A Unified Evaluation of CNN-based RGB 6D Pose Estimation Models

Vasu G. Patel
Department of Civil and Environmental Engineering
Stanford University
vgpatel1@stanford.edu

Raghav Khandelwal
Department of Computer Science
Stanford University
raghav68@stanford.edu

Ken Hong
Department of Computer Science
Stanford University
kenhong@stanford.edu

Abstract

Recently, many CNN-based models have been proposed to solve the challenge of 6D pose estimation from a single RGB image under severe occlusion or truncation. Some of these approaches used to address the challenge include SingleShot (SSD-6D), Pixel-wise Voting Network (PVNET) and EfficientPose (EPOS) [18, 10, 4]. While these models all claim to have demonstrated state-of-the-art performance at the time of their release, the metrics and methodologies used to evaluate them are not completely uniform and thus, it is difficult to obtain a clear picture on their relative performance. This paper aims to perform a uniform, thorough, and systematic evaluation of these models on standardized datasets with consistent metrics so we can provide an accurate understanding of the strengths and weaknesses of each model, while also verifying the expected performance results given in the original papers. Evaluation was performed on the three models listed above on LINEMOD and Occlusion LINEMOD, two datasets widely used for the task at hand. All models were evaluated on three metrics: ADD, 5 cm/5 degree, and 2D projection error. The original papers for EPOS only gave results for the ADD metric, and the original papers for PVNet did not include the 5 cm/5 degree metric. The original paper for SSD-6D reported results for all three metrics. We hope that our results and analysis will lead to a stronger understanding of the strengths and weaknesses of these models.

1. Introduction

Object pose estimation aims to detect objects and estimate their orientations and translations relative to a canonical frame. Real-time object detection and 6D pose estimation is crucial for augmented reality, virtual reality, and robotics. Currently, methods relying on depth data acquired by RGB-D cameras are quite robust. Detecting objects of interest in images is an important task in computer vision and a lot of work in this research field have developed highly accurate methods to tackle such a problem. More recently, some works not only focus on the accuracy on the task's evaluation metrics, but also on efficiency, in order to make their methods applicable in real world scenarios with computational and runtime limitations. This paper focuses on comparing different approaches that are used for the specific task of recovering the 6D pose of an object, i.e., rotation and translation in 3D, from a single RGB image of the object. This problem is quite challenging, as complications such as severe occlusions, variations in lighting and appearance, and cluttered background objects all contribute to increased difficulty.

Given the importance of this task, many models have been presented for solving the task with varying architectures and approaches. Several of these models, such as EfficientPose and PVNet, claim to achieve very good state-of-the-art performance on shared evaluation metrics and datasets, such as the ADD metric on LINEMOD [4, 10]. However, different papers tend to place their focus on different metrics and evaluation methods, making it difficult for us to cross-compare the relative performance of these models on a wide variety of applications. Our goal, then, is to conduct a unified, thorough, and systematic evaluation of these models on standardized datasets with consistent metrics so we can provide an accurate understanding of the strengths and weaknesses of each model, while also verifying the expected performance results given in the original papers. To this end, we will evaluate three models: namely, Single Shot Detection 6D (SSD-6D), Pixel-wise Voting Network (PVNet) and Efficient Pose (EPOS) [18, 10, 4].

In SSD-6D, a deep CNN architecture takes the image as input and directly detects the 2D projections of the 3D bounding box vertices. It is claimed to be end-to-end trainable and accurate even without any a posteriori refinement [18]. In contrast to this, PVNet creates a vector-field representation for the keypoint localization phase of the pipeline. In contrast to coordinate or heatmap-based representations,

learning such a representation biases the network to focus on local features of objects and spatial relations between object parts [10]. EPOS is an extension of the 2D object detection architecture family, EfficientDet, in an intuitive way, and also aims to predict the 6D poses of objects. Two extra subnetworks are added to predict the translation and rotation of objects, analogous to the classification and bounding box regression subnetworks of other models. [4].

