Supplementary Material for Confidence-based 6D Object Pose Estimation

Wei-Lun Huang, Chun-Yi Hung, and I-Chen Lin, *Member, IEEE*

I. Network Architecture

We adopted the same base architecture applied in [1] which contains an encoder and two decoders for different tasks. The encoder is Darknet-53 of YOLOv3 and the decoders are feature pyramid networks upon the result of encoder. The details of the architecture are depicted in Figure 1. For two decoders, we constructed the feature pyramid networks by upsampling the features and concatenating them with previous features through lateral connections.

II. Comparison of Loss Functions

We compared our loss terms in the regression branch (as listed in Eq. 1) with those in [1] (as listed in Eq. 2) in details. Our regression loss L_{reg} and confidence loss L_{conf} are as follows:

$$\begin{aligned} L_{reg} &= \sum_{i=1}^{C}\sum_{j=1}^{K} \delta_{ih_{gt}} w_i c_j \|g_j - e_j\|_1 \\ L_{conf} &= \sum_{i=1}^{C}\sum_{j=1}^{K} -\delta_{ih_{gt}} w_i \log(c_j), \end{aligned} \tag{1}$$

where e_j, c_j and g_j represent the estimated location, the estimated confidence and the ground-truth location of the j^{th} keypoint, respectively. w_i represents the median frequency weights. $\delta_{ih_{gt}}$ is a Kronecker delta function. It outputs 1 when $i = h_{gt}$, the ground-truth class; otherwise, it outputs zero.

For the convenience of reading, we rewrote the loss used in [1] in term of similar variable notations with tildes:

$$\begin{aligned} \tilde{L}_{reg} &= \sum_{i=1}^{C}\sum_{j=1}^{K} \delta_{ih_{gt}} w_i \|g_j - \tilde{e}_j\|_1 \\ \tilde{L}_{conf} &= \sum_{i=1}^{C}\sum_{j=1}^{K} \delta_{ih_{gt}} w_i \|\tilde{c}_j - e^{-\tau\|g_j - \tilde{e}_j\|_2}\|_1, \end{aligned} \tag{2}$$

where $\tilde{L}_{reg}$ is an L1 loss without the confidence-based reweighting and the confidence loss $\tilde{L}_{conf}$ is in a supervised form where the ground-truth confidence is decided based on the current regression loss, and τ is a modulating factor.

By contrast, our confidence loss L_{conf} is unsupervised and encourages regions to increase their confidences. Our regression loss L_{reg} encourages improving predictions from the high-confidence regions. The regression loss and the confidence loss compete with each other, and the network gradually assigns regions with less prediction error high confidence. Furthermore, with our confidence-based reweighting, regions of high confidence have higher weights in the regression loss. It means our training process focuses more on these regions, and they have more opportunity to converge and to improve the inference results.

W.-L. Huang, C.-Y. Hung, and I.-C. Lin are with the Institute of Multimedia Engineering, College of Computer Science, National Yang Ming Chiao Tung University (former National Chiao Tung University), Taiwan.

III. Training Dataset

We appreciated and applied the tool[1] provided by Peng et al. [2] to generate 20000 synthetic training data. A synthetic image was generated by pasting the objects cropped from the real images on the random background. These objects were pasted at random locations, orientations and scales in arbitrary order to increase the diversity. To prevent overfitting, we also applied online data augmentation including the random cropping, resizing, rotation, blurring, color jittering and the random erasing technique [3]. The details are listed in Table I. Examples are shown in Figure 2.

IV. Qualitative Results

Additional visual results for the Occlusion LINEMOD dataset are shown in Figure 3 and 4.

