Impact of Segmentation and Color Spaces in 6D Pose Estimation

Nuno Pereira
Universidade da Beira Interior, NOVA LINCS
Rua Marquês d'Ávila e Bolama, 6201-001,
Covilhã, Portugal
Email: nuno.pereira@ubi.pt

Luís A. Alexandre
Universidade da Beira Interior, NOVA LINCS
Rua Marquês d'Ávila e Bolama, 6201-001,
Covilhã, Portugal
Email: luis.alexandre@ubi.pt

***Abstract*—6D pose estimation is an open challenge due to complex world objects and many possible problems when capturing data from the real world, *e.g.*, occlusions, truncations, and noise in the data. Achieving accurate 6D poses will improve results in other open problems like robot grasping or positioning objects in augmented reality. MaskedFusion is one of the most accurate methods for 6D pose estimation but before estimating the pose, the object needs to be detected and segmented. One of the most important stages in the MaskedFusion 6D pose pipeline is image segmentation because, with good image segmentation, it is possible to discard the background or other non-relevant data that are around the object leaving only the data that are most relevant to the 6D pose estimation. We study the impact of using different image segmentation methods in the MaskedFusion 6D object pose estimation and we also study the impact of the color spaces in the MaskedFusion and DenseFusion methods. The experiments conducted, show how robust MaskedFusion is and that using some filtering operations after the predicted masks improves the accuracy of the method. We also show that, with one of the semantic segmentation methods tested, we achieve on average 97% accuracy on the LineMOD dataset, only 0.2% worst than the baseline that uses the ground truth masks provided by the dataset. With the modifications of the color spaces, we improved MaskedFusion in 1.1% and DenseFusion 0.3% in the LineMOD dataset and reach also 0.3% improvement for the MaskedFusion and 0.4% for the DenseFusion in the YCB-Video dataset.**

I. Introduction

Image segmentation aims to extract meaningful information from images so it facilitates further analysis. Our focus is to evaluate the impact of different quality masks obtained from image segmentation methods in pipelines like MaskedFusion [1] and to evaluate the impact of different color spaces in the methods MaskedFusion [1] and DenseFusion [2]. These methods try to solve the 6D pose estimation problem [3], [4], [5]. Inside of the image segmentation methods, we will focus on deep learning methods that can classify each pixel of an RGB image. This type of method is named semantic segmentation.

Most methods for semantic segmentation require that each pixel has a label associated with it such that it is possible to predict a label for every pixel of the image. Not only the class but also the boundaries of each object matter to the prediction. The output prediction also reflects the spatial-relationship of all objects present in the image.

In the color spaces study, we will focus on three main color spaces, RGB, HSV, and Gray. In computer vision, ambient light can be a notable problem. It can create artifacts, alter the colors or cause shadows in the captured scene, therefore, constituting a problem in many computer vision algorithms. These problems can affect the performance of 6D pose estimation methods.

The RGB color space is widely used, although it does not represent the color as humans perceive it. If we want to isolate an object just using color, it is hard to do it because there may be many similar colors in the image.

The HSV color space has three channels, as does RGB, but instead of Red, Green, and Blue we have Hue, Saturation, and Value, or intensity. The Hue channel represents the color. For example, red is a color but light red or dark red is not. The saturation channel is the amount of color present. It differentiates pale red from pure red. Finally, the value or intensity represents the brightness of the color, light red or dark red. So in the Hue channel, each color has its own value, all different reds are mapped into a unique value. The lightness or darkness of the color does not affect the hue channel, so this channel is useful to extract specific colors from images. In real photographs, you will obtain varied saturation throughout the images depending on the intensity of the color present in them. The intensity channel shows the brightness of the colors and this channel, is usually highly affected by the light source.

Multiple areas, like autonomous driving, robotics, image search engines, human-machine interactions, object detection, and more specific 6D pose estimation that is the focus of this paper, use semantic segmentation to aid in the solution of the problems.

6D pose estimation is an open problem because there is no satisfactory solution for it under all possible circumstances. 6D pose estimation can be used in several tasks like grasping, robotic manipulation, augmented reality, and others. It is as important in robotic tasks as in augmented reality, where the pose of real objects can affect the interpretation of the scene and the pose of virtual objects can also change the augmented reality experience. It can also be useful in human-robot interaction tasks such as learning from demonstration and human-robot collaboration.

