World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:15, No:8, 2021

6D Posture Estimation of Road Vehicles from Color Images

Yoshimoto Kurihara, Tad Gonsalves

Open Science Index, Computer and Information Engineering Vol:15, No:8, 2021 publications.waset.org/10012170/pdf

***Abstract*—**Currently, in the field of object posture estimation, there is research on estimating the position and angle of an object by storing a 3D model of the object to be estimated in advance in a computer and matching it with the model. However, in this research, we have succeeded in creating a module that is much simpler, smaller in scale, and faster in operation. Our 6D pose estimation model consists of two different networks – a classification network and a regression network. From a single RGB image, the trained model estimates the class of the object in the image, the coordinates of the object, and its rotation angle in 3D space. In addition, we compared the estimation accuracy of each camera position, i.e., the angle from which the object was captured. The highest accuracy was recorded when the camera position was 75°, the accuracy of the classification was about 87.3%, and that of regression was about 98.9%.

***Keywords*—**AlexNet, Deep learning, image recognition, 6D posture estimation.

I. Introduction

THE 6D posture estimation of objects is a very important technology in the field of robotics, automatic driving and nursing care robots. The 6D posture estimation of an object is a technique to estimate the position of the object to be estimated, i.e., the coordinates in the x, y, and z axes, and the direction of the object to be estimated, i.e., the rotation angle around the x, y, and z axes, when a point in the image is set as the origin. If this estimation becomes possible, robots will be able to estimate more than humans, and will be able to predict the next moment's situation from the direction of other vehicles or pedestrians in an automatic driving situation [1], [2] or to grasp an object accurately even with a thin arm [3]. Moreover, it also has tremendous applications in augmented reality [4].

The 6D pose estimation is easier for objects with texture. However, the estimation of untextured objects poses a challenging task [5]. Various methods have been proposed for estimating the posture of an object by matching it with a 3D model stored beforehand in a computer, but the latest methods are all computationally very expensive, and take a long time to estimate the posture [6]. Brachmann et al. have demonstrated the feasibility of 6D pose estimation from a single RGB image [7]. This has motivated us to experiment 6D pose estimation from single RGB images, because of lower computational load and faster response.

Since AlexNet [8] won the ILSVRC (International Large Scale Visual Recognition Challenge) in 2012 by a large margin over the accuracy of the second-ranked networks, which contained manually tuned parameters, the use of CNNs has become common in image recognition. Currently, CNNs are used for various image recognition applications including camera images [9]-[11], and various networks have been proposed as the successor of AlexNet. The accuracy of CNNs is constantly improving. Hence, this study uses AlexNet to estimate the posture of 3D objects. Its greatest advantage is that it requires relatively less computational resources and runs faster than other existing models.

The contribution of our 6D model is in the area of autonomous or self-driving land vehicles. In the autonomous driving technology, the positioning of the visual camera is extremely important to gather the most relevant information for driving. As demonstrated by our deep learning model, if the front camera is placed at an angle of 75° with respect to the vehicle plane, the classification as well as pose estimation accuracy of objects in front of the driving vehicle is optimal.

This paper is organized as follows: Section II introduces related studies on 6D pose estimation, Section III describes our deep learning network model. Section IV explains the creation of dataset and the experimental setup, while Section V discusses the experimental results. The paper closes with a brief conclusion, indicating points for further research.

II. Related Works

There are many studies on 6D pose recognition of objects found in literature. For example, PoseCNN and DeepIM are two of the recent methods. Fundamental to PoseCNN model is a CNN model which estimates the 3D co-ordinates of an object. The 3D rotation of the object is estimated by means of regression [12], [13]. DeepIM [14] is a model that estimates the pose of an object by storing a 3D model of the object in the computer in advance, rendering images of the object viewed from various angles, and repeatedly matching these images with the pose of the detected object in the observed images. In addition to these, there is research on posture estimation using template matching which considers images and their gradients to detect objects, making them suitable for detecting untextured objects. They can directly provide a coarse estimation of the object pose which is especially important for robots interacting with their environment [15]-[18].

Other methods include posture estimation by detecting feature points of objects using deep neural networks and linking them to the PnP problem [19]. In addition, there is research that uses RGB-D images for pose estimation [7], [20], [21]. One method often used is to estimate the reference pose of an object

Y. Kurihara is with the Dept. of Information & Communication Sciences, Sophia University, Tokyo, Japan (e-mail: y-kurihara-8u6@eagle.sophia.ac.jp).
T. Gonsalves, PhD, is with the Dept. of Information & Communication Sciences, Sophia University, Tokyo, Japan (corresponding author; e-mail: t-gonsal@ sophia.ac.jp).