V. Additional Test on Unoccluded data

The proposed framework and the main experiments are presented for estimating 6D poses of partially occluded objects. To evaluate the performance for unoccluded cases, we conducted an additional test on LINEMOD data [4]. This test focuses on eight object classes coexisting in LINEMOD and Occlusion LINEMOD datasets so as to save data preparation and training. First, we applied our pretrained models *Ours*(*ind.-class*), as described in subsection V.D of the main manuscript, to test data of corresponding classes in LINEMOD. We found that the accuracy of classes (*can*, *driller*, *eggbox*) is higher than 95%, and we directly used the results. For the other classes (*ape*, *cat*, *duck*, *glue*, *holepuncher*), we took data generated in supplementary Section III and applied identical hyperparameters and online data augmentation except random erasing to retrain the models.

Table II compares the statistics of the additional test of the proposed method with those of related methods. The accuracy of our model surpasses that of Pix2Pose [5] but fell behind accuracy of PVNet [2] and HybridPose [6]. The results are reasonable since we did not precisely adjust the training data and data augmentation for unoccluded scenarios.

[1] https://github.com/zju3dv/pvnet-rendering

In such a circumstance, our models may not learn the benefit of using holistic object views, and the performance is relatively low in classes (*ape*, *duck*). On the other hand, this rapid test demonstrates that the propose method can reach high accuracy for the majority of unoccluded object classes with only partially occluded training data and it is of high potential for unoccluded scenarios.

REFERENCES

[1] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann, "Segmentation-driven 6d object pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3385–3394.

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

[3] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," *arXiv preprint arXiv:1708.04896*, 2017.

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

[5] K. Park, T. Patten, and M. Vincze, "Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation," in *Proceedings of the IEEE Conference on Computer Vision*, 2019, pp. 7668–7677.

[6] C. Song, J. Song, and Q. Huang, "Hybridpose: 6d object pose estimation under hybrid representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 431–440.

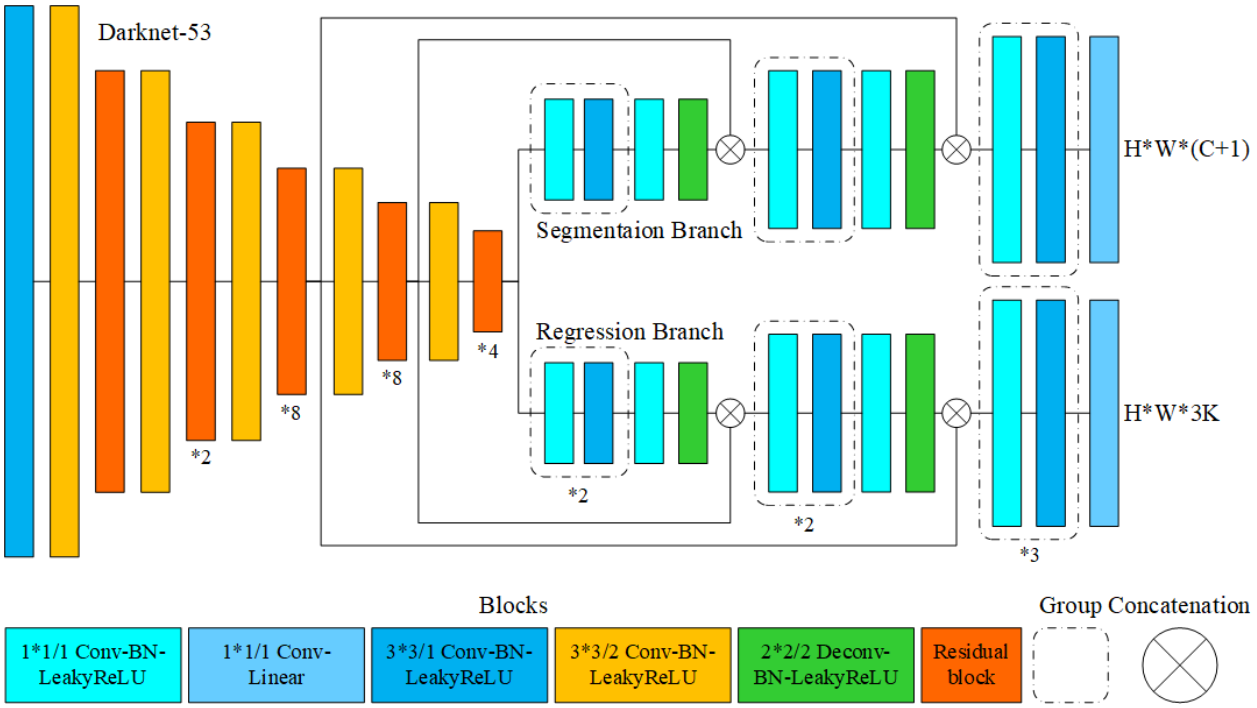
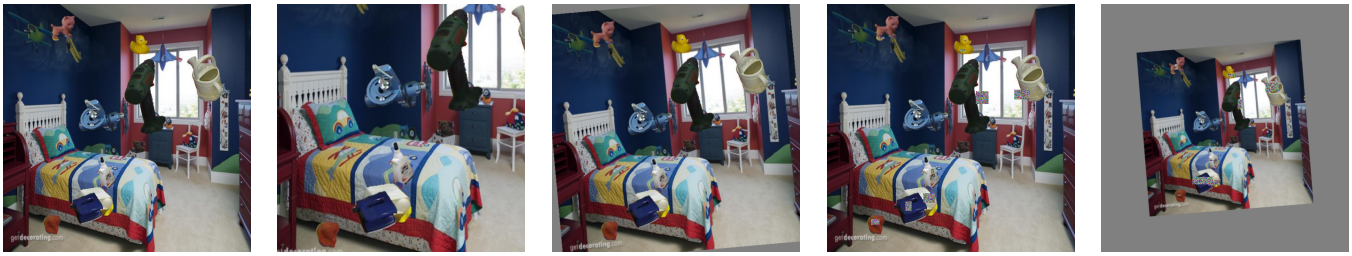


