Image Processing Techniques to Improve Deep 6DoF Detection in RGB Images

Heitor Felix
Voxar Labs/CIn
Universidade Federal de Pernambuco
Recife, Brazil
hcf2@cin.ufpe.br

Francisco Simões
Departamento de Informática
IFPE - Campus Belo Jardim
Belo Jardim, Brazil
francisco.simoes@belojardim.ifpe.edu.br

Kelvin Cunha
Voxar Labs/CIn
Universidade Federal de Pernambuco
Recife, Brazil
kbc@cin.ufpe.br

Veronica Teichrieb
Voxar Labs/CIn
Universidade Federal de Pernambuco
Recife, Brazil
vt@cin.ufpe.br

***Abstract*—Six degrees of freedom (6DoF) Object Detection has great relevance in computer vision due to its use in applications on several areas, such as augmented reality and robotics. Even with the improved results provided by deep learning techniques, object detection of textured and non-textured objects is still a challenge. The objective of this work was to seek improvements in the six degrees of freedom detection of non-textured objects using a Convolutional Neural Network (CNN) approach through the preprocessing of the images that were used for training the network. A State of the art research was carried out on techniques that use CNN to detect objects in six degrees of freedom. We also searched for filters with enhancement factors for detection. Finally, a detection technique based on a CNN was selected and adapted to use single-channel images (grayscale) as input, instead of using three-channel images (RGB) as in the original proposition, aiming to increase its robustness while reducing the complexity of the input images. The technique was also tested with the application of two different preprocessing filters to enhance the objects' contours on the single-channel images, one being the "pencil effect", and the other based on local binary patterns (LBP). With this study, it was possible to evaluate the impact on the CNN detection performance due to the application of both of the filters. The proposed technique used with one channel images and the filters on the images still could not surpass the results of the technique with the three-channel image (RGB), although it indicated paths for improvement. The pencil filter also proved to be more robust than the LBP filter, as expected.**

***Index Terms*—Computer Vision, 6DoF Object Detection, Convolutional Neural Networks, Local Binary Patterns, Pencil Filter**

I. Introduction

The problem of locating an object in a scene, retrieving its position and orientation relative to the standpoint of the camera viewing it, is known as six degrees of freedom (6DoF) pose estimation. This problem has great relevance in computer vision because it enables different applications such as defining where virtual information will be added in a real scene to assist an industry professional in performing an augmented reality task [1] and also enables the interaction between a real object and a robot [2], [3]. By estimating the pose of an object in 6DoF, it is possible to use object detection techniques on images, in which the information of the object of interest is retrieved from each frame captured by the camera and related to a previously known information of the object to enable the recovery of the pose. Depending on the type of object, different visual and/or geometric characteristics can be used, eg.: texture, edges, color, contours, and others. Traditionally in computer vision, the most common 6DoF object detection techniques focus on textured objects [4], [5], due to the possibility of creating efficient and robust descriptors for various transformations and challenges as the object's scale, lighting, rotation, and other changes, etc. Due to the difficulty of creating descriptors for poorly textured objects, detecting such objects represents a major challenge for the area.

Recently, with the strengthening of deep learning area of study, more specifically the strengthening of the Convolutional Neural Networks (CNN) use, several techniques have been proposed to solve the problem of detecting poorly textured objects [6]–[8]. Such techniques have shown great potential due to the high precision and real-time performance achieved [8], [9]. Despite recent advances, many problems remain open. The need of a large amount of data for the operation of these solutions, the difficulty of generalizing the behavior of the techniques in challenging scenarios and the large computational costs that makes it difficult to use these techniques in devices with low computing power, such as mobile devices which are examples of open problems in the area.

Among the many approaches used to improve the performance of machine learning techniques, including artificial neural networks, data preprocessing can significantly improve the accuracy and robustness of inference [10]. Therefore, to be able to solve problems using machine learning approaches, various experiments and researches are performed to preprocess the data that will be used in network learning. In CNNs in which input data are images, image processing methods may be used to preprocess the data that will be used in the neural network [11].

In the work published by Rambach et al. [12], image processing was used in the data preprocessing step of a CNN designed to detect objects in 6DoF. The motivation behind the use of the contour extractor application, called ”pencil effect”, in this work, lies on the generalization of the image objects' appearance. In the preprocessing step, the three-channel RGB image is converted into a single-channel grayscale image and then the ”pencil effect” is applied to the image, which decreases the relevance of the object's appearance during training step. Other benefits observed after the application of the filter are the invariance of the results even when there are changes to the scene illumination and the highlighting of the objects' contours in the image. By the usage of the images' preprocessing technique, improvements were obtained in the accuracy of the estimation of the tested objects' pose by the use of images without preprocessing.

