Vote from the Center: 6 DoF Pose Estimation in RGB-D Images by Radial Keypoint Voting (Supplementary Material)

Yangzheng Wu, Mohsen Zand, Ali Etemad, and Michael Greenspan

Dept. of Electrical and Computer Engineering, Ingenuity Labs Research Institute
Queen's University, Kingston, Ontario, Canada

S.1 Overview

We document here some addition implementation details, results and further hyperparameter experiments. In Sec. S.2, the complete details of the Fully Convolutional ResNet backbone are provided, in sufficient detail to recreate the network. In Sec. S.3, the 6 DoF pose estimation accuracy results for each individual object in the three data sets are presented, as well as some extra bounding box image samples. Finally in Sec. S.4, six additional experiments are included investigating the impact of the number of keypoints, the number of skip connections in the backbone network, the use of a combined vs. separate networks for each keypoint, the depth of the backbone network, the accumulator space resolution, as well as the impact of combining the three different voting schemes into various multi-scheme voting configurations.

S.2 ResNet Backbone Structure

As shown in Table S.10, we modified ResNet into a Fully Convolutional Network. To start with, we replaced the Fully Connected Layer with a convolutional layer for the following up sampling layers. We then applied up-sampling to the feature map with a combination of convolution, bilinear interpolations, and skip concatenations from the residual blocks. We apply more skip layers than did PVNet [S.17], under the assumption that the convolutional feature maps would preserve more local features than the alternative bilinear interpolation, especially for deeper small scale feature maps. This design choice was supported by the experiment described in Sec. S.4.2.

We conducted an experiment on three objects, ape, driller and eggbox in Occlusion LINEMOD with different fully convolutional ResNets structures. Each network is trained until fully convergence with consistent hyper parameter sets. The resutls are shown in Table S.1. Deeper ResNet has a tiny performance improvement on three objects tested with a minor sacrifice of speed. We ended up with $ResNet152\_32s$ when conducting the full test on all three datasets.

Table S.1: ADD(S) metrics for 3 LMO objects on different ResNet backbones with ICP.

LMO	ape	driller	eggbox
ResNet18_32s	60.2	77.9	81.9
ResNet34_32s	60.2	77.9	81.9
ResNet50_32s	60.8	78.4	81.9
ResNet101_32s	61.3	78.4	81.9
ResNet152_32s	61.3	78.8	82.3

S.3 Accuracy Results Per Object and BOP Benchmark

The detailed LINEMOD and Occlusion LINEMOD ADD(s) results, and the YCB-Video ADD(s) and AUC results categorized per object are listed in Table S.8, Table S.7 and Table S.9, respectively. Some additional successful images showing recovered and ground truth bounding boxes are displayed in Figure S.3.

As can be seen in Table S.8, the original LINEMOD dataset is mostly saturated, with results from a number of different methods that are close to perfect. Nevertheless, RCVPose+ICP outperformed all alternatives at 99.7%, with 100% ADD(s) for three objects, including the only perfect scores for the driller and holepuncher objects.

The results in Table S.7 show Occlusion LINEMOD to be quite challenging. This is not only because of the occluded scenes, but is also due to the fact that the meshes are not very precisely modelled, and that some ground truth poses are not accurate for some cases.

The YCB-Video dataset has two evaluation metrics, as shown in Table S.9. In general, AUC is more foregiving than ADD(s) since AUC has a tolerance of up to 10 cm [S.26]. For some objects like the master chef can and the power drill, RCVPose performs slightly worse in AUC compared to PVN3D [S.4], while still performing better in ADD(s).

All three datasets were also evaluated by the standardized metrics proposed by BOP [S.7]. The results in Table. S.2 show that our average recall outperformed CosyPose [S.10] on LINEMOD and occlusion LINEMOD. Although we did not perform better on YCB-Video, we did perform better for the average results over all three datasets. Our method also runs at 18 fps which is also more time efficient compared to 0.36 fps for CosyPose.