Our work in this paper specifically consists of the following: we used open-source implementations of the above models along with pre-trained weights to evaluate their performance on two open-source benchmark datasets for 6D Pose Estimation, specifically LINEMOD and LINEMOD Occlusion. We made modifications to the inference code in order to generate data that allows us to obtain the three metrics: ADD, 5 cm/5 degree, and 2D projection error, and we present these results both quantitatively and visually along with an analysis of the implications involved.

2. Related Work

Our literature review reveals some recent work on 6D object pose tracking using RGB-D images using deep learning techniques. se(3)-TrackNet [20] provides relative pose from RGB-D images using an end-to-end network. Tracking is done by propagating over time given an initial condition using the predicted relative poses. PoseRBPF [5] outputs a full object pose distribution by combining particle filtering with a deep auto-encoder. However, these models only work satisfactorily when the object velocity is maintained at some constant velocity, which might not be adequate for the case of fast object motion. This limitation is overcome by ROFT: Real-time Optical Flow-aided [11] method, which uses Kalman filtering for 6D object pose and velocity tracking from a RGB-D image stream.

2.1. Classical Methods

Most commonly used methods for predicting 6D pose from RGB images include local key-points and feature matching [9] [16]. Such traditional methods are invariant to changes in scale, rotation, illumination and viewpoints. These methods being fast and robust to occlusion and scene clutter; however, they are only reliable to handle textured objects in high resolution images.

2.2. RGB-D Methods

The advent of commodity depth cameras has spawned many RGB-D object pose estimation methods. [15] extended previous work using discriminative learning and cascaded detection for higher accuracy and efficiency respectively. Their paper proposed using regression forests to predict dense object coordinates, to segment the object and recover its pose from dense correspondences. They also extended their method to handle uncertainty during inference and deal with solely-RGB images [3].

2.3. CNN Methods

In recent years, research in most pose estimation tasks has been dominated by CNNs. Techniques such as viewpoints and keypoints [19] and render for CNN [17] cast object categorization and 3D pose estimation into classification tasks, specifically by discretizing the pose space. In parallel to these developments, on the 2D object detection task, there has been a progressive trend towards single shot CNN frameworks as an alternative to two-staged methods such as Faster-RCNN [14] that first find a few candidate locations in the image and then classifies them as objects or background. Recent advances in single shot architecture like YOLO [12] [13] and SSD [8] have been shown to be fast and accurate by some metrics.

3. Technical Approach

We adapted and modified openly-available open-source versions of the following three models, and evaluated them on the datasets described below. Performance was evaluated with the metrics given below.

3.1. Datasets

We used two datasets explicitly designed to test 6D pose estimation models, namely, LINEMOD and Occlusion LINEMOD [6, 1]. In both datasets, several objects were omitted from our experiments due to incomplete data or unavailable training weights.

3.1.1 LINEMOD

LINEMOD is the most commonly used dataset used for 6D pose estimation for texture-less objects in cluttered scenes as shown in Figure 1. There are 15,783 images in LINEMOD for 15 objects. Each object features in about 1,200 instances [6]. The objects {eggbox, glue, phone} were omitted in our experiment.

Figure 1. LINEMOD dataset sample image

3.1.2 Occlusion LINEMOD

Occlusion LINEMOD is a multi-object detection and pose estimation dataset that contains additional annotations for all objects in a subset of the LINEMOD images as shown in Figure 2 [1]. The objects {benchvise, driller, eggbox, glue} were omitted in our experiment.

Figure 2. Occlusion LINEMOD dataset sample image

3.2. Models

3.2.1 Single Shot 6D Pose (SSD-6D)

In this model, a single-shot deep CNN architecture (Figure 3) takes the image as input and directly detects the 2D projections of the 3D bounding box vertices. It is end-to-end trainable and accurate even without any a posteriori refinement. Since we do not need a refinement step, we also do not need a precise and detailed textured 3D object model that is needed by other methods. Rather, we only need the 3D bounding box of the object shape for training. This can be derived from other easier to acquire and approximate 3D shape representations [18].