Estimating an object's 6D pose is a challenging problem due to the geometric variability of objects and how they appear in the real world. Captured scenes from the real world might have occlusions and truncations on some objects. Obtaining the data to retrieve the 6D pose is also a problem, as RGB-D data can be hard to obtain for certain types of objects, *e.g.*, fully metallic objects and meshed objects such as office garbage bins. Some ambients also generate noise or interference in the captured data and this can lead to errors because, if the data captured has issues some methods will not work or will predict wrong poses.

There were a lot of computer vision techniques for Semantic Segmentation before the appearance of deep learning. All of them relied on hand-engineered features to classify each pixel independently. After the appearance of deep learning, the semantic segmentation methods improved substantially. Especially after convolutional neural networks emerged and showed their advantages in the Imagenet competition. Convolutional neural networks improved the state-of-the-art of semantic segmentation by increasing the accuracy of the predictions through the creation of hierarchies of representations.

We tested the robustness of MaskedFusion with respect to different approaches for semantic segmentation. We show that MaskedFusion can adapt to different quality masks, by just training it again with lower quality ones.

With the knowledge gained from these experiments, we improve the pipeline of MaskedFusion by using better image segmentation methods that produce better quality masks and are almost able to achieve the 6D pose estimation accuracy of ground truth masks.

With the color spaces experiments conducted, we found which color space can improve the pose estimation, under which conditions. The HSV color space improved the overall accuracies for data with color and low texture.

II. 6D Pose Methods

Our focus was on methods that estimate the 6D pose of an object from RGB-D data.

MaskedFusion [1] is a pipeline divided into 3 sub-tasks that combined can solve the task of object 6D pose estimation. MaskedFusion is a modular pipeline, for each sub-task a neural network is used to solve it. However, since the pipeline is modular, every sub-task can have different types of methods that will solve the task at hand and can be replaced easily.

In the first sub-task, the detection and segmentation for each object in the scene occurs. For that, a neural network based on the encoder-decoder architecture is used. It classifies each pixel of the RGB image captured and predicts the mask and the location for each object in the scene. After the predicted mask, a median and a dilate filter are used on the mask. After the filters applied to the mask, *bit-wise and* operations are used on the original RGB and depth images. With this technique, each object is segmented from the original data and these resultant images are cropped within the object boundaries and are ready for the next sub-task. In the second sub-task, with the masks obtained in sub-task 1 for each object and the RGB-D data, it is possible to estimate the object 6D pose. For each type of input data, the method has different neural networks to extract features. After all the features are extracted they are combined and then another neural network is used to extract the most meaningful features and then regress the estimated 6D pose of the object. After this sub-task, it is possible to obtain the rotation matrix and the translation vector according to the 6D pose estimated from the 6D pose neural network, but it is also possible to feed this out to the next sub-task named pose refinement was a neural network can refine the 6D pose of the object by approximating the values estimated with the ground truth one provided. This last sub-task is optional but advised by the authors because it can improve the accuracy of the method.

DenseFusion [2] is a method that, from an image with detected objects, can crop these objects and estimate their 6D poses. It has 2 sub-tasks similar to the two final tasks of MaskedFusion, where from the input data extract features from the RGB image and Depth and estimate the 6D pose of the cropped object. In their first sub-task, they estimate the preliminary 6D pose of the object, and then it is refined in their second sub-task. The refinement sub-task is the same as in MaskedFusion since the authors of MaskedFusion used the same neural network for the refinement as DenseFusion.

III. Impact of the Semantic Segmentation

In this section, we present all the experiments conducted and how we improved methods like MaskedFusion to achieve less error overall.

All the experiments presented in this document were executed on a desktop computer with SSD NVME, 64GB of RAM, an NVIDIA GeForce GTX 1080 Ti, and Intel Core i7-7700K CPU.

A. Experimental Setup

We did two main experiments to evaluate two hypotheses. The first consists of the belief that the use of post-processing operations over the predicted masks could improve the accuracy in the estimation of the 6D pose of an object. The second hypothesis regards the robustness of MaskedFusion: we hypothesize that it is robust enough to deal with lower quality masks that can be produced when dealing with real-world data.