S.4 Extra Hyperparameter Experiments

S.4.1 Number of Keypoints

Previous works have used between 4 [S.16], and up to 8 [S.4, S.17] or more [S.18] keypoints per object, selected from bounding box corners [S.20, S.21, S.13] or

Table S.2: LINEMOD, Occlusion LINEMOD and YCB-Video evaluated based on BOP Average Recall metrics [S.7]

	AR_{VSD}			AR_{MSSD}			AR_{MSPD}			Average
Method	LM	LMO	YCB-V	LM	LMO	YCB-V	LM	LMO	YCB-V	AR
PVNet [S.17]	-	0.43	-	-	0.54	-	-	0.75	-	-
EPOS [S.6]	-	0.39	0.63	-	0.50	0.68	-	0.75	0.78	-
SO-Pose [S.1]	-	0.44	0.65	-	0.58	0.73	-	0.82	0.76	-
CosyPose [S.10]	0.67	0.58	0.83	0.81	0.75	**0.90**	**0.84**	**0.83**	0.85	0.78
RCVPose+ICP	**0.74**	**0.68**	**0.86**	**0.83**	**0.77**	0.86	0.83	0.79	**0.86**	**0.80**

using the FPS algorithm [S.18, S.4, S.17]. It has been suggested that a greater number of keypoints is preferable to improve robustness and accuracy [S.4, S.17], especially for pure RGB methods in which at least 3 keypoints need to be visible for any view of an object to satisfy the constraints of the P3P algorithm [S.19, S.2].

We examined the impact of the number of keypoints on pose estimation accuracy. Sets of 3, 4 and 8 keypoints were selected for the ape, driller and eggbox LINEMOD objects, using the Bounding Box selection method described in Sec. 4.5 in the main paper. The results indicate that increasing the number of RCVPose keypoints does not impact pose estimation accuracy, which changed at most only 0.4% between these settings for all three objects. This is likely due to the high accuracy of keypoint location estimation under radial voting, which removes the added benefit of redundant keypoints. Given that the time and memory expense scale linearly with the number of keypoints, we settled upon the use of the minimal 3 keypoints for RCVPose for all of our experiments.

S.4.2 Number of Skip Connections

There were five different network architectures proposed in the intial ResNet paper [S.3]. While some 6 DoF pose recovery works use variations of ResNet-18 [S.17, S.23, S.27, S.22] others use ResNet-50 [S.24, S.15]. Some customize the structure by converting it to an encoder [S.22, S.15, S.27, S.23], adding extra layers and skip connections [S.17] while others use the original ResNet unaltered [S.4, S.14].

We conducted an experiment which examined the impact of the number of skip connections on mean keypoint estimation error $\bar{\epsilon}$. We increased the number of skip connections for ResNet-18, from 3 to 5. Such skip connections serve to improve the influence of image features during upsampling. The results are displayed in Table S.3, and show that increasing the skip connections from 3 to 5, decreased both the mean and the standard deviation of the keypoint estimation error by a large margin, in all cases. We included 5 skip connections in our architecture, for all experiments, as shown in Fig S.1.

Table S.3: Average keypoint estimation error mean (μ [mm]) and standard deviation (σ [mm]) for different ResNet-18 backbone skip connections. Increasing the skip connections reduced the error of the estimation

	# of skip connections			
	3		5	
	μ	σ	μ	σ
ape	2.4	1.1	1.8	0.8
driller	3.6	1.2	2.7	0.8
eggbox	3.5	1.7	2.4	1.2

Table S.4: Keypoint localization error, for training all three keypoints' radii simultaneously in one network and separately in three networks: $\bar{\epsilon}$ mean ($\mu_{\{sim|sep\}}$) and standard deviation ($\sigma_{\{sim|sep\}}$) for radial voting schemes

	$\bar{\epsilon}$ [mm]			
	simultaneously		separately	
	μ_{sim}	σ_{sim}	μ_{sep}	σ_{sep}
ape	1.7	0.9	**1.3**	**0.7**
driller	2.6	1.4	**2.2**	**1.0**
eggbox	2.5	1.3	**2.0**	**0.7**

S.4.3 Number of Networks

Some of the 6 DoF pose estimators trained a single distinct network for each individual object [S.4, S.17, S.23] whereas other multi-class methods trained a single network for all classes combined [S.9, S.25, S.1]. We conducted an experiment on the optimal configuration of the number of networks. As shown in Table. S.4, The radii regression is more accurate when a single network is trained separately on each keypoint compared to training simultaneously on all three keypoints per object. Therefore, we trained separate networks for each keypoint among each objects to achieve the best performance, with a small sacrifice of the time performance.