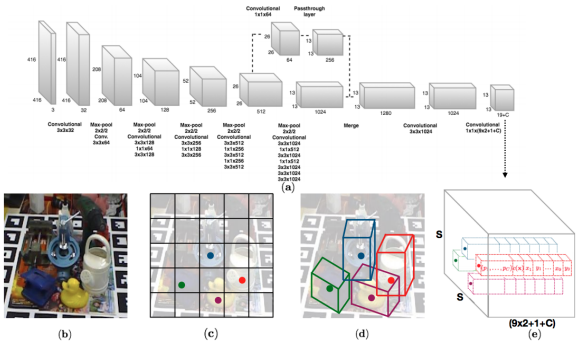

Figure 3. SSD-6D model architecture [8]

3.2.2 Pixel-wise Voting Network (PVNet)

PVNet (Figure 4) estimates the 6D pose using a two-stage pipeline: first, detecting 2D keypoints using CNN; second, computing the 6D pose using the PnP algorithm. This method uses a pixel-wise voting network to detect the keypoints of the image in a RANSAC-like fashion, which is capable of handling occluded and truncated objects in an image. RANSAC-based voting also gives the probability distribution of each keypoint, allowing us to estimate the 6D pose with an uncertainty-driven PnP [10].

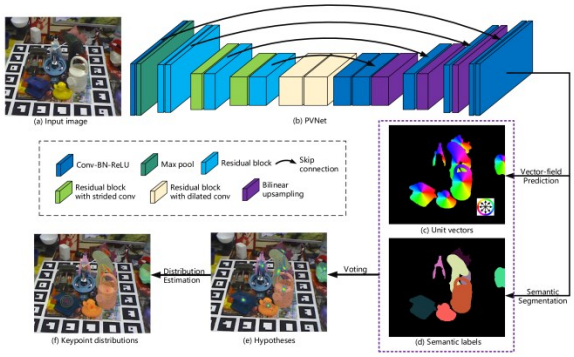

Figure 4. PVNet model architecture [10]

3.2.3 EfficientPose (EPOS)

EPOS extends the 2D object detection architecture family EfficientDet in an intuitive way to also predict the 6D poses of objects. Two extra subnetworks are added to predict the translation and rotation of objects, analogous to the classification and bounding box regression subnetworks of other such models. Since these subnets are relatively small and share the computation of the input feature maps with the already existing networks, we are able to get the full 6D pose without much additional computational cost. Through the seamless integration in the EfficientPose architecture (Figure 5), the approach is also capable of detecting multiple object categories as well as multiple object instances and can estimate their 6D poses all within a single shot. Because the 6D pose is directly obtained through regression, there is no need for further post-processing steps like RANSAC and PnP. This makes the runtime of our method nearly independent from the number of objects per image [4].

3.3. Evaluation Metrics

3.3.1 2D projection error

A pose estimate is accepted by the 2D projection error metric if, for every single vertex of the model, the distance between 2D projection of the 3D object mesh vertices using estimate and ground truth pose is less than 5 pixels. This metric greatly emphasizes accuracy in the 2D projection and de-emphasizes accuracy in the actual 3D location of the

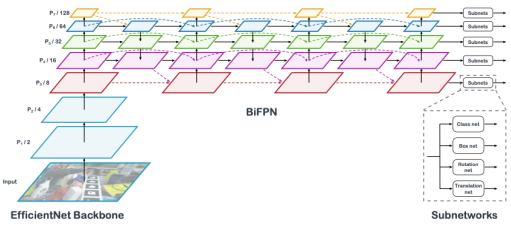

Figure 5. EPOS model architecture [4]

object, which makes it well-suited for determining suitability for 2D applications such as in augmented reality [2].