1) Post-processing Operations over the Masks: To evaluate if the first hypothesis was correct we created the first experiment that consisted of the following flow: after training the MaskedFusion method with the ground truth masks, we also trained each semantic segmentation neural network on the same dataset. Then we feed-forward the predicted masks with and without post-processing on the test subset of the dataset from each semantic segmentation method to the pre-trained MaskedFusion 6D pose and refinement network.

2) Impact of low Quality Masks: For the second experiment, our main goal was to show how robust MaskedFusion is when using lower quality masks and how much it could improve if trained using predicted masks instead of the ground truth masks. The flow of our experiment is: training each of

TABLE I

Quantitative evaluation of 6D pose using the ADD metric on the LineMOD dataset. Symmetric objects are presented in italic and were evaluated using ADD-S. The presented values were obtained using the dataset masks to train the MaskedFusion and the evaluation used the masks generated by the semantic segmentation methods. Bold shows best results in a given row.

Objects	GT Mask	SegNet	SegNet w/ Operations	Deeplabv3 (ResNet101)	Deeplabv3 (ResNet101) w/ Operations	FCN (ResNet101)	FCN (ResNet101) w/ Operations
ape	**89.5**	81.0	80.0	84.6	87.5	82.9	86.7
bench vi.	98.1	94.2	95.1	98.1	98.1	**99.0**	98.1
camera	99.0	99.0	99.0	**100.0**	**100.0**	99.0	99.0
can	96.0	88.1	87.1	97.0	**98.0**	95.0	95.0
cat	**100.0**	97.0	97.0	**100.0**	**100.0**	**100.0**	99.0
driller	**97.0**	90.0	91.0	95.0	96.0	94.0	95.0
duck	**94.3**	87.7	90.6	91.4	93.3	88.7	90.6
eggbox	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	99.1
glue	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
hole p.	**98.1**	95.2	95.2	96.2	95.2	96.2	96.2
iron	**97.9**	96.9	95.9	95.9	96.9	96.9	96.9
lamp	**99.0**	97.1	96.2	97.1	96.2	98.1	**99.0**
phone	94.2	94.2	94.2	95.2	95.2	**98.1**	**98.1**
Average	**97.2**	93.9	93.9	96.2	96.6	96.0	96.3

TABLE II

Evaluation of the pre-trained weights on COCO val2017. Bold values are the best values for each metric.

Model (Backbone)	Mean IOU	Global Pixelwise Accuracy
SegNet	63.2	91.3
FCN (ResNet101)	63.7	91.9
DeepLab v3 (ResNet101)	**67.4**	**92.4**

the semantic segmentation neural networks, we generate masks from each method for all subsets (train, validation, and test) of the dataset. Then these predicted masks were used to train the MaskedFusion 6D pose and refinement networks. With this experiment, we can see the adaptation of the MaskedFusion to different quality masks.

B. Dataset

The most used dataset to tackle the 6D pose estimation problem is LineMOD[6]. It has 15 low-textured objects but many methods only use 13 objects. These 15 objects are present in over 18000 images and all of them are annotated. This dataset was captured with a Kinect camera, that automatically aligned the RGB and depth images. The dataset consists of RGB-D data and has masks for each object. The 3D model of each object is also available in the dataset with the corresponding maximum diameter. LineMOD is considered a dataset with mild occlusion with some objects partial covering others. This dataset was captured in a light controlled environment.

C. Evaluation Metrics

To evaluate the impact of the image segmentation on MaskedFusion we used three different neural networks to segment the objects presented in the scene to crop each object and then estimate its 6D pose. We made two different experiments for each method and we also studied the impact of the post-processing operations that were made to the mask after its prediction presented by the MaskedFusion [1] authors.

As in previous methods [2], [3], [4], [5] that tackle the 6D pose estimation using the LineMOD dataset, we use the Average Distance of Model Points (ADD) [6] as an evaluation metric for non-symmetric objects, and for the egg-box and glue (symmetric objects) we use the Average Closest Point Distance (ADD-S) [5].