Open Science Index, Computer and Information Engineering Vol:15, No:8, 2021 publications.waset.org/10012170/pdf

from a color image, and then repeat the process of position and pose estimation by selecting the nearest neighbor points in each point cloud, using ICP and other methods [22].

III. Network Model

The network constructed in this study is shown in Fig. 1. The network simultaneously learns two models: a model for classifying objects, whose output is the type of object (hereinafter referred to as the classification model), and a model for pose estimation, whose output is the coordinates in the x, y, and z axes and the rotation angle around the x, y, and z axes (hereinafter referred to as the regression model). Specifically, the camera images described in Section 4 *A* are input to the two models, and the necessary parts are extracted from the text file described in Section 4 *A* and correspond to the output of each model in the network. Both the classification model and the regression model are based on AlexNet. The loss function is NLL Loss for the classification model, and RMSE Loss for the regression model.

For simplicity, we use up to nine objects for detection, but it is possible to estimate the pose of many objects by increasing the number of output nodes.

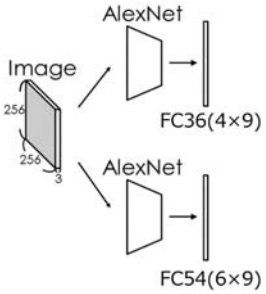


Fig. 1 Network model for 6D pose estimation

IV. ML Experiment

A. Creation of Dataset

The dataset was created using the game development platform Unity. Three types of objects (white vehicle, truck, and tank truck) are placed at random positions on the road at the bottom of Fig. 2, and the camera is fixed at a point on the arc in the figure. The image which the camera shot was output to an external file. At the same time, the name of the object, the relative coordinates of the object with the center of the arc in Fig. 2 as the origin (hereinafter referred to as the coordinates of the object), and the rotation angles around the x-, y-, and z-axes (hereinafter referred to as the posture of the object) were output to a text file. Using this text file and the camera images as a set of data, 30,000 data records were created for each of the camera positions of 0°, 15°, 30°, 45°, 60°, 75°, and 90°. Here, "when the camera position is x°" means "when the angle between the direction of the camera's line of sight and the horizon is x°", and the same expression is used hereafter.

Figs. 3-5 are examples of input images. Since it is difficult to understand the disparity for each camera position with these images alone, Figs. 6-8 are images taken by fixing the position of the object and moving only the camera position.

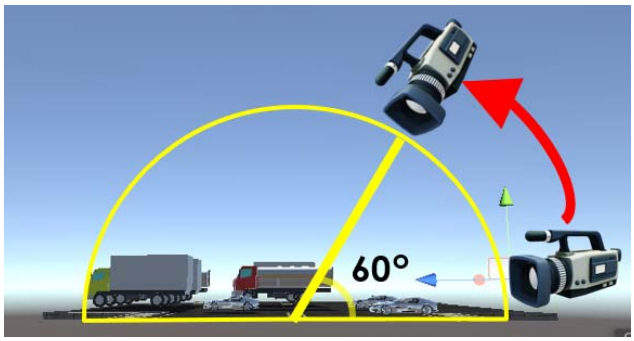


Fig. 2 Data set creation setting



Fig. 3 Input image (camera position: 0°)



Fig. 4 Input image (camera position: 75°)



Fig. 5 Input image (camera position: 90°)

Open Science Index, Computer and Information Engineering Vol:15, No:8, 2021 publications.waset.org/10012170/pdf



Fig. 6 Image visibility (camera position: 0°)



Fig. 7 Image visibility (camera position: 75°)

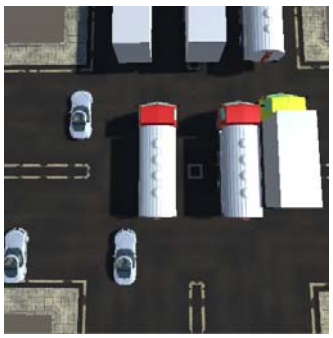


Fig. 8 Image visibility (camera position: 90°)

B. Experimental Setup

The batch size was 256, the number of epochs was 20, and the learning coefficients were 1.0×10^{-7} for classification, 1.0×10^{-4} for regression, and 5.0×10^{-4} for weight decay. We chose this value for the learning coefficient because the learning converges very quickly when the learning coefficient is large, and the weights tend to fall into local optima when the learning coefficient is smaller than this value.