3.3.2 ADD metric

A pose estimate is accepted by the ADD metric in the following manner. A 6D pose estimate is taken to be correct if the mean distance between true 3D coordinate of mesh wire vertices and those estimated given the pose is less than 10% of the object's diameter. For most objects, threshold is approx. 2cm but for smaller objects (like ape) threshold reduces to 1cm. For rotationally symmetric objects, with one degree of rotational freedom the metrics is changed as following [18].

$$s = \frac{1}{|\mathcal{M}|} \sum_{x_1 \in \mathcal{M}} \min_{\mathcal{M}} ||(\mathbf{R}\mathbf{x} + \mathbf{t}) - (\widehat{\mathbf{R}}\mathbf{x} + \widehat{\mathbf{t}})||$$

where $(\mathbf{R}, \mathbf{t})$ are the ground truth rotation and translation, $(\widehat{\mathbf{R}}, \widehat{\mathbf{t}})$ are the predicted ones, and $\mathcal{M}$ the vertex set of 3D model.

3.3.3 5 cm 5 degree metric

A pose estimate is accepted by the 5 cm 5 degree metric based on the following criteria. Let r, t be the ground-truth rotational and translational components and r', t' be their estimated equivalents. An estimated pose is accepted by the metric if and only if:

$$e_r < 5^\circ \text{ and } e_t < 5 \text{ cm}$$

where $e_r = r - r'$ and $e_t = t - t'$ [7].

4. Results

4.1. Quantitative Results

Referring to Figure 6 and Table 1, we observe that EfficientPose has the highest mean accuracy (96.74%) followed by PVNet (83.67%) and SSD-6D (46.84%) for the LINEMOD dataset evaluated using the ADD metric. However, for the other two metrics: 2D projection error and 5 cm/5 degree, we observe that EfficientPose falls behind in performance to PVNet (see Table 7 and 8).

	SSD-6D	PVNet	EPOS
ape	21.90	46.48	**87.71**
benchvise	69.40	99.32	**99.61**
cam	30.49	86.67	**97.94**
can	68.85	96.36	**98.52**
cat	37.23	77.74	**98.00**
driller	62.64	96.43	**99.90**
duck	23.19	54.93	**90.99**
holepuncher	39.11	80.40	**95.05**
iron	59.86	98.88	**99.69**
lamp	55.76	99.52	**100.00**
mean	46.84	83.67	**96.74**

Table 1. ADD Metric on LINEMOD

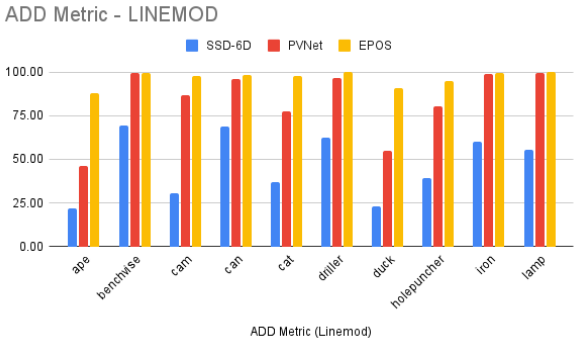

Figure 6. ADD Metric on LINEMOD

	SSD-6D	PVNet	EPOS
ape	93.24	98.29	**98.48**
benchvise	91.18	**99.81**	93.40
cam	86.96	**99.31**	98.24
can	93.80	**99.80**	92.22
cat	94.31	**99.70**	98.40
driller	73.44	**97.32**	92.27
duck	93.80	**98.78**	98.59
holepuncher	90.29	**100.00**	98.86
iron	63.53	**99.39**	96.83
lamp	57.10	**98.37**	88.58
mean	83.77	**99.08**	95.59

Table 2. 2D Projection Error on LINEMOD

On the Occlusion LINEMOD dataset, however, EfficientPose outperforms all the models for all three metrics of accuracy: ADD (71.58%), 2D projection (79.99%) and 5cm 5degree (55.82%), as seen in Tables 4, 5, 6 and Figures 9, 10, 11. PVNet is the second-best model in terms of performance on 6D pose estimation of occluded objects.

