Door Recognition and Handle Detection Using Deep Convolutional Neural Nets in Indoor Environments

Adrian Llopart[1] and Nils A. Andersen[1] and Ole Ravn[1]

***Abstract*— In this paper, we present a new method for robustly identifying and detecting doors and cabinets, with special emphasis on extracting useful features from handles for future robotic manipulation. The novelty of this approach relies on the usage of a Convolutional Neural Net (CNN) as a form of reducing the search space instead of the traditional methods, based on visual geometric features or depth properties. The framework consists of the following components: The implementation of a CNN to extract a Region of Interest (ROI) from an image corresponding to a door or cabinet, as proposed by [13]. Several vision based techniques to detect handles inside the ROI and its 3D positioning. A complementary plane segmentation method to differentiate door/cabinet from the handle. An algorithm to fuse both approaches robustly and extract essential information from the handle for robotic grasping (i.e. handle point cloud, door plane model, grasping locations, turning orientation, orthogonal vector to door). The system assumes no prior knowledge of the environment (unlike [1] and [2]). The algorithm can be easily modified to accommodate several types of handles, but those with cylindrical form and horizontal steady state orientation will be specially addressed.**

I. INTRODUCTION

With the technological improvements of the last years, mobile robots have become greatly autonomous, capable of fulfilling diverse services and tasks without human intervention. Despite this, the truth is that mobile robotics are far from having the autonomy required for them to explore correctly human-made environments.

A very clear example of this is the necessity of human intervention to open closed doors when robots move in an indoors environment. A wide variety of methods have been explored to solve this situation; namely detection of doors and navigating through them. These techniques incorporate the usage of either lasers scans, images ([3], [4], [5], [6] and [7]) or depth data ([8], [9], [10] and [11]) to solve the door detection problem. Whilst it is true that some of them accomplish significantly positive results, there is an inherent loss of conceptual information, important limitations for each approach and, more significantly, a loss of processing speed due to all the information required. With the development of pattern recognition and artificial intelligence techniques, novel algorithms are proposed that, in a manner, replicate the learning process and adaptability of human beings whilst, simultaneously, reducing computational time. [12] directly detects handles using a 2D sliding window classifier but is unable to recognize any other object. This is precisely what this paper presents: a novel method to not only enhance the autonomy of robots when facing the problem of door detection, but also add adaptability to unknown and dynamically changing scenarios in real time.

[1]All authors are with Department of Electrical Engineering, Technical University of Denmark, Elektrovej 326, 2800, Lyngby, Denmark adllo@elektro.dtu.dk

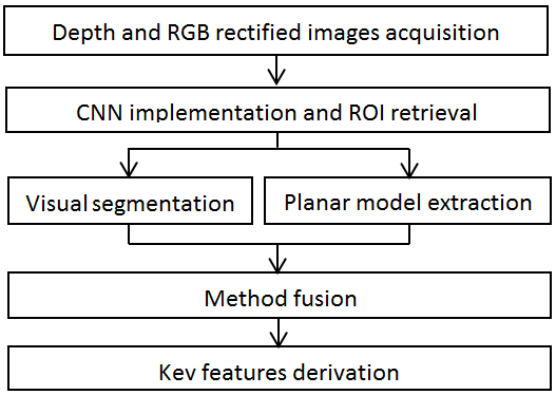


Fig. 1: Flowchart of the proposed method

The general aim of this approach is to replicate the human methodology for opening doors. Initially, the human will search for doors or cabinets because these objects have bigger surfaces and allow for a faster and better recognition. Then, the human will instinctively find the handle (following certain assumptions, like height and difference in color between door surface and handle). Finally, the handle position will be estimated and grasped.

To achieve this, rectified RGB and Depth images will be procured from a Kinect sensor. A Convolutional Neural Net, previously trained over several hundred door and cabinet images from *ImageNet*, will take the rectified RGB image as input and generate bounding boxes around doors and cabinets. Two methods will then be used to obtain the handles point cloud: The first one is a visual segmentation approach based on k-means color clusterization of the region of interest. The second one is a plane model extraction of the point cloud generated inside the ROI. Finally, the results from both techniques will be merged and a final estimate of the handles point cloud will be produced, from which essential key features for robotic manipulation will be derived. The entire process is shown in Fig. 1.

II. CONVOLUTIONAL NEURAL NET

The novelty and strength of the presented method in this paper, similarly to [13], resides in the use of CNN as a preliminary approach to recognize, detect and segment a ROI out of a full image. This will increase responsiveness