Among the different image processing techniques, Local Binary Patterns (LBP) [13] features a robust form of texture classification. After its first publication, the technique has undergone several improvements, LBP modifications have been published, and the technique can currently be used for face detection and unsupervised texture segmentation [14], [15].

The objective of this work is to test and evaluate image processing techniques that will be applied to preprocess images used in deep learning techniques for 6DoF detection of poorly textured objects. The specific objectives are:

1) Evaluate the preprocessing technique proposed by [12] for detecting objects in single-channel images using CNN;
2) Propose the use of another image processing technique based on LBP, that will be used on the data preprocessing stage of the images used by CNN to detect objects in 6DoF;
3) Evaluate and compare the proposed image processing with the processing used by [12] about 6DoF detection accuracy.

II. Related Works

In a study published by Rambach in [12], the use of an image preprocessing technique called “pencil effect” to detect objects with texture abstraction was demonstrated. The ”pencil effect” acts as an edge detector, the algorithm uses a dilatation filter with an elliptical structure to calculate the local maximum and divide, for each pixel, the value of the original image by the value of the dilated image. The technique is also robust to lighting variations, minimizes capture noise and highlights the edges of image objects. The purpose of the technique is to improve detection results by increasing its robustness. In Rambach's work, a CNN with a PoseNet [16] based architecture was used. Image preprocessing transforms three-channel RGB images into one-channel grayscale images to enable pencil effect application. The training was done by creating synthetic data from the dataset LINEMOD [17] with noise applied to the generated image. The work also tested two loss functions to be used during network training. With the new proposition, the developed technique achieved better results than networks using three-channel RGB images which were selected for comparison in the publication. The results motivated the use of the technique in real scenarios even with a polluted background and enabled augmented reality remote user assistance applications [18].

III. Image Preprocessing for Object Detection with CNN

A. CNN for object detection in 6DoF

The network proposed by Tekin et al. [8] uses the same architecture as the YOLO network by adding depth to the network output layer that enables the estimation of a bounding box of the detected object. The network output is the prediction of eight points, in 2D coordinates, and the bounding box centroid of the detected object's position, the object classification probabilities for each class used in the training, and the confidence value of its position. To get the position as a function of the rotation and translation of the object to the camera it is necessary to apply the PnP algorithm (Perspective-N-Points) using the bounding box points returned by the network, the intrinsic parameters of the camera used and the 3D points of the model of the detected object. For training, only the coordinates of the nine points of the bounding box of the object are used. Because it achieves good results about state-of-the-art 6DoF object detection and achieves real-time performance at approximately 50 fps, this network was chosen to be covered in this work.

B. Preprocessing

To use the preprocessing technique proposed in [12], which aims to improve detection results by increasing its robustness, only the pencil filter was integrated with the architecture used in [8]. During image preprocessing, the image is converted from a three-channel RGB image into a single-channel grayscale image. The image resulting from the application of the ”Pencil Effect” can be seen in Fig. 1.

In the tests of other image processing techniques, LBP was chosen to be applied. [19]. The LBP technique has been used since its proposition as a powerful resource for texture classification. The technique has also been widely used in facial detection and unsupervised texture segmentation [14], [15]. A variation of LBP was also used to calculate the texture contrast of the pixels of an image. The method also creates a point neighborhood like the circular neighborhood LBP. After calculating the points, a simple variance of the values is calculated. This results in higher values for higher contrast between the texture of neighboring pixels and smaller values for lower contrast, resulting in the extraction of contours from an image. In this work, the LBPs went through the same process mentioned for the ”pencil filter” described above.

C. Training

The dataset used during the training and testing of the proposed work is LINEMOD [17] which is also used in the