Figure 7. 2D Projection Error on LINEMOD

	SSD-6D	PVNet	EPOS
ape	41.81	**90.86**	66.57
benchvise	80.23	**99.81**	71.19
cam	52.65	**99.31**	82.16
can	83.17	**99.70**	62.01
cat	42.12	**98.20**	67.76
driller	60.95	**98.02**	81.47
duck	38.59	**91.46**	60.75
holepuncher	55.38	**97.72**	78.88
iron	30.34	**97.65**	88.76
lamp	51.54	**99.42**	71.98
mean	53.68	**97.21**	73.15

Table 3. 5 cm/5 degree Metric on LINEMOD

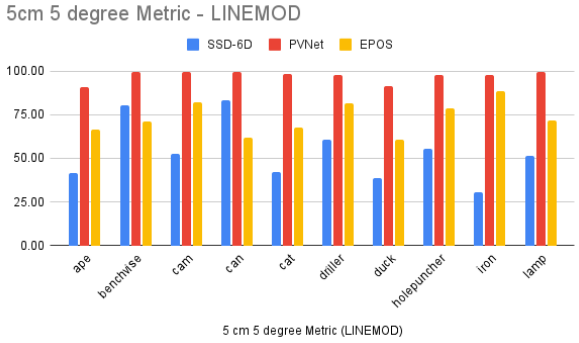

Figure 8. 5 cm/5 degree Metric on LINEMOD

SSD-6D has the lowest performance in every metric across every dataset.

4.2. Qualitative Results

Qualitatively, we observe that EfficientPose determines the rotational position of the objects less accurately than PVNet, which may contribute to its lower score in the 5 degree 5 cm and 2D Projection metrics. As seen in Figure 12, PVNet captures the corners of the object very accurately even with occlusions, while the same cannot be said about EfficientPose. This is true across many examples in LINEMOD, even when the object is fully visible, as shown in Figure 13. The green cube is the ground-truth and the blue is the estimated pose in all figures.

	SSD-6D	PVNet	EPOS
ape	0.60	15.21	**56.57**
can	9.44	62.47	**91.12**
cat	0.08	19.12	**68.58**
duck	0.26	33.04	**65.21**
holepuncher	4.38	40.33	**76.43**
mean	2.95	34.04	**71.58**

Table 4. ADD Metric on Occlusion LINEMOD

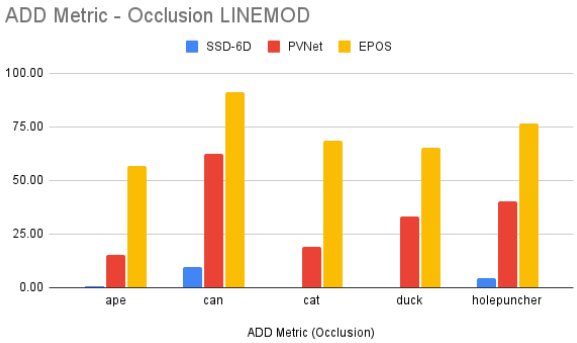

Figure 9. ADD Metric on Occlusion LINEMOD

	SSD-6D	PVNet	EPOS
ape	4.36	70.51	**75.66**
can	8.45	83.93	**85.85**
cat	2.11	63.27	**83.15**
duck	2.97	60.39	**76.98**
holepuncher	7.27	68.03	**78.30**
mean	5.03	69.23	**79.99**

Table 5. 2D Projection Error on Occlusion LINEMOD

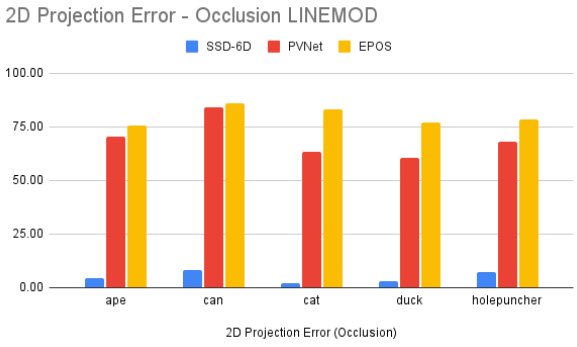

Figure 10. 2D Projection Error on Occlusion LINEMOD

	SSD-6D	PVNet	EPOS
ape	0.60	38.55	**39.60**
can	4.47	68.85	**72.20**
cat	0.00	19.12	**68.98**
duck	0.09	16.30	**44.48**
holepuncher	2.15	46.28	**53.85**
mean	1.46	37.82	**55.82**

