetit, F. Moreno-Noguer, and P. Fua, “EPnP: An accurate O(n) solution to the PnP problem,” *International Journal of Computer Vision*, vol.81, no.2, pp.155–166, 2009.
- [17] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp.6517–6525, 2017.
- [18] J. Tremblay, T. To, B. Sundaralingam, *et al.*, “Deep object pose estimation for semantic robotic grasping of household objects,” in *Proceedings of the 2nd Conference on Robot Learning*, Zurich, Switzerland, pp.306–316, 2018.
- [19] G. Du, K. Wang, S. Lian, *et al.*, “Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: A review,” *Artificial Intelligence Review*, vol.54, no.3, pp.1677–1734, 2021.
- [20] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv preprint*, arXiv: 1503.02531, 2015.
- [21] H. Felix, W. M. Rodrigues, D. Macêdo, *et al.*, “Squeezed deep 6DoF object detection using knowledge distillation,” in *Proceedings of the International Joint Conference on Neural Networks*, Glasgow, UK, pp.1–7, 2020.
- [22] J. Wang, K. Sun, T. Cheng, *et al.*, “Deep high-resolution representation learning for visual recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.43, no.10, pp.3349–3364, 2020.
- [23] E. Rublee, V. Rabaud, K. Konolige, *et al.*, “ORB: An efficient alternative to SIFT or SURF,” in *Proceedings of 2011 International Conference on Computer Vision*, Barcelona, Spain, pp.2564–2571, 2011.
- [24] W. Kehl, F. Manhardt, F. Tombari, *et al.*, “SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again,” in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, pp.1530–1538, 2017.
- [25] S. Zakharov, I. Shugurov, and S. Ilic, “DPOD: 6D pose object detector and refiner,” in *Proceedings of the IEEE International Conference on Computer Vision*, Seoul, Korea (South), pp.1941–1950, 2019.
- [26] C. Song, J. Song, and Q. Huang, “HybridPose: 6D object pose estimation under hybrid representations,” in *Proceeding of the Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp.428–437, 2020.
- [27] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol.24, no.6, pp.381–395, 1981.
- [28] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proceedings of International Conference on Learning Representations*, San Diego, CA, USA, pp.1–14, 2015.
- [29] K. He, X. Zhang, S. Ren, *et al.*, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp.770–778, 2016.
- [30] Y. Zhang, D. Li, B. Jin, *et al.*, “Monocular 3D reconstruction of human body,” in *Proceedings of 2019 Chinese Control Conference (CCC)*, Guangzhou, China, pp.7889–7894, 2019.
- [31] S. Jia, Z. Gan, Y. Xi, *et al.*, “A deep reinforcement learning bidding algorithm on electricity market,” *Journal of Thermal Science*, vol.29, no.5, pp.1125–1134, 2020.
- [32] Y. Guan, D. Li, S. Xue, *et al.*, “Feature-fusion-kernel-based gaussian process model for probabilistic long-term load forecasting,” *Neurocomputing*, vol.426, pp.174–184, 2021.
- [33] L. J. Ba and R. Caruana, “Do deep nets really need to be deep?” in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, Cambridge, MA, USA, pp.2654–2662, 2014.
- [34] A. Romero, N. Ballas, S. E. Kahou, *et al.*, “Fitnets: Hints for thin deep nets,” in *Proceedings of International Conference on Learning Representations*, San Diego, CA, USA, pp.1–12, 2015.
- [35] S. Zagoruyko and N. Komodakis, “Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer,” in *Proceedings of International Conference on Learning Representations*, Toulon, France, pp.1–13, 2017.
- [36] J. Yim, D. Joo, J. Bae, *et al.*, “A gift from knowledge distillation: Fast optimization, network minimization and transfer learning,” in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp.7130–7138, 2017.
- [37] Y. Liu, C. Shu, J. Wang, *et al.*, “Structured knowledge distillation for dense prediction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, early access, DOI: [10.1109/TPAMI.2020.3001940](https://doi.org/10.1109/TPAMI.2020.3001940), 2020.
- [38] S. Hinterstoisser, V. Lepetit, S. Ilic, *et al.*, “Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes,” in *Proceedings of 11th Asian Conference on Computer Vision*, Daejeon, Korea, pp.548–562, 2012.
- [39] J. Xiao, J. Hays, K. A. Ehinger, *et al.*, “Sun database: Large-scale scene recognition from abbey to zoo,” in *Proceedings of 2010 IEEE Computer Society Conference on*

Computer Vision and Pattern Recognition, San Francisco, CA, USA, pp.3485–3492, 2010.

- [40] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proceedings of International Conference on Learning Representations*, San Diego, CA, USA, pp 1–15, 2015.
- [41] E. Brachmann, F. Michel, A. Krull, *et al.*, “Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp.3364–3372, 2016.
- [42] Z. Li, G. Wang, and X. Ji, “CDPN: Coordinates-based disentangled pose network for real-time RGB-based 6-DoF object pose estimation,” in *Proceedings of the International Conference on Computer Vision*, Seoul, Korea (South), pp.7677–7686, 2019.
- [43] G. Wang, F. Manhardt, F. Tombari, *et al.*, “GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Los Alamitos, CA, USA, pp.16611–16621, 2021.



GUAN Qi received the B.S. degree in measurement and control from Southeast University, Nanjing, China, in 2019. She is currently pursuing the M.S. degree in control engineering with Shanghai Jiao Tong University, Shanghai, China. Her research interests are 6D pose estimation and real-time application in deep learning.

(Email: qiguan@sjtu.edu.cn)



(Email: zihaosheng@sjtu.edu.cn)

SHENG Zihao received the B.S. degree in automation from Xi’an Jiaotong University, Xi’an, China, in 2019. He is currently pursuing the M.S. degree in control engineering with Shanghai Jiao Tong University, Shanghai, China. His research interests include intelligent transportation systems, autonomous driving, and intelligent control.



XUE Shibe (corresponding author) received the Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China, in 2013. From 2014 to 2016, he was a PostDoctoral Researcher with the University of New South Wales, Canberra, ACT, Australia, and then, he worked as a PostDoctoral Researcher with the Department of Physics, Taiwan Cheng Kung University, Tainan, China. In July 2017, he joined Shanghai Jiao Tong University, Shanghai, China, where he is currently an Associate Professor with the Department of Automation. He was selected for the Shanghai Pujiang Program funded by the Shanghai Science and Technology Committee in 2018. His research interests include quantum control, optimization and intelligent control of complex systems. (Email: shbxue@sjtu.edu.cn)