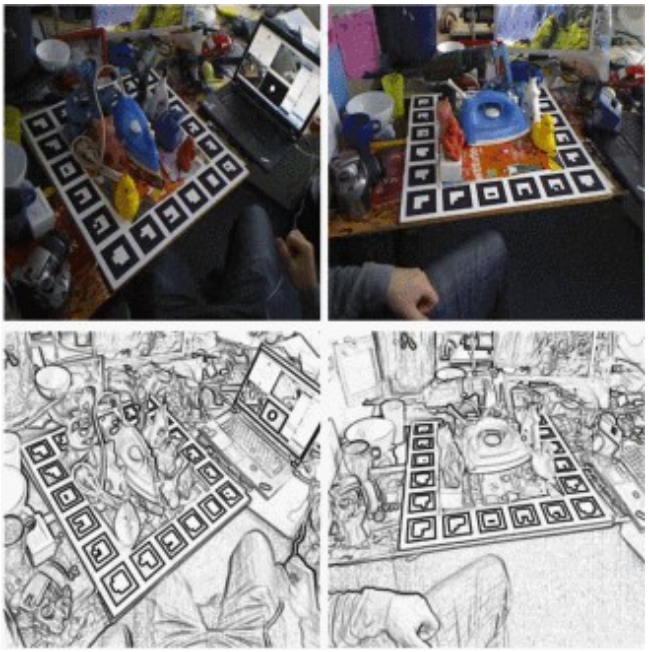

Fig. 1. Pencil Effect result. Adapted from [12]

works [8] and [12] for a reliable comparison with state-of-the-art techniques. Also, LINEMOD is one of the main datasets used for object detection, having several challenges such as occlusion, polluted background, blurring and lighting change.

LINEMOD is made up of thirteen objects, where each object has about twelve hundred color images in 640x480 JPEG format and a 3D model. Also, for each image, the binary mask for the image objects and the rotation and translation vector values of the 6DoF location of the object are informed.

In this work, the dataset has been divided into three parts for each object. 50% of the images were used for training, 25% of the images were used for validation and 25% were used for testing.

Training images went through a process of synthetic data augmentation as used by [8], [12]. During synthetic data augmentation, 100 new images are created for each image for the object being trained. In the image, the object is segmented from its binary mask. The segmented object is added to a random background from the dataset VOC2012 [20], besides that, the object in the image is rotated and translated aiming to avoid overfitting and to increase the number of images available for training. After the object is inserted into the background, the selected image preprocessing filter is applied.

The training is done with a total of 1,000 epochs for each object and for each image processing technique. After every five epochs, the training is validated using all images from the validation set. Validation set images do not have its background changed. Only the image preprocessing filter is applied so that the network is not skewed to the training set, which has a random and artificial domain, that is, making the training set hit a specific real dataset. After the end of each validation step, if the score found is the best up to the moment, the net weights are saved.

D. Evaluation Metrics

In the present work, two evaluation metrics of the 6DoF pose estimation were used. These metrics are also used in related works and evaluate the prediction of the pose of objects in different ways, which are relevant to various areas such as augmented reality. The metrics used will be the 2D Reprojection [21] and 3D Model Pose [21].

The 2D Reprojection metric is defined as the average of 2D reprojected object point hit on the image from the test set images, where n is the number of test images. The reprojection hit is calculated using (1), when the projection distance is lower or equal than five pixels, it is considered to be a hit. [21].

$$\Delta_{2D}(x_1, x_2) = \begin{cases} 1, & if\ \|x_1 - x_2\| \leq 5pixels \\ 0, & otherwise \end{cases} \quad (1)$$

The 6D Pose metric is defined as the average of correctly designed object points in the virtual environment, considering its position and orientation from the test set images. Projection hit is calculated using the 3D model points of the object, its true extrinsic parameter matrix, and its estimated extrinsic parameter matrix. For each point of the transformed 3D model using the extrinsic parameter matrix, the pose error is calculated from the distance between the transformed point using the correct parameter and the transformed point using the estimated parameter. The pose error is calculated using (2), where n is the total number of points in the 3D model of the object. 6D Pose hit is considered if m is lower or equal than 10% of object model diameter [21].

$$m = \frac{1}{n} \sum_{i=1}^{n} \|(r_{gt} x_i + t_{gt}) - (r_{pr} x_i + t_{pr})\| \quad (2)$$

IV. Results

All results were obtained using a computer with the Intel® Xeon® E5-2609 processor, 16.0 GB of RAM, and the Nvidia Geforce RTX 2080 Ti graphics card with a dedicated memory of 12.00 GB. Windows 10 64-bit version and CUDA version 10.0 was used. Python version 3.7, OpenCV version 3.4 and PyTorch version 1.1.0 were used for development. Results were divided into Preliminary Results and Overall Results. Due to the high cost of computational power and the long period required to train each dataset object for each preprocessing method, about 8 hours per training, performing preliminary tests made it possible to select only those techniques that obtained the highest results in the Preliminary Results to the deep analysis stage.

A. Preliminary results

In these tests, the ”pencil effect”, LBP and LBP VAR preprocessing techniques, all applied to the CNN object detection network proposed by [8], were evaluated. For evaluation, the 2D Reprojection and 6D Pose metrics were used. Preliminary

tests were performed using only the object *Ape* of LINEMOD dataset. The results obtained can be seen in Table I.