S.4.4 ResNet Backbone Depth

A further experiment tested different ResNet depths, from 18 to 152 layers. The results are plotted in Fig. S.2, and indicate that the substantially deeper networks exhibit only a minor reduction in average keypoint estimation error $\bar{\epsilon}$.

Despite the rather minor improvement due to increased depth, we nevertheless used ResNet-152 with 5 skip connections in the RCVPose in our experiments, as shown in Fig. S.1 compared to PVNet. It is likely that we would have received very similar results had we based our backbone network on ResNet-18, albeit with a faster training cycle and smaller memory footprint.

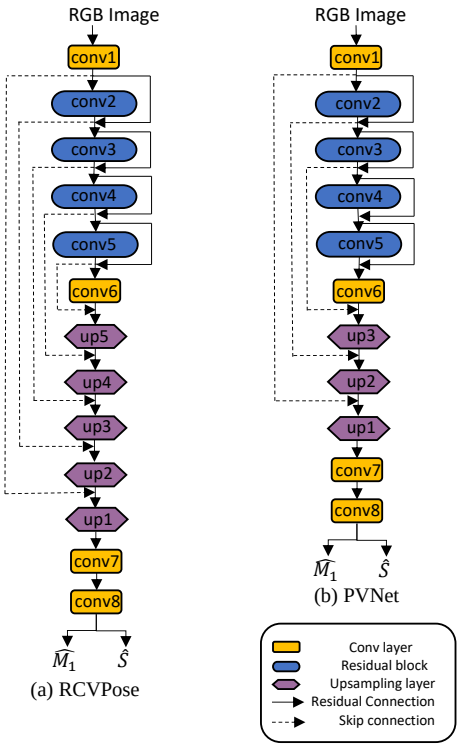

Fig. S.1: Backbone network structure for (a) RCVPose and (b) PVNet: Denser skip connections allow more local image features to be kept during upsampling

S.4.5 Accumulator Space Resolution

We varied the accumulator space resolution to evaluate the balance of accuracy and efficiency. Resolution ρ refers to the linear dimension of a voxel edge (i.e. voxel volume = ρ^3). We selected 6 different resolutions from $\rho = 1$ mm to 16 mm, and ran the voting module for each ρ value with the same system, for all 3 scaled bounding box keypoints of all test images of the LINEMOD ape object.

The results are listed in Table S.5 which shows the means μ_r and standard deviations σ_r of the keypoint estimation errors $\bar{\epsilon}$ and ADD metric, and both the time and space efficiencies, for varying voxel resolutions. As expected, the voting module was faster and smaller, and the keypoint estimation error was greater, at coarser resolutions. The ADD value, which is the main metric used to identify a successful pose estimation event, remains nearly constant up to a resolution of 5 mm. The $\rho = 5$ mm voxel size therefore achieved both an acceptable speed of 24 fps, an efficient memory footprint of 3.4 Mbtyes, and close to the highest ADD value, and so it was subsequently used throughout the experiments.

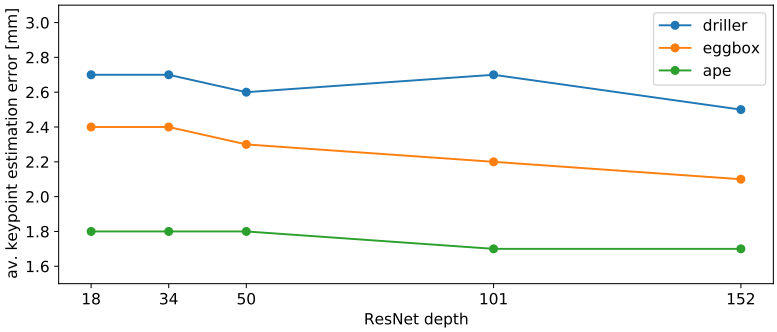

Fig. S.2: Mean keypoint estimation error [mm] vs. ResNet depth

Table S.5: Accumulator space resolution ρ [mm] impact on accuracy $\bar{\epsilon}$ [mm], ADD [%], processing speed [fps], and memory [Mbyte], for LINEMOD ape test images. The processing speed includes only the accumulator space time performance