Table 6. 5 cm/5 degree Metric on Occlusion LINEMOD

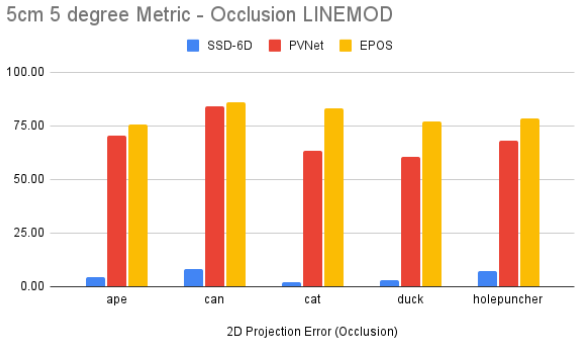

Figure 11. 5 cm/5 degree Metric on Occlusion LINEMOD

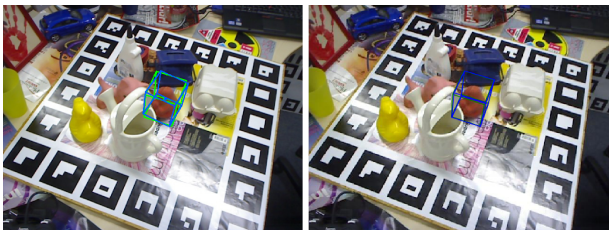

Figure 12. Poor rotational accuracy with EPOS (Left: EPOS; Right: PVNet).

Despite the fact that PVNet tends to be very accurate and precise on LINEMOD most of the time and not demonstrating rotational inaccuracies as shown in the previous figures, it does occasionally exhibit catastrophic modes of failure. These modes of failure are not present in the test dataset in EPOS. As seen in the left image in Figure 14, there are examples of PVNet not only being unable to detect the correct object, but giving an estimate of object out of the main test frame entirely. In the right image of Figure 14, we see that it also exhibits unpredictable behavior with some estimates, which may be due to numerical instability in the model. These catastrophic failures contribute to an overall lack of robustness in PVNet as compared to EfficientPose.

SSD-6D's results with the test dataset of LINEMOD exhibits both rotational and translational inaccuracy compared with PVNet and EfficientPose. As seen in Figure 15, while the model is successful in detecting the object, it falls short in precise keypoint detection and pose estimation leading to

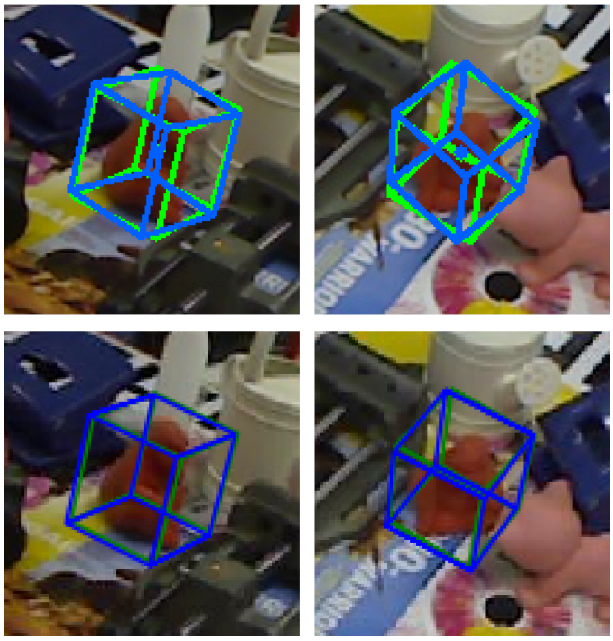

Figure 13. More examples of poor rotational accuracy with EPOS (Top: EPOS; Bottom: PVNet).