TABLE I
PRELIMINARY TEST RESULTS

Metric	LBP	LBP VAR	Pencil Effect
6D Pose	10.97	13.92	11.65
2D Reprojection	60.52	79.29	77.02

Due to the lower accuracy obtained from LBP compared to the other processing techniques evaluated, LBP was not explored in the Overall Tests.

B. Overall Results

In the Overall Tests, the first results obtained were from the execution of the "pencil effect" filter proposed by Rambach et al. [12] in the preprocessing of the image applied to the adopted CNN, which was proposed in Tekin et al. [8].

A comparison of the results obtained is carried out in the test performed with the values reported in the publication [12]. The detection result was compared using the same image processing technique, following the same training steps.

The analyzed CNN architectures are different. While in the results obtained by [12] the architecture used is based on Posenet [16], which predicts rotation and translation vectors directly, the architecture proposed by [8] predicts bounding box of the object and estimates the rotation and translation vectors in the sequence. The 6D Pose assessment metric was used. The results of the comparison can be seen in Table II.

TABLE II
NETWORK ARCHITECTURE CHANGE

Object	6D Pose (%)	
	Rambach Network from [12]	**Network from Tekin**
Ape	4.37	**11.65**
Benchvise	21.74	**40.92**
Cam	1.25	**29.67**
Can	2.09	**41.81**
Cat	2.54	**25.51**
Driller	12.46	**33.00**
Duck	4.78	**19.49**
Eggbox	1.43	**25.88**
Glue	7.38	**19.67**
Holepuncher	3.88	**22.33**
Iron	38.22	**39.58**
Lamp	27.35	**36.60**
Phone	5.39	**35.62**

Tests were also performed using LBP with texture contrast through pixel variance (LBP VAR) as the image preprocessing technique. The results obtained were compared with the results of the "pencil effect" also used as a preprocessing technique. In the tests of each technique, the same CNN with the same training steps was used. The comparison of the techniques was made using the 2D Reprojection and 6D Pose metrics. The results obtained from the 2D Reprojection metric, in this case, was better to evidence the differences in the results of the techniques. The result can be seen in Table III.

TABLE III
COMPARISON OF "PENCIL EFFECT" AND LBP VAR

Object	2D Reprojection (%)	
	Pencil Effect	**LBP VAR**
Ape	77.02	**79.29**
Benchvise	**49.83**	42.24
Cam	**66.33**	58.67
Can	**66.56**	54.18
Cat	**72.79**	63.27
Driller	**40.07**	36.03
Duck	**75.40**	60.70
Eggbox	**76.68**	**76.68**
Glue	**77.70**	69.84
Holepuncher	**66.99**	68.61
Iron	**43.40**	38.54
Lamp	**55.56**	50.98
Phone	**72.88**	69.28

Performing a direct comparison between the result obtained from the application of the "Pencil Effect" and the results reported by [8] in its publication. As in the previous result, the 2D Reprojection metric shows results with more evident differences than the 6D pose metrics. The result can be seen in Table IV.

TABLE IV
COMPARISON OF "PENCIL EFFECT" AND TEKIN RESULTS FROM [8]

Object	2D Reprojection (%)	
	Pencil Effect	**RGB Image**
Ape	77.02	**92.10**
Benchvise	49.83	**95.06**
Cam	66.33	**93.24**
Can	66.56	**97.44**
Cat	72.79	**97.41**
Driller	40.07	**79.41**
Duck	75.40	**94.65**
Eggbox	76.68	**90.33**
Glue	77.70	**96.53**
Holepuncher	66.99	**92.86**
Iron	43.40	**82.94**
Lamp	55.56	**76.87**
Phone	72.88	**86.07**

V. DISCUSSIONS

Analyzing the preliminary results, the application of LBP was not positive, especially when comparing the results of the LPB Var and the "Pencil Effect" techniques. LBP was proposed to highlight the texture of objects and was not effective in detecting poorly textured objects in an environment with occlusion and lighting variation. While LBP has as its characteristic to classify the textures of the image, LBP VAR and "pencil effect" preprocessing techniques have among their characteristics the highlight of the object's contours, thus providing the best values.

In the Overall Results, it was possible to observe a big difference between the results obtained by utilizing the same preprocessing technique on networks with different architectures. It was expected and improved due to what has been proven by [6], [7] and adopted by [8]. Estimating the pose

of objects in 6DoF with CNN is best done by predicting the points of the scene object and then recovering the rotation and translation coordinate values with the points rather than predicting the rotation and translation coordinates directly as is made in [16]. By analyzing the results it can be shown that this statement is also valid for grayscale single-channel images. Comparing the two networks that have the same goal, that use the same preprocessing techniques and training steps, it is evident that estimating the pose directly from the bounding box is more efficient.