ρ	$\bar{\epsilon}$ [mm]		ADD	speed	memory
[mm]	μ_r	σ_r	[%]	[fps]	[Mbyte]
0.5	1.65	0.63	61.5	1.6	$4\times957^3 = 3517$
1	1.75	0.81	61.5	5	$4\times479^3 = 440.61$
2	2.33	0.52	61.3	12	$4\times239^3 = 54.81$
4	6.27	0.72	61.3	20	$4\times118^3 = 6.57$
5	6.33	0.69	61.3	24	$4\times95^3 = 3.43$
8	11.73	2.37	55.2	32	$4\times58^3 = 0.78$
16	17.92	5.52	45.7	40	$4\times28^3 = 0.09$

S.4.6 Ensemble Multi-scheme Voting

The accumulator space is represented exactly the same for all three voting schemes, and is handled in exactly the same manner to extract keypoint locations through peak detection, once the voting has been completed. It is therefore possible and straightforward to combine voting schemes, by simply adding their resulting accumulator spaces prior to peak detection.

We implemented this and compared the impact of all possible combinations of offset, vector, and radial voting schemes. The results are shown in Table S.6, which also includes the results from each individual voting scheme for comparison. It can be seen that the radial voting scheme outperforms all other alternatives, yielding a lower mean and standard deviation of keypoint estimation error $\bar{\epsilon}$. The next best alternative was the combination of all three schemes, which was greater than 3.5X less accurate than pure radial voting. Combing radial and offset voting slightly improved results over pure offset voting in two of the three objects. Curiously, combining radial and vector voting degraded

Table S.6: Combined Accumulator Space: $\bar{\epsilon}$ mean ($\mu_{\{v|o|r\}}$) and standard deviation ($\sigma_{\{v|o|r\}}$) for different combination of 3 voting schemes, with $\bar{r}$ = mean distance of keypoints to object centroid

		$\bar{\epsilon}$ [mm]															
		vector + offset		vector + radial		radial + offset		vector + offset + radial		vector		offset		radial	
	$\bar{r}$ [mm]	μ_v	σ_v	μ_o	σ_o	μ_r	σ_r	μ_r	σ_r	μ_r	σ_r	μ_r	σ_r	μ_r	σ_r
ape	142.1	20.2	12.4	12.7	6.7	9.8	6.2	7.2	1.2	12.5	7.6	10.4	5.3	1.8	0.8
driller	318.8	22.3	11.7	13.3	7.9	8.7	3.4	5.7	2.3	11.3	8.2	9.5	3.5	2.7	0.8
eggbox	197.3	21.6	13.5	17.4	10.5	12.1	5.2	6.4	3.3	13.7	8.5	11.4	4.7	2.4	1.2

Table S.7: Occlusion LINEMOD Accuracy Results. Non-symmetric objects are evaluated with ADD, and symmetric objects (annotated with *) are evaluated with ADD-s