In the ADD metric (1), the mean of the pairwise distances between the 3D model points of the ground truth pose and the estimated pose is calculated. Assuming the ground truth rotation R and translation t and the estimated rotation $\tilde{R}$ and translation $\tilde{t}$. In equations (1) and (2) M represents the set of 3D model points and m is the number of points.

$$\text{ADD} = \frac{1}{m} \sum_{x \in M} \left\| (Rx + t) - (\hat{R}x + \hat{t}) \right\| \qquad (1)$$

For the symmetric objects (egg-box and glue), the matching between points is ambiguous for some poses. In these cases we used the ADD-S metric:

$$\text{ADD-S} = \frac{1}{m} \sum_{x_1 \in M} \min_{x_2 \in M} \left\| (Rx_1 + t) - (\hat{R}x_2 + \hat{t}) \right\| \qquad (2)$$

D. Results

In this section, we present the results of the experiments and analyze the influence of different semantic segmentation methods on the 6D pose estimation task.

1) Semantic Segmentation: We used three neural networks to tackle the task of semantic segmentation: SegNet [7], FCN-ResNet101 [8] and DeepLab v3 [9]. For SegNet we used a standard implementation, for the FCN and DeepLab v3 we used ResNet101 as backbone. All of the networks used were

TABLE III
Quantitative evaluation of 6D pose using the ADD metric on the LineMOD dataset. Symmetric objects are presented in italic and were evaluated using ADD-S. The presented values were obtained using the masks generated by the semantic segmentation methods to train and evaluate the MaskedFusion. Bold shows best results in a given row.

Objects	GT Mask	SegNet	SegNet w/ Operations	Deeplabv3 (ResNet101)	Deeplabv3 (ResNet101) w/ Operations	FCN (ResNet101)	FCN (ResNet101) w/ Operations
ape	**89.5**	82.0	81.0	85.6	88.5	83.9	87.7
bench vi.	98.1	94.2	95.1	98.1	98.1	**99.0**	98.1
camera	99.0	99.0	99.0	**100.0**	**100.0**	99.0	99.0
can	96.0	89.1	88.1	98.0	**99.0**	96.0	96.0
cat	**100.0**	97.0	97.0	**100.0**	**100.0**	**100.0**	99.0
driller	**97.0**	90.5	91.5	95.5	96.5	94.5	95.5
duck	**94.3**	88.2	91.1	91.9	93.8	89.2	91.1
eggbox	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	99.1
glue	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
hole p.	**98.1**	95.7	95.7	96.7	95.7	96.7	96.7
iron	**97.9**	97.4	96.4	96.4	97.4	97.4	97.4
lamp	**99.0**	97.1	96.2	97.1	96.2	98.1	**99.0**
phone	94.2	95.2	95.2	96.2	96.2	**99.1**	**99.1**
Average	**97.2**	94.3	94.3	96.6	97.0	96.4	96.7

pre-trained on COCO [10] train2017 dataset and then fine-tuned for the LineMOD dataset.

In Table II we present the evaluation results on COCO val2017 for Mean IOU and Global Pixel-wise Accuracy for the three models.

2) 6D Pose Estimation: In the first experiment with MaskedFusion, we trained it in the LineMOD dataset using the ground truth masks to achieve the best possible results, since in the real world it would be very difficult to obtain these precise masks. The baseline results for MaskedFusion using the ground truth masks present in the dataset are shown in the second column of Table I. With the weights obtained from MaskedFusion's 200 training epochs, we can evaluate the influence that image segmentation has, for the three segmentation methods.

We trained each semantic segmentation method for 50 epochs and saved the weights of the best accuracy run of the validation subset of LineMOD. For the test subset of LineMOD, we predict the mask of the input images and then we create new binary images with the predicted masks that can be used as input for MaskedFusion 6D pose estimation network during its evaluation. During the evaluation of the MaskedFusion in the test subset of the LineMOD, we used the RGB-D data provided by the dataset but, for the masks used to crop the objects and to extract features from the shape of the object, we use the predicted masks.

On the top row of Table I we have the ground truth mask and the three methods tested, each with and without the filtering operations proposed by the MaskedFusion authors. The obtained results show that the proposed filtering operation can increase slightly the accuracy of the model compared with its counterpart. The MaskedFusion using DeepLab v3 masks with the filter operations median and dilate applied to its masks achieved 96.6% of average accuracy, only 0.6% below the baseline that used ground truth masks.