Analyzing the comparison of the techniques used in network preprocessing, shown in Table III, the "Pencil Effect" was better than the LBP VAR. Although LBP VAR highlights object contours more clearly than the "pencil effect", LBP VAR has flaws in highlighting contours in regions that have similar textures. Another problem with the filter is its sensitivity to the variation of lighting and reflection, unlike the "pencil effect" that aims to deal with such problems. With the results and analysis, it can be concluded that the "pencil effect" is better at segmenting different objects for 6DoF object detection.

Even with the improvements obtained by changing the architecture of the CNN used, the comparison of the results shown in Table IV demonstrates that while the 'pencil effect' ensures better invariance to lighting changes, training CNN to detect poorly textured objects and to highlight the edges of objects, which is a relevant feature for detection, it was not possible to obtain better results than the same network using three-channel RGB images.

VI. Conclusions

In this research it was possible to compare variations of the technique proposed by Tekin et al. [8] by applying the "pencil effect" [12] and LBP image processing filters [19] on single-channel images.

It was possible to verify, as indicated by Tekin et al. [8], a greater precision in using network architecture to predict points and then compute the pose instead of directly computing the 6DoF pose, even for single-channel images. In addition, it was possible to observe better results in the use of the "pencil effect" than in the use of LBP.

This work also concludes that the technique proposed by Rambach et al. is sensitive to network variations and does not improve performance in any network where the pencil effect preprocessing technique is applied. More complex and deep networks such as Tekin et al. network, learn to highlight the important features of objects and benefit from the amount of information of three-channel images over single-channel images. Different network architectures show significantly different results for the same purpose. There may be a chance that there are layers that already perform contour segmentation and abstraction of lighting changes, which would make the benefits of applying the preprocessing techniques proposed in this work minimal. Researches that seeks to analyze the learning acquired by deep networks has relevance to problems such as how the networks react to the changes of information representation.

References

[1] P. Fraga-Lamas, T. M. Fernández-Caramés, Ó. Blanco-Novoa, and M. A. Vilar-Montesinos, "A review on industrial augmented reality systems for the industry 4.0 shipyard," *IEEE Access*, vol. 6, pp. 13 358–13 375, 2018.

[2] C. Moreno and L. Alberto, "Robot asistente para personas con problemas de movilidad," 2016.

[3] C. Hernandez, M. Bharatheesha, W. Ko, H. Gaiser, J. Tan, K. van Deurzen, M. de Vries, B. Van Mil, J. van Egmond, R. Burger *et al.*, "Team delft's robot winner of the amazon picking challenge 2016," in *Robot World Cup*. Springer, 2016, pp. 613–624.

[4] D. G. Lowe *et al.*, "Object recognition from local scale-invariant features." in *iccv*, vol. 99, no. 2, 1999, pp. 1150–1157.

[5] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[6] 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 the IEEE International Conference on Computer Vision*, 2017, pp. 3828–3836.

[7] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1521–1529.

[8] 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*, 2018, pp. 292–301.

[9] M. Garon and J.-F. Lalonde, "Deep 6-dof tracking," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 11, pp. 2410–2418, 2017.

[10] J. Huang, Y.-F. Li, and M. Xie, "An empirical analysis of data preprocessing for machine learning-based software cost estimation," *Information and software Technology*, vol. 67, pp. 108–127, 2015.

[11] D. Cireşan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," *arXiv preprint arXiv:1202.2745*, 2012.

[12] J. Rambach, C. Deng, A. Pagani, and D. Stricker, "Learning 6dof object poses from synthetic single channel images," in *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE, 2018, pp. 164–169.

[13] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Proceedings of 12th International Conference on Pattern Recognition*, vol. 1. IEEE, 1994, pp. 582–585.

[14] L. Guo, D. Xu, and Z. Qiang, "Background subtraction using local svd binary pattern," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 86–94.

[15] Y. Duan, J. Lu, J. Feng, and J. Zhou, "Context-aware local binary feature learning for face recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 5, pp. 1139–1153, 2018.

[16] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2938–2946.

[17] 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.

[18] J. Rambach, A. Pagani, M. Schneider, O. Artemenko, and D. Stricker, "6dof object tracking based on 3d scans for augmented reality remote live support," *Computers*, vol. 7, no. 1, p. 6, 2018.

[19] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 7, pp. 971–987, 2002.

[20] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[21] E. Brachmann, F. Michel, A. Krull, M. Ying Yang, S. Gumhold *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*, 2016, pp. 3364–3372.