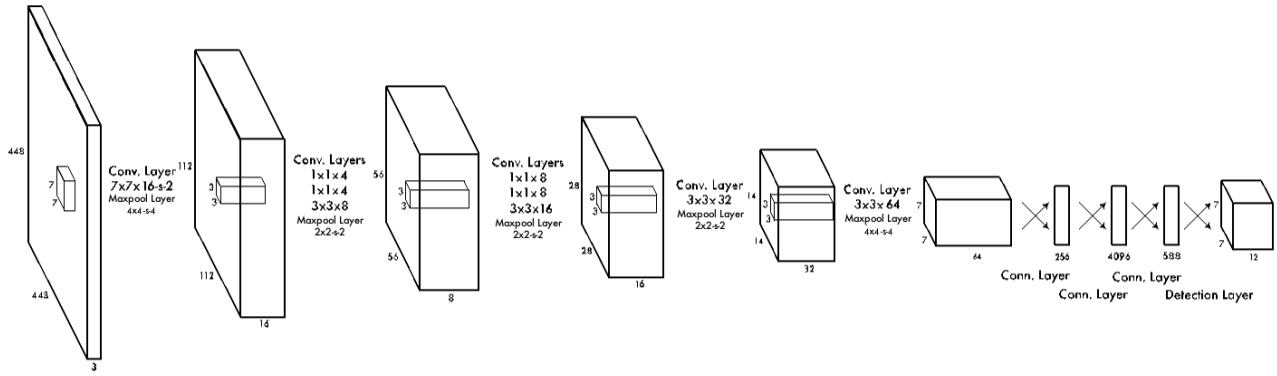
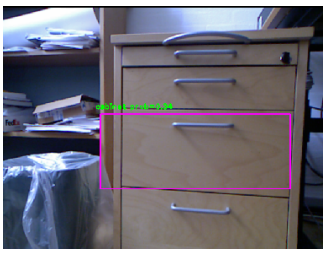


Fig. 2: Architecture of the Convolutional Neural Net. The system models detection as a regression problem to a 7x7x12 tensor. This tensor encodes bounding boxes and class probabilities for all objects in the image.

and performance since it significantly reduces the amount of data that needs to be processed in later steps of this method. To achieve this, the proposed CNN must be extremely fast in its entire process, optimized and reliable, generating correct bounding boxes around those objects it has been trained to detect with a high level of precision. Hence, a unified architecture that predicts bounding boxes and class probabilities directly from full images is required: The YOLO Detection System [14]. The whole detection pipeline is a single network that divides the image into regions and predicts bounding boxes and probabilities for each region. Thus, it can be optimized end-to-end directly on detection performance resulting in an extremely fast runtime. Since real time processing of video streams with state-of-the-art performance is of utmost importance, a small neural net model is applied in the proposed methodology: the *Darknet Reference Model*. The speed of execution of the neural net is directly proportional to the GPU utilized. Redmon *et al.* [14] claim to reach up to 150 fps on a Titan X with this model and 20 classes.

A training is done that focuses specifically on two classes: doors and cabinets (which includes also drawers and lockers). The images are randomly selected from *ImageNet*, with a total of 510 images for the doors, and 420 for the cabinets. The usage of random training images with different lighting conditions, rotations and scale allows for a trained neural net that is able to predict general properties instead of overfitting, gaining great robustness in front of transformation, deformation and illumination changes. The whole end-to-end detection process operates at around 14 FPS. Some results of the bounding boxes found by the CNN can de seen in Fig. 3.

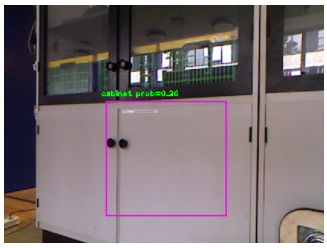
Lastly, a great advantage of using a CNN at the beginning of the process is that the system can be expanded by training it to detect other objects and interact with them accordingly, without affecting the performance of door and handle detection presented in this paper.



(a) Cabinet n°1

(b) Cabinet n°2

(c) Red door



(d) Cabinet n°3

Fig. 3: Resulting bounding boxes for several cabinets and doors. The resulting door and handle point clouds are depicted in Fig. 7

III. Handle's point cloud generation

As mentioned above, the execution of a trained neural net allows for the selection of a ROI around the identified object. All information outside the bounding box is neglected from further processing allowing for faster and more precise results. Two approaches will be considered to derive the final handle point cloud: visual segmentation and planar model extraction.

A. ***Method 1***. *Visual segmentation*

Two assumptions are taken into consideration when visually detecting handles and generating the corresponding point cloud. The former, the region of interest given by the CNN correctly includes only the desired object (door/cabinet surface). The latter, door and cabinet surfaces have a clear color contrast with their handles.

If both of these premises are fulfilled, a K-means color clustering can be applied to the ROI. With this method, the image is partitioned into k clusters (in this case specifically, and given the second assumption, $k = 2$).

The implementation of this method to the provided ROI, outputs an image containing only two colors; ideally the one corresponding to the surface of the door and the one of the handle. In reality, the color segmentation, even though close to the exact colors, is sometimes slightly offset. However, the goal is not to extract the precise colors but rather to create a clear differentiation between door plane and handle, from which a mask can be extracted. One of the main false positives this method provides is the detection of the lock beneath the handle.

To limit the detection specifically to handles, a Canny Edge Detector is applied to the clustered image to obtain only the contours. The detected contours are evaluated in size, form and orientation to see if they match the desired handles or are, simply, unnecessary elements: A size threshold is applied which allows for the deletion of contours that are too small (mostly due to lighting conditions that may have affected the k-means color clusterization step) or too big (if the ROI is not precise enough, some wall sections might be included in it). Since this paper focuses specifically on door levers with a cylindrical horizontal form, a rotated rectangle is approximated over the area of all the remaining contours. If said rectangle can be overlapped and its angle is close to the horizontal (the angle is derived from the longest axis of the rectangle) then the contour is kept. If not, it is removed.

The outcome of the ROI extraction, k-means color clustering and contour segmentation is a binary mask which robustly showcases the handle. The whole process is depicted in Fig. 4.

To obtain the handles point cloud, the rectified depth image is required. The usage of the Kinect sensor and the ROS firmware allows for the procurement of