The experimental environment used in this study is shown in Table I.

TABLE I
EXPERIMENTAL ENVIRONMENT IN THIS STUDY

CPU	AMD® Ryzen threadripper 2990wx 32-core processor × 64
GPU	NVIDIA GeForce GTX 1080 Ti/PCIe/SSE2
Main memory	62.8 GB
Programming language	Python3.6
Framework	pytorch

V. RESULTS

For reasons of space, only results that are considered important are described below. Figs. 9-11 show the graphs of the change in accuracy per epoch for the classifications at 0°, 75°, and 90°, respectively, and Figs. 12-14 show the change in loss per epoch for the classifications, respectively, Figs. 15-17 show the change of loss per epoch in regression, respectively.

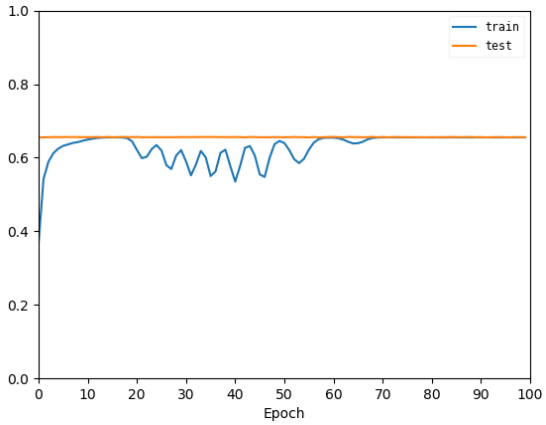


Fig. 9 Changes in accuracy for each epoch in the classification (camera position: 0°)

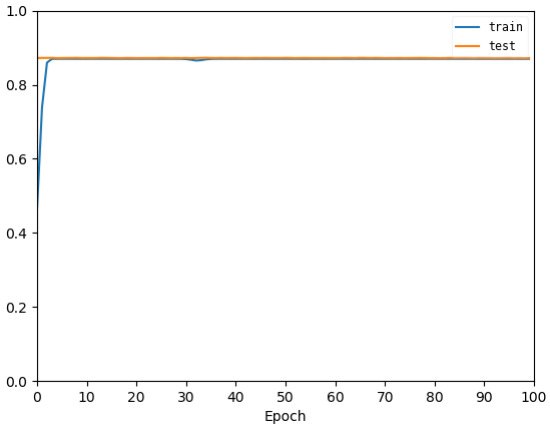


Fig. 10 Changes in accuracy for each epoch in the classification (camera position: 75°)

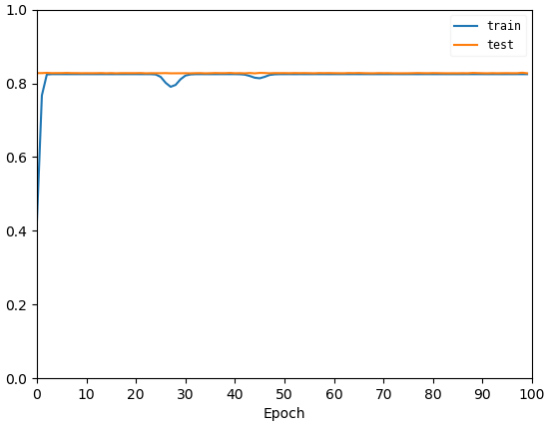


Fig. 11 Changes in accuracy for each epoch in the classification (camera position: 90°)

Open Science Index, Computer and Information Engineering Vol:15, No:8, 2021 publications.waset.org/10012170/pdf

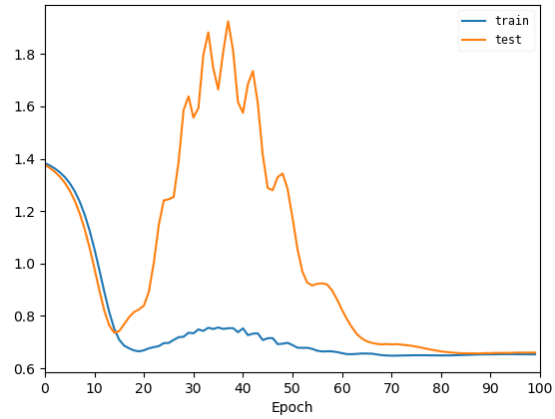


Fig. 12 Changes in loss per epoch in classifications (camera position: 0°)