For the second experiment, we trained each semantic segmentation method for 50 epochs and saved the weights of the best accuracy run of the validation subset of LineMOD. With the best accuracy weights, we predicted masks for all the subsets (train, validation, test) of the LineMOD. With the predicted masks for each method, we trained MaskedFusion to check if it could adapt itself to lower quality masks. We did the same process for the methods that had the masks with the filtering operations as post-process. Table III contains the results.

In this experiment we trained MaskedFusion for 200 epochs for each method since here the input data were changed by the different previous segmentation methods. The presented results show that the use of filtering operations after the prediction of the mask improves the accuracy of the estimated pose, and we showed that MaskedFusion can adapt itself to lower quality masks since in this experiment all the masks used were provided by semantic segmentation methods. The use of the masks produced by the method DeepLab v3 achieved again the best average accuracy for this problem resulting in an accuracy 0.2% below the ground truth baseline.

IV. Impact of the Color Space

A. Experimental Setup

In our experiments, we used DenseFusion [2] and MaskedFusion [1], but we did not use the first sub-task (semantic segmentation) of the MaskedFusion. Our primary goal is to report the impact of the different color spaces and/or channels in the 6D pose estimation. Since MaskedFusion is a modular framework, it was effortless to remove the semantic segmentation sub-task and use the ground truth masks to make the operations for the detection, crop, and background removal. This also enables us to have a direct comparison between

TABLE IV

Quantitative evaluation of 6D pose using the ADD metric on the LineMOD dataset. Symmetric objects are presented in italic and were evaluated using ADD-S. Bold numbers are the best in a row for MaskedFusion and underline bold numbers are the best in a row for DenseFusion both methods were trained for 200 epochs.

	MaskedFusion [1]						DenseFusion [2]					
Objects	RGB	HSV	Gray	H	S	V	RGB	HSV	Gray	H	S	V
ape	89.5	**97.1**	86.7	67.6	34.3	82.9	92.3	**92.8**	85.8	70.4	37.1	85.6
bench vi.	98.1	99.0	**100.0**	88.3	89.3	99.0	93.2	93.9	90.2	83.5	84.5	**94.2**
camera	**99.0**	98.0	98.0	87.3	75.5	97.1	94.4	**95.8**	92.4	82.6	70.9	92.4
can	96.0	**98.0**	97.0	80.2	91.1	93.1	93.1	**93.7**	92.1	77.3	88.1	90.1
cat	**100.0**	97.0	95.0	81.0	86.0	97.0	**96.5**	96.4	94.6	77.5	82.5	93.5
driller	97.0	**99.0**	95.0	91.0	88.0	94.0	87.0	**87.4**	85.0	81.0	78.0	84.0
duck	94.3	**96.2**	93.4	51.9	35.8	88.7	**92.3**	92.0	90.1	49.8	33.8	86.6
eggbox	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**99.8**	**99.8**	95.8	**99.8**	**99.8**	**99.8**
glue	**100.0**	**100.0**	99.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	98.1	**100.0**	**100.0**	**100.0**
hole p.	98.1	**99.0**	95.2	89.5	74.3	**99.0**	92.1	92.4	89.8	83.5	68.3	**93.1**
iron	97.9	95.9	97.9	94.8	91.8	**99.0**	97.0	97.5	95.4	93.9	90.8	**98.0**
lamp	99.0	98.1	**100.0**	95.2	97.1	**100.0**	95.3	95.8	91.8	91.5	93.4	**96.3**
phone	94.2	**100.0**	99.0	93.3	94.2	99.0	92.8	92.7	89.4	91.8	92.8	**97.6**
Average	97.2	**98.3**	96.6	86.2	81.3	96.1	94.3	**94.6**	91.6	83.3	78.5	93.2

DenseFusion and MaskedFusion and know the possible improvements that can be achieved in both methods.

To perform our tests, we choose to compare the HSV color space and each of its channels with the RGB color space. We tested MaskedFusion using the RGB, HSV, Grayscale, H (Hue), S (Saturation), and V (Value). We evaluated the DenseFusion [2] and MaskedFusion [1] training both from randomly initialized weights for 200 epochs in the LineMOD dataset for all the mentioned color spaces and channels and 100 epochs in the YCB-Video dataset.