		Object								
Mode	Method	ape	can	cat	driller	duck	eggbox*	glue*	holepuncher	ADD(s)[%]
RGB	Oberweger [S.12]	12.1	39.9	8.2	45.2	17.2	22.1	35.8	36.0	27.1
	Hu et al. [S.8]	17.6	53.9	3.3	62.4	19.2	25.9	39.6	21.3	30.4
	Pix2Pose [S.14]	22.0	44.7	22.7	44.7	15.0	25.2	32.4	49.5	32.0
	DPOD [S.27]	-	-	-	-	-	-	-	-	32.8
	PVNet [S.17]	15.8	63.3	16.7	25.2	**65.7**	50.2	49.6	39.7	40.8
	PPRN [S.22]	-	-	-	-	-	-	-	-	58.4
RGB +D ref	YOLO6D [S.21]	-	-	-	-	-	-	-	-	6.4
	SSD6D+ref [S.9]	-	-	-	-	-	-	-	-	27.5
	PoseCNN [S.26]	9.6	45.2	0.9	41.4	19.6	22.0	38.5	22.1	24.9
	DPOD+ref [S.27]	-	-	-	-	-	-	-	-	47.3
RGB-D	PVN3D [S.4]	33.9	88.6	39.1	78.4	41.9	80.9	68.1	74.7	63.2
	RCVPose	60.3	92.5	50.2	78.2	52.1	81.2	72.1	75.2	70.2
	RCVPose+ICP	**61.3**	**93**	**51.2**	**78.8**	53.4	**82.3**	**72.9**	**75.8**	**71.1**

results for all objects compared to pure vector voting, as did combining vector and offset voting. Based on these results, it seems possible that there may be better ways than simply adding the individual accumulator spaces to ensemble the information from these three voting schemes to reduce error further.

Table S.8: LINEMOD Accuracy Results: Non-symmetric objects are evaluated with ADD, and symmetric objects (annotated with *) are evaluated with ADD-s

		Object													
Mode	Method	ape	bench-vise	camera	can	cat	driller	duck	eggbox*	glue*	hole-puncher	iron	lamp	phone	mean
RGB	SSD6D+ref [S.9]	2.6	15.1	6.1	27.3	9.3	12.0	1.3	2.8	3.4	3.1	14.6	11.4	9.7	9.1
	Pix2Pose [S.14]	58.1	91	60.9	84.4	65	76.3	43.8	96.8	79.4	74.8	83.4	82	45	72.4
	DPOD [S.27]	53.3	95.3	90.4	94.1	60.4	97.7	66	99.7	93.8	65.8	99.8	88.1	74.2	83.0
	PVNet [S.17]	43.62	99.9	86.9	95.5	79.3	96.4	52.6	99.2	95.7	81.9	98.9	99.3	92.4	86.3
	PPRN [S.22]	84.5	98.7	93.7	97.8	87.3	96.9	88.5	98.5	99.5	84.5	99.1	98.7	92.5	93.9
	DeepIM [S.11]	77	97.5	93.5	96.5	82.1	95.0	77.7	97.1	99.4	52.8	98.3	97.5	87.7	88.6
RGB +D ref	YOLO6D [S.21]	21.6	81.8	36.6	68.8	41.8	63.5	27.2	69.6	80	42.6	75	71.1	47.7	56.0
	SSD6D+ref [S.9]	-	-	-	-	-	-	-	-	-	-	-	-	-	34.1
	DPOD+ref [S.27]	87.7	98.5	96.1	99.7	94.7	98.8	86.3	99.9	96.8	86.8	100	96.8	94.7	95.2
RGB-D	DenseFusion [S.23]	92.3	93.2	94.4	93.1	96.5	87.0	92.3	99.8	100.0	92.1	97.0	95.3	92.8	94.3
	PVN3D [S.4]	97.3	99.7	99.6	**99.5**	**99.8**	99.3	98.2	99.8	100.0	99.9	99.7	**99.8**	99.5	99.4
	RCVPose	99.2	99.6	99.7	99	99.4	99.7	99.4	98.7	99.7	99.8	99.9	99.2	99.1	99.43
	RCVPose+ICP	**99.6**	**99.7**	**99.7**	99.3	99.7	**100**	**99.7**	99.3	**100.0**	**100**	**99.9**	99.5	**99.7**	**99.7**

Table S.9: YCB Video AUC [S.26] and ADD(s) [S.5] results:Non-symmetric objects are evaluated with ADD, and symmetric objects (annotated with *) are evaluated with ADD-s. The AUC metrics is based on the curve with ADD for non-symmetries and ADDs with symmetries