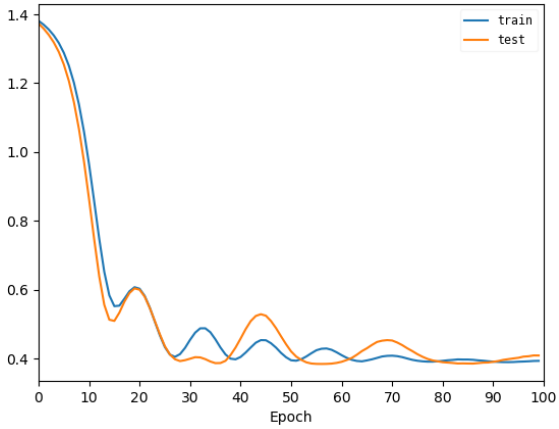


Fig. 13 Changes in loss per epoch in classifications (camera position: 75°)

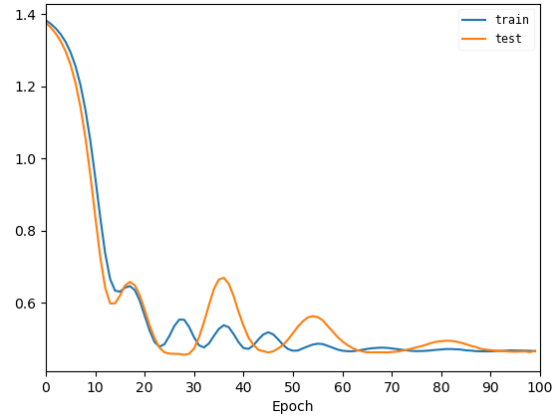


Fig.14 Changes in loss per epoch in classifications (camera position: 90°)

Figs. 18-20 summarize the best result for each angle. Fig. 18 is a graph of the highest accuracy of accuracy for class classification, Fig. 19 is a graph of the highest accuracy of loss for class classification, and Fig. 20 is a graph of the highest accuracy of loss for each epoch for regression.

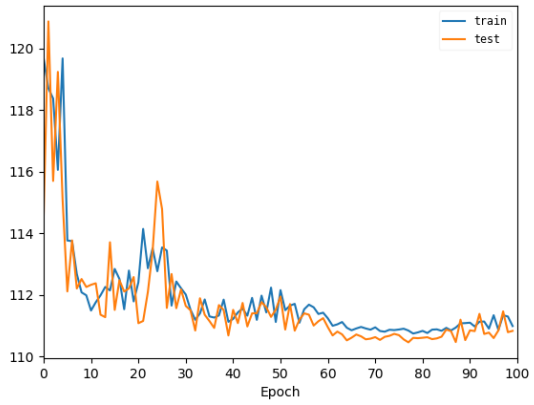


Fig. 15 Change in loss per epoch in regression (camera position: 0°)

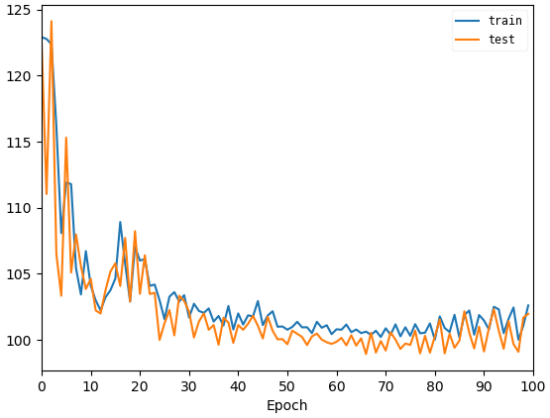


Fig. 16 Change in loss per epoch in regression (camera position: 75°)

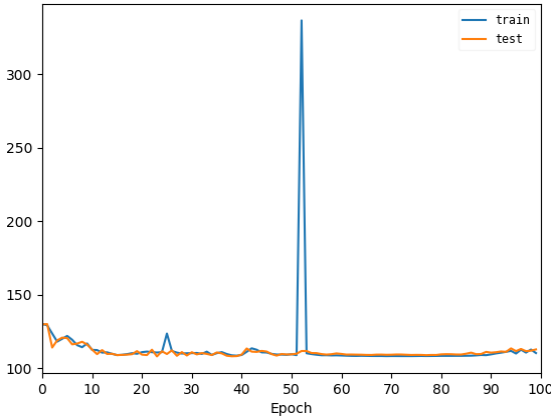


Fig. 17 Change in loss per epoch in regression (camera position: 90°)

The highest accuracy was recorded when the camera position was 75°, the accuracy of the classification was about 87.3%, the loss of the classification was about 0.392, and the loss of the regression was about 98.9.