B. Datasets

In our experiments, we use the LineMOD [6] dataset and YCB-Video [11] dataset because they are widely utilized in this area of research. The YCB-Video dataset has 21 objects, these objects have different shapes and textures, and there are mild occlusions presented in the captured data. It is composed of 92 RGB-D videos, each with a subset of the objects placed in the scene. We use the dataset in the same splits has previous works, [1], [2], [5] where 80 videos were used for training and 12 for testing. LineMOD [6] dataset is specified in the previous section III-B. These datasets were captured in a light-controlled environment.

As in previous works in 6D pose estimation [1], [2], [3], [4], [5] we use the same evaluation metrics for the LineMOD dataset. We use the Average Distance of Model Points (ADD) for non-symmetric objects and for symmetric objects the Average Closest Point Distance (ADD-S) [5] is used. These evaluation metrics are described in section III-C. For the YCB-Video we present the area under the ADD-S 2 curve as in DenseFusion [2] and MaskedFusion [1].

C. Results

In Table IV we present the quantitative evaluation for the LineMOD dataset of the two methods that we used to estimate the object's 6D pose. On average both methods had less pose error using the HSV color space. Since LineMOD is a dataset where the objects have less texture and is colorful, using a different color space as HSV, improved slightly the accuracy.

The YCB-Video dataset quantitative evaluation is presented in Table V comparing each color space for both methods used in this experience. In the YCB-Video dataset, the objects have more texture and they are less colorful thus not enabling the HSV to have a big advantage over the other color spaces. In the MaskedFusion experiments, we obtained the same average accuracy in the HSV and Gray color space. The color space gray has higher scores in objects that have more texture or objects that are black, comparing to the other color spaces.

During inference, we took an average of 0.014 seconds to estimate the 6D pose of an object. Our experiments took on average more 0.002 seconds to estimate the 6D pose of an object comparing its execution time with the RGB color space that did not need any color space conversion. These times were obtained using the computer described above.

V. Conclusion

In this paper, we presented multiple experiments that were done to further analyze two important factors that influence the 6D pose estimation problem: the quality of segmentation masks for MaskedFusion and the color space used for representing the captured images for MaskedFusion and DenseFusion. We concluded that, for the two 6D pose estimation methods that we compared, using different color spaces can improve the accuracy of the estimated pose. For MaskedFusion, that uses semantic segmentation masks of the objects to remove the background and information not related with the object, using state-of-the-art methods in the semantic segmentation with post-processing filters to smooth the obtained masks can improve the accuracy of the 6D pose estimated.

TABLE V

QUANTITATIVE EVALUATION OF 6D POSE (AREA UNDER THE ADD-S CURVE(AUC)) ON THE YCB-VIDEO DATASET. BOLD NUMBERS ARE THE BEST IN A ROW FOR MASKEDFUSION AND UNDERLINE BOLD NUMBERS ARE THE BEST IN A ROW FOR DENSEFUSION BOTH METHODS WERE TRAINED FOR 100 EPOCHS.