Refine	Metric	Method	002 master chef can	003 cracker box	004 sugar box	005 tomato soup can	006 mustard bottle	007 tuna fish can	008 pudding box	009 gelatin box	010 potted meat can	011 banana	019 pitcher base	021 bleach cleanser	024 bowl*	025 mug	035 power drill	036 wood block*	037 scissors	040 large marker	051 large clamp*	052 extra large clamp*	061 foam brick*	mean
No	AUC	PoseCNN [S.26]	83.9	76.9	84.2	81.0	90.4	88.0	79.1	87.2	78.5	86.0	77.0	71.6	69.6	78.2	72.7	64.3	56.9	71.7	50.2	44.1	88.0	75.8
		DF(per-pixel)[S.23]	95.3	92.5	95.1	93.8	95.8	95.7	94.3	97.2	89.3	90.0	93.6	94.4	86.0	95.3	92.1	89.5	90.1	95.1	71.5	70.2	92.2	91.2
		PVN3D[S.4]	**96.0**	96.1	97.4	96.2	97.5	96	97.1	97.7	93.3	96.6	**97.4**	96.0	90.2	97.6	**96.7**	**90.4**	**96.7**	**96.7**	93.6	88.4	**96.8**	*95.5*
		RCVPose	95.7	**97.2**	**97.6**	**98.2**	**97.9**	**98.2**	**97.7**	97.7	**97.9**	**97.9**	96.2	**99.2**	**95.2**	**98.4**	96.2	89.1	96.2	95.9	**95.2**	**94.7**	95.7	**96.6**
	ADD (s)	PoseCNN [S.26]	50.2	53.1	68.4	66.2	81.0	70.7	62.7	75.2	*59.5*	72.3	53.3	50.3	69.6	58.5	55.3	64.3	35.8	58.3	50.2	44.1	88.0	59.9
		DF(per-pixel)[S.23]	70.7	86.9	90.8	84.7	90.9	79.6	89.3	95.8	79.6	76.7	87.1	87.5	86.0	83.8	83.7	89.5	77.4	89.1	71.5	70.2	92.2	82.9
		PVN3D[S.4]	80.5	94.8	96.3	88.5	96.2	89.3	95.7	96.1	88.6	93.7	**96.5**	93.2	90.2	95.4	95.1	**90.4**	92.7	91.8	93.6	88.4	**96.8**	91.8
		RCVPose	**93.6**	**95.7**	**97.2**	**94.7**	**97.2**	**96.4**	**97.1**	**96.5**	**90.2**	**96.7**	95.7	**97.8**	**94.9**	**96.3**	**95.4**	89.3	**94.7**	**92.4**	**96.4**	**94.7**	95.7	**95.2**
Yes	AUC	PoseCNN [S.26] +ICP	95.8	92.7	98.2	94.5	98.6	97.1	97.9	98.8	92.7	97.1	97.8	96.9	81.0	94.9	98.2	87.6	91.7	97.2	75.2	64.4	97.2	93.0
		DF(iterative)[S.23]	96.4	95.8	97.6	94.5	97.3	97.1	96.0	98.0	90.7	96.2	97.5	95.9	89.5	96.7	96.0	92.8	92.0	97.6	72.5	69.9	92.0	93.2
		PVN3D[S.4]+ICP	95.2	94.4	97.9	95.9	**98.3**	96.7	**98.2**	**98.8**	93.8	98.2	**97.6**	97.2	92.8	97.7	**97. 1**	**91.1**	95.0	98.1	95.6	90.5	**98.2**	96.1
		RCVPose+ICP	**96.2**	**97.9**	97.9	**99**	98.2	**98.6**	98.1	98.4	**98.4**	**98.3**	97.2	**99.6**	**96.9**	**98.7**	96.4	90.7	**96.4**	**96.6**	**96.2**	**95.1**	96.6	**97.2**
	ADD (s)	PoseCNN [S.26] +ICP	68.1	83.4	97.1	81.8	98.0	83.9	96.6	98.1	83.5	91.9	96.9	92.5	81.0	81.1	97.7	87.6	78.4	85.3	75.2	64.4	97.2	85.4
		DF(iterative)[S.23]	73.2	94.1	96.5	85.5	94.7	81.9	93.3	96.7	83.6	83.3	96.9	89.9	89.5	88.9	92.7	92.8	77.9	93.0	72.5	69.9	92.0	86.1
		PVN3D[S.4]+ICP	79.3	91.5	96.9	89.0	**97.9**	90.7	97.1	**98.3**	87.9	96.0	**96.9**	95.9	92.8	96.0	95.7	91.1	87.2	91.6	95.6	90.5	**98.2**	92.3
		RCVPose+ICP	**94.7**	**96.4**	**97.6**	**95.4**	97.7	**96.7**	**97.4**	97.9	**92.6**	**97.2**	96.7	**98.4**	**95.3**	**97.1**	**96.2**	**91.2**	**94.9**	**93.2**	**96.7**	**94.9**	96.6	**95.9**