VI. Conclusion

For the dataset used in this study, the highest accuracy is obtained when the camera is positioned at 75° resulting in an accuracy of 87.3% for classification and about 98.9% for

regression.

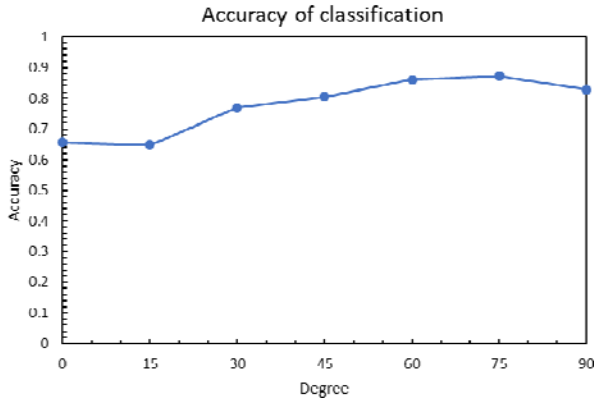


Fig. 18 Comparison of the accuracy of classifications for each angle

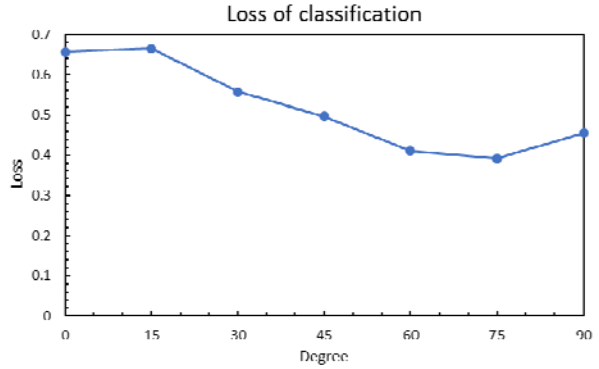


Fig. 19 Comparison of the loss of classifications for each angle

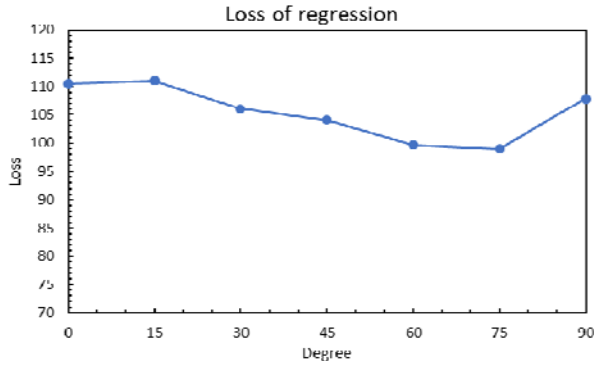


Fig. 20 Comparison of the loss of regression for each angle

In this study, we adopted AlexNet because of its simplicity in implementation; accuracy concerns were secondary in this preliminary study. In the future, we would like to improve the accuracy by adopting relatively new networks such as ResNet. Further, we used the Unity automobile asset to create a dataset necessary for training automated driving-related technologies, but we believe the same results can be obtained by using other assets. For example, when we apply the network to the research of garbage sorting robots, we can obtain the same results by using the assets of plastic bottles and cans.

REFERENCES

[1] X. Chen, H. Ma, J. Wan, B. Li and T. Xia, "Multi-view 3D Object Detection Network for Autonomous Driving," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6526-6534, doi: 10.1109/CVPR.2017.691.

[2] D. Wu, Z. Zhuang, C. Xiang, W. Zou and X. Li, "6D-VNet: End-To-End 6DoF Vehicle Pose Estimation from Monocular RGB Images," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019, pp. 1238-1247, doi: 10.1109/CVPRW.2019.00163.

[3] Menglong Zhu, Konstantinos G Derpanis, Yinfei Yang, Samarth Brahmbhatt, Mabel Zhang, Cody Phillips, Matthieu Lecce, and Kostas Daniilidis. Single image 3d object detection and pose estimation for grasping. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 3936–3943. IEEE, 2014.