Fig. 1: Network architecture. The number below a block or a group represents the repeat times of that block or group.



(a) A synthetic image (b) Resize (c) Rotate (d) Erase (e) Result

Fig. 2: The results where different types of data augmentation are applied: (a) No data augmentation. (b) Random resizing. (c) Random rotation. (d) Random erasing. (e) All types of data augmentation.

TABLE I: Image augmentation and color augmentation (transforms.ColorJitter() in PyTorch). $U(begin, end)$ represents a uniform distribution.

Scale	Rotate	Erase area	Brightness	Contrast	Saturation	Hue
$U(0.6, 1.4)$	$U(-30°, 30°)$	$U(0.02, 0.33)$	0.1	0.1	0.05	0.05

TABLE II: Comparison with the state-of-the-art methods in terms of ADD(-S)-0.1d, tested on the LINEMOD dataset. For Ours, we trained an individual model for each of the eight objects with data in supplementary Section III that were originally prepared for Occlusion LINEMOD test. (*: symmetric objects)

	Pix2Pose [5]	PVNet [2]	HybridPose (update) [6]	Ours (trained with data for Occlusion test)
Ape	58.1	43.62	63.1	36.95
Can	84.4	95.47	98.5	96.85
Cat	65.0	79.34	89.4	78.34
Driller	76.3	96.43	98.5	98.91
Duck	43.8	52.58	65.0	38.22
Eggbox*	96.8	99.15	100.0	99.91
Glue*	79.4	95.66	98.8	86.20
Holepuncher	74.8	81.92	89.7	87.54
Average	72.33	80.52	87.88	77.87