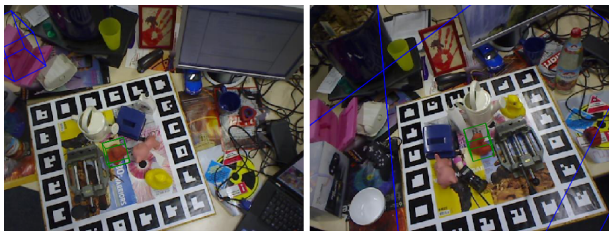

Figure 14. Catastrophic failure with PVNet.

Figure 15. Lack of translational and rotational accuracy with SSD-6D (Left: SSD-6D; Right: PVNet)

an estimate that is only roughly correct. However, as seen in Figure 16, SSD-6D also occasionally exhibits catastrophic modes of failure, and as such, we make the evaluation that it falls short in robustness.

4.3. Discussion

Taking a look at the qualitative results, we observe that the EfficientPose model demonstrates the best performance on the ADD metric. However, it falls behind to PVNet in both the 5 cm/5 degree and 2D projection metrics. This tells

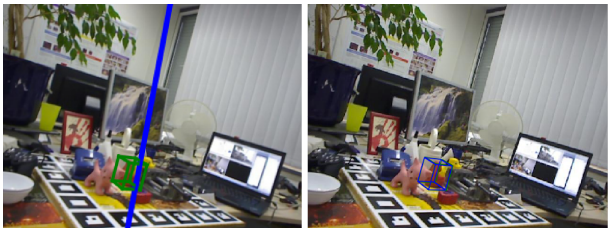

Figure 16. Catastrophic failure with SSD-6D (Left: SSD-6D; Right: PVNet)

us several things about the model: firstly, the 5 cm/5 degree metric requires both rotational and translational estimates to be within a certain error threshold, whereas the ADD metric is a combined metric in which weakness in one area can be compensated for by exceedingly high performance in another area.

Looking at the qualitative results of EfficientPose, specifically instances such as Figure 12 and 13, we see that EPOS demonstrates poor rotational accuracy on LINEMOD but very good translational accuracy. Thus, we posit that the high performance on the ADD metric is caused by the translational performance compensating for the rotational performance.

Given that the EfficientPose architecture contains discrete subnetworks for increasing rotational and translational performance, as seen in Figure 5, perhaps the rotational performance can be improved upon by additional tuning or training of the rotational subnetwork[4]. While rotational performance is not very relevant for some vision-related tasks, such as detection, it is very important for many tasks such as applications in Augmented Reality. We note that the authors of EfficientPose only included ADD data in their paper. Thus, we suggest that future papers on solutions to this task include both rotational and translational metrics, ideally as separate data, so that we can better evaluate the strengths and weaknesses of each model and their applicability to the tasks at hand.

Given PVNet's lower quantitative results, specifically within the ADD metric which seems to be the default benchmark for this task, model aggregation sites such as Paperswithcode rank EfficientPose to be a higher-performing model for this task than PVNet. However, we do not think that we should be hasty in discounting PVNet as a viable alternative for a state-of-the-art model.

Figure 7 and 8 show that PVNet manages to outperform EfficientPose in several evaluation metrics, and as mentioned before, these types of evaluation metrics may be better suited to evaluating performance in different types of vision tasks. Qualitatively speaking, the visual performance of PVNet is much more precise and demonstrates more exacting estimation when compared to EfficientPose (see Figure 13 for good examples of this.) We suggest that future model architects refer to PVNet's subnetwork training for specifically rotational accuracy to see how it's possible to improve on the very good performance that this model exhibits.

As mentioned above, however, problems with robustness and consistency arise with PVNet in a manner that is not observed in EfficientPose, for instance in Figure 14. These inconsistencies will prevent PVNet in its current state from reaching a production-level state of readiness for any type of application except ones that are non-critical in nature, and given the frequency that 6D Pose Estimation tasks are used in such critical scenarios (e.g. self-driving applications.)