Fig. S.3: Occluded LINEMOD sample results: Blue box = ground truth, green box = estimate

Table S.10: ResNet Backbone structure compared to PVNet

	ResNet Backbone Strcture					
Layer	ResNet-152 32s(RCVPose)	ResNet-101 32s	ResNet-50 32s	ResNet-34 32s	ResNet-18 32s	ResNet-18 8s(PVNet)
conv1	7×7, 64, stride 2					
conv2	3×3 max pool, stride 2					
conv3	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 8$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$
conv4	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 36$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$
conv5	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$
conv6	$\begin{bmatrix} 3\times3, \textit{stride } 1, \textit{padding } 1 \\ \textit{batch norm} \\ \textit{ReLU} \end{bmatrix}$					
up5	$\begin{bmatrix} \textit{conv } 3\times3, \textit{stride } 1, \textit{padding } 1 \\ \textit{batch norm} \\ \textit{ReLU} \\ \textit{bilinear interpolation} \end{bmatrix}$					
up4	$\begin{bmatrix} \textit{conv } 3\times3, \textit{stride } 1, \textit{padding } 1 \\ \textit{batch norm} \\ \textit{ReLU} \\ \textit{bilinear interpolation} \end{bmatrix}$					
up3	$\begin{bmatrix} \textit{conv } 3\times3, \textit{stride } 1, \textit{padding } 1 \\ \textit{batch norm} \\ \textit{ReLU} \\ \textit{bilinear interpolation} \end{bmatrix}$					$\begin{bmatrix} \textit{conv } 3\times3, \textit{stride } 1, \textit{padding } 1 \\ \textit{batch norm} \\ \textit{Leaky ReLU} \\ \textit{bilinear interpolation} \end{bmatrix}$
up2	$\begin{bmatrix} \textit{conv } 3\times3, \textit{stride } 1, \textit{padding } 1 \\ \textit{batch norm} \\ \textit{ReLU} \\ \textit{bilinear interpolation} \end{bmatrix}$					$\begin{bmatrix} \textit{conv } 3\times3, \textit{stride } 1, \textit{padding } 1 \\ \textit{batch norm} \\ \textit{Leaky ReLU} \\ \textit{bilinear interpolation} \end{bmatrix}$
up1	$\begin{bmatrix} \textit{conv } 3\times3, \textit{stride } 1, \textit{padding } 1 \\ \textit{batch norm} \\ \textit{ReLU} \\ \textit{bilinear interpolation} \end{bmatrix}$					$\begin{bmatrix} \textit{conv } 3\times3, \textit{stride } 1, \textit{padding } 1 \\ \textit{batch norm} \\ \textit{Leaky ReLU} \\ \textit{bilinear interpolation} \end{bmatrix}$
conv7	$\begin{bmatrix} 3\times3, \textit{stride } 1, \textit{padding } 1 \\ \textit{batch norm} \\ \textit{ReLU} \end{bmatrix}$					$\begin{bmatrix} 3\times3, \textit{stride } 1, \textit{padding } 1 \\ \textit{batch norm} \\ \textit{Leaky ReLU} \end{bmatrix}$
conv8	$1 \times 1, \textit{stride } 1, \textit{padding } 0$					$1 \times 1, \textit{stride } 1, \textit{padding } 0$