	MaskedFusion [1]						DenseFusion [2]					
Objects	RGB	HSV	Gray	H	S	V	RGB	HSV	Gray	H	S	V
002_master_chef_can	95.9	**96.2**	95.9	95.6	95.6	96.1	94.3	**94.6**	94.3	94.0	94.0	94.5
003_cracker_box	96.0	96.5	96.2	95.6	**96.6**	95.3	94.0	94.5	94.2	93.5	**94.6**	93.3
004_sugar_box	97.6	97.8	97.7	97.6	**97.9**	97.7	95.7	95.9	95.6	95.7	**95.9**	95.8
005_tomato_soup_can	94.2	**94.3**	94.3	93.8	94.1	94.1	90.3	**90.5**	90.4	89.9	90.2	90.1
006_mustard_bottle	97.6	97.9	96.9	**98.1**	97.0	96.3	95.2	95.5	95.0	**95.6**	94.5	93.9
007_tuna_fish_can	96.7	97.0	**97.1**	96.7	97.0	96.5	95.4	95.5	95.4	95.4	**95.7**	95.3
008_pudding_box	96.3	96.8	96.5	94.6	97.0	**97.2**	95.1	95.9	94.9	93.4	95.9	**96.0**
009_gelatin_box	98.0	97.9	**98.2**	98.1	98.2	97.5	97.2	97.6	**97.7**	97.3	97.4	96.7
010_potted_meat_can	89.4	89.4	**89.5**	88.8	88.9	89.0	88.1	**88.3**	**88.3**	87.5	87.5	87.7
011_banana	**97.5**	**97.5**	97.3	**97.5**	97.2	96.4	95.0	**95.3**	94.8	95.0	94.7	93.9
019_pitcher_base	97.4	97.7	98.1	97.9	97.7	97.6	96.1	96.4	96.4	**96.6**	96.4	96.2
021_bleach_cleanser	93.8	**96.4**	95.1	93.8	95.0	94.0	94.6	94.9	94.5	94.6	**95.8**	94.8
024_bowl	**90.1**	87.8	90.0	90.0	88.6	88.0	88.4	**89.1**	88.0	88.2	86.8	86.2
025_mug	97.0	97.3	97.3	**97.6**	97.5	97.2	95.8	95.8	93.2	**96.3**	96.3	96.0
035_power_drill	96.4	**96.9**	96.5	95.9	94.5	96.1	93.9	**94.2**	93.8	93.3	92.0	93.6
036_wood_block	90.6	91.7	**92.4**	91.7	89.7	91.2	90.2	90.9	90.7	**91.3**	89.2	90.8
037_scissors	**93.2**	91.1	91.0	92.0	90.9	92.4	92.4	**93.0**	92.3	91.2	90.1	91.6
040_large_marker	97.0	97.3	**97.5**	96.6	97.2	96.9	95.7	95.7	95.8	95.3	**95.9**	95.6
051_large_clamp	72.1	72.0	**72.3**	72.0	71.3	71.8	69.7	70.2	**70.3**	69.6	68.9	69.4
052_extra_large_clamp	69.6	**72.2**	**72.2**	71.0	71.5	72.1	64.5	65.0	65.1	65.9	66.5	**67.0**
061_foam_brick	94.2	**95.2**	94.6	93.7	92.5	94.5	92.0	**92.7**	92.1	91.5	90.3	92.3
Average	92.9	**93.2**	**93.2**	92.8	92.7	92.8	91.1	**91.5**	91.1	91.0	90.9	91.0

In terms of using different color spaces for the 6D pose estimation, we learned that if we are estimating poses of colorful objects, the HSV color space could improve these methods because the color features might be more relevant to the neural networks. For objects with significant textures, for example, objects packed in boxes with labels and drawings in them, the HSV color space did not perform as well as in other cases, and we concluded that for these types of objects we should use the Gray or the RGB color space.

Overall, using the HSV color space, we improved MaskedFusion in 1.1% and DenseFusion 0.3% in the LineMOD dataset and 0.3% for the MaskedFusion and 0.4% for the DenseFusion in the YCB-Video dataset. These are small improvements but these methods already have high accuracies in these datasets, so it is very difficult to obtain large significant gains.

ACKNOWLEDGMENT

This work was supported by NOVA LINCS (UIDB/04516/2020) with the financial support of FCT-Fundação para a Ciência e a Tecnologia, through national funds, and partially supported by project 026653 (POCI-01-0247-FEDER-026653) INDTECH 4.0 – New technologies for smart manufacturing, cofinanced by the Portugal 2020 Program (PT 2020), Compete 2020 Program and the European Union through the European Regional Development Fund (ERDF).

REFERENCES

[1] N. Pereira and L. A. Alexandre, "MaskedFusion: Mask-based 6d object pose estimation," in *19th IEEE International Conference on Machine Learning and Applications (ICMLA 2020)*, December 2020.

[2] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," in *CVPR*, 2019, pp. 3343–3352.

[3] 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 *ICCV*, 2017, pp. 1521–1529.

[4] 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.

[5] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[6] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal templates for real-time detection of textureless objects in heavily cluttered scenes," in *2011 international conference on computer vision*. IEEE, 2011, pp. 858–865.

[7] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[8] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, p. 3431–3440.

[9] L.-C. Chen and Papandreou, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*. Springer, 2014, pp. 740–755.

[11] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *ICAR 2015*. IEEE, 2015, pp. 510–517.