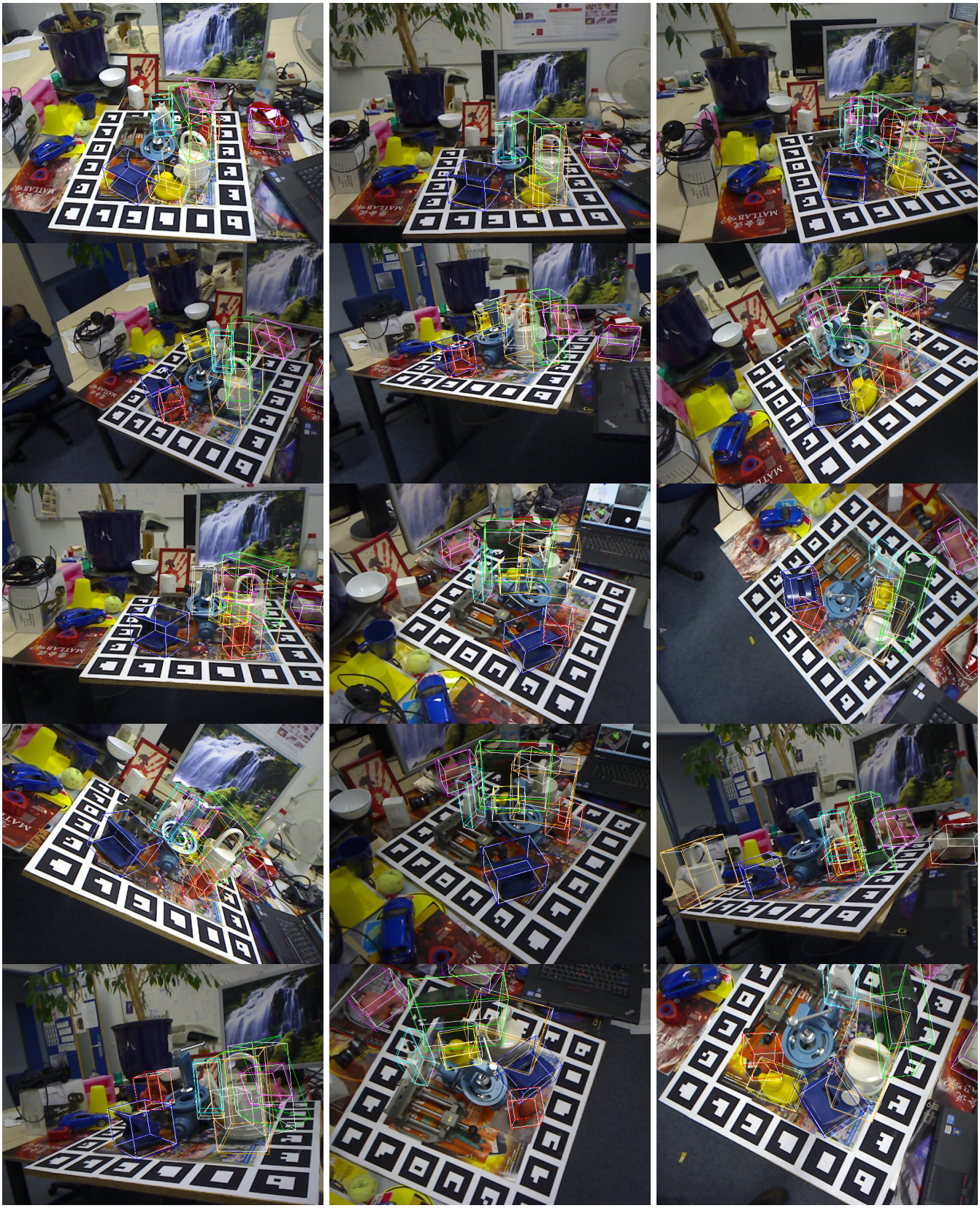


Fig. 3: Visualizations of results for the Occlusion LINEMOD dataset. White 3D bounding boxes represent the ground truth poses while 3D bounding boxes in other color represent our predicted poses of different classes, respectively.

Fig. 4: Visualizations of results for the Occlusion LINEMOD dataset. White 3D bounding boxes represent the ground truth poses while 3D bounding boxes in other color represent our predicted poses of different classes, respectively.