The results from the SSD-6D model leave much to be desired, which is expected given that it is the oldest of these models and both EfficientPose and PVNet stated improvements compared to this as a benchmark model [18, 10, 4]. We included this as a baseline comparison, and it is clear that over a span of three years, performance and consistency on the task at hand has been improved greatly. It is quite clear that the bounding box methods used in SSD-6D are non-performant, especially on tasks involving occlusion, and it is good to see that later models such as PVNet and EfficientPose move away from a bounding-box and corner-based approach to a more generalized and denser approach.

5. Conclusion

Among the three approaches we evaluated for the 6D pose estimation task, we found EfficientPose to be the most robust as well as the most performant on the ADD metric on the LINEMOD and Occlusion LINEMOD datasets. The non-bounding-box based approaches proved to demonstrate higher performance, and significantly so when the task included occlusions. We note that PVNet does demonstrate very good performance, especially in the 2D projection and 5 cm/5 degree metrics, which make it a suitable contender for EfficientPose in some types of tasks. However, robustness concerns prevent it from being functional in a production setting. As such, we would recommend future model creators evaluate their models both quantitatively and holistically to ensure that no performance is being left on the table. We note that SSD-6D is outdated, and does not perform as well as the other two models in any of the tasks. The newer architectures, such as the approach taken by EfficientPose, extends earlier architectures in intuitive and efficient ways and combines efficiencies of the base networks with additional benefits yielded by subnetworks. Although these subnetworks can contribute to great performance if applied correctly, as seen in EPOS's high performance in many metrics, we should be careful to train them for optimal performance. Issues such as the poor rotational performance of EPOS compared with PVNet can possibly be ameliorated through careful tuning and training.

5.1. Future Work

Models we evaluated for our project mainly involves estimating the 6D pose of the object with and without occlusions and truncation. Future work in this area should also include velocity tracking-based approaches, as well as those that extend 6D pose estimation using Extended Kalman filters. Velocity tracking of objects can have enormous applications for autonomous driving, clash detection, workplace safety assessment, and Augmented Reality.

References

[1] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In *ECCV*, 2014.

[2] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and C. Rother. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3364–3372, 2016.

[3] E. Brachmann, F. Michel, A. Krull, Y. Yang, S. Gumhold, and C. Rother. *Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image*, pages 3364–3372. IEEE, United States, 2016.

[4] Y. Bukschat and M. Vetter. Efficientpose: An efficient, accurate and scalable end-to-end 6d multi object pose estimation approach, 2020.

[5] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox. Poserbpf: A rao-blackwellized particle filter for 6d object pose tracking. *CoRR*, abs/1905.09304, 2019.

[6] S. Hinterstoißer, V. Lepetit, S. Ilic, S. Holzer, G. R. Bradski, K. Konolige, and N. Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *ACCV*, 2012.

[7] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox. Deepim: Deep iterative matching for 6d pose estimation. In *European Conference on Computer Vision (ECCV)*, 2018.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015.

[9] D. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999.

[10] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *CVPR*, 2019.

[11] N. A. Piga, Y. Onyshchuk, G. Pasquale, U. Pattacini, and L. Natale. ROFT: real-time optical flow-aided 6d object pose and velocity tracking. *CoRR*, abs/2111.03821, 2021.

[12] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[13] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242, 2016.

[14] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

[15] R. Rios-Cabrera and T. Tuytelaars. Discriminatively trained templates for 3d object detection: A real time scalable approach. In *2013 IEEE International Conference on Computer Vision*, pages 2048–2055, 2013.

[16] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International Journal of Computer Vision*, 66:231–259, 2005.

[17] H. Su, C. R. Qi, Y. Li, and L. J. Guibas. Render for CNN: viewpoint estimation in images using cnns trained with rendered 3d model views. *CoRR*, abs/1505.05641, 2015.

[18] B. Tekin, S. N. Sinha, and P. Fua. Real-Time Seamless Single Shot 6D Object Pose Prediction. In *CVPR*, 2018.

[19] S. Tulsiani and J. Malik. Viewpoints and keypoints. *CoRR*, abs/1411.6067, 2014.

[20] B. Wen, C. Mitash, B. Ren, and K. E. Bekris. se(3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains. *CoRR*, abs/2007.13866, 2020.