[4] F. Tang, Y. Wu, X. Hou and H. Ling, "3D Mapping and 6D Pose Computation for Real Time Augmented Reality on Cylindrical Objects," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, no. 9, pp. 2887-2899, Sept. 2020, doi: 10.1109/TCSVT.2019.2950449.

[5] C. Wu, L. Chen, Z. He and J. Jiang, "Pseudo-Siamese Graph Matching Network for Textureless Objects' 6D Pose Estimation," in IEEE Transactions on Industrial Electronics, doi: 10.1109/TIE.2021.3070501.

[6] A. Doumanoglou, R. Kouskouridas, S. Malassiotis and T. Kim, "Recovering 6D Object Pose and Predicting Next-Best-View in the Crowd," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 3583-3592, doi: 10.1109/CVPR.2016.390.

[7] 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," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 3364-3372, doi: 10.1109/CVPR.2016.366.

[8] Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks". NIPS12, VoL1, pp-1097-1105, (2012).

[9] Komatsu R, Gonsalves T. Comparing U-Net Based Models for Denoising Color Images. AI. 2020; 1(4):465-486. https://doi.org/10.3390/ai1040029

[10] Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 2014, arXiv:1409.1556. Available online: https://arxiv.org/abs/1409.1556 (accessed on 1 May 2020).

[11] Galea, C.; Farrugia, R.A. Matching Software-Generated Sketches to Face PhotographsWith a Very Deep CNN, Morphed Faces, and Transfer Learning. IEEE Trans. Inf. Forensics Secur. 2017, 13, 1421–1431.

[12] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, Dieter Fox, "PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes", arXiv preprint arXiv: 1711.00199, (2019).

[13] Alirezazadeh, P., Yaghoubi, E., Assunção, E., Neves, J. C., & Proença, H. (2019, September). Pose Switch-based Convolutional Neural Network for Clothing Analysis in Visual Surveillance Environment. In 2019 International Conference of the Biometrics Special Interest Group (BIOSIG) (pp. 1-5). IEEE.

[14] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, Dieter Fox, "DeepIM: Deep Iterative Matching for 6D Pose Estimation", arXiv preprint arXiv: 1804.00175, (2019).

[15] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab, "Dominant Orientation Templates for Real-Time Detection of Texture-Less Objects," in IEEE Conference on Computer Vision and Pattern Recognition, 2010.

[16] Muja, M., Rusu, R. B., Bradski, G., & Lowe, D. G. (2011, May). Rein-a fast, robust, scalable recognition infrastructure. In 2011 IEEE international conference on robotics and automation (pp. 2939-2946). IEEE.

[17] Hinterstoisser S, Cagniart C, Ilic S, Sturm P, Navab N, Fua P, Lepetit V, "Gradient response maps for real-time detection of texture less objects.", IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 34(5):876-888 (2012)

[18] E. Muñoz, Y. Konishi, C. Beltran, V. Murino and A. Del Bue, "Fast 6D pose from a single RGB image using Cascaded Forests Templates," 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016, pp. 4062-4069, doi: 10.1109/IROS.2016.7759598.

[19] Tremblay J, To T, Sundaralingam B, Xiang Y, Fox D,Birchfield S, "Deep object pose estimation for semantic robotic grasping of household

Open Science Index, Computer and Information Engineering Vol:15, No:8, 2021 publications.waset.org/10012170/pdf

objects.", In: Conference on Robot Learning, pp 306-316, (2018)

[20] V. L. Tran and H. -Y. Lin, "3D Object Detection and 6D Pose Estimation Using RGB-D Images and Mask R-CNN," 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2020, pp. 1-6, doi: 10.1109/FUZZ48607.2020.9177601.

[21] L. Peng, Y. Zhao, S. Qu, Y. Zhang and F. Weng, "Real Time and Robust 6D Pose Estimation of RGBD Data for Robotic Bin Picking," 2019 Chinese Automation Congress (CAC), 2019, pp. 5283-5288, doi: 10.1109/CAC48633.2019.8996450.

[22] Zeng A, Yu KT, Song S, Suo D, Walker E, Rodriguez A, Xiao J, "Multi-view self-supervised deep learning for 6D pose estimation in the amazon picking challenge.", IEEE International Conference on Robotics and Automation (ICRA), pp 1386-1383, (2017)

Open Science Index, Computer and Information Engineering Vol:15, No:8, 2021 publications.waset.org/10012170/pdf