References

[S.1] Di, Y., Manhardt, F., Wang, G., Ji, X., Navab, N., Tombari, F.: So-pose: Exploiting self-occlusion for direct 6d pose estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12396–12405 (2021)

[S.2] Gao, X., Hou, X., Tang, J., Cheng, H.: Complete solution classification for the perspective-three-point problem. IEEE Trans. Pattern Anal. Mach. Intell. **25**, 930–943 (2003)

[S.3] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

[S.4] He, Y., Sun, W., Huang, H., Liu, J., Fan, H., Sun, J.: Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)

[S.5] Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In: Asian conference on computer vision. pp. 548–562. Springer (2012)

[S.6] Hodan, T., Barath, D., Matas, J.: Epos: Estimating 6d pose of objects with symmetries. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11703–11712 (2020)

[S.7] Hodaň, T., Sundermeyer, M., Drost, B., Labbé, Y., Brachmann, E., Michel, F., Rother, C., Matas, J.: Bop challenge 2020 on 6d object localization. In: European Conference on Computer Vision. pp. 577–594. Springer (2020)

[S.8] Hu, Y., Hugonot, J., Fua, P., Salzmann, M.: Segmentation-driven 6d object pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3385–3394 (2019)

[S.9] Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N.: Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In: Proceedings of the IEEE international conference on computer vision. pp. 1521–1529 (2017)

[S.10] Labbé, Y., Carpentier, J., Aubry, M., Sivic, J.: Cosypose: Consistent multi-view multi-object 6d pose estimation. In: European Conference on Computer Vision. pp. 574–591. Springer (2020)

[S.11] Li, Y., Wang, G., Ji, X., Xiang, Y., Fox, D.: Deepim: Deep iterative matching for 6d pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 683–698 (2018)

[S.12] Oberweger, M., Rad, M., Lepetit, V.: Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 119–134 (2018)

[S.13] Papandreou, G., Zhu, T., Chen, L.C., Gidaris, S., Tompson, J., Murphy, K.: Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 269–286 (2018)

[S.14] Park, K., Patten, T., Vincze, M.: Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7668–7677 (2019)

[S.15] Park, K., Patten, T., Vincze, M.: Neural object learning for 6d pose estimation using a few cluttered images. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 656–673. Springer International Publishing, Cham (2020)

[S.16] Pavlakos, G., Zhou, X., Chan, A., Derpanis, K.G., Daniilidis, K.: 6-dof object pose from semantic keypoints. In: 2017 IEEE international conference on robotics and automation (ICRA). pp. 2011–2018. IEEE (2017)

[S.17] Peng, S., Liu, Y., Huang, Q., Zhou, X., Bao, H.: Pvnet: Pixel-wise voting network for 6dof pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4561–4570 (2019)

[S.18] Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019)

[S.19] Quan, L., Lan, Z.: Linear n-point camera pose determination. IEEE Transactions on pattern analysis and machine intelligence **21**(8), 774–780 (1999)

[S.20] Rad, M., Lepetit, V.: Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3828–3836 (2017)

[S.21] Tekin, B., Sinha, S.N., Fua, P.: Real-time seamless single shot 6d object pose prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 292–301 (2018)

[S.22] Trabelsi, A., Chaabane, M., Blanchard, N., Beveridge, R.: A pose proposal and refinement network for better 6d object pose estimation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2382–2391 (2021)

[S.23] Wang, C., Xu, D., Zhu, Y., Martín-Martín, R., Lu, C., Fei-Fei, L., Savarese, S.: Densefusion: 6d object pose estimation by iterative dense fusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3343–3352 (2019)

[S.24] Wang, G., Manhardt, F., Shao, J., Ji, X., Navab, N., Tombari, F.: Self6d: Self-supervised monocular 6d object pose estimation. In: European Conference on Computer Vision. pp. 108–125. Springer (2020)

[S.25] Wang, G., Manhardt, F., Tombari, F., Ji, X.: Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16611–16621 (2021)

[S.26] Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes (2018)
[S.27] Zakharov, S., Shugurov, I., Ilic, S.: Dpod: 6d pose object detector and refiner. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1941–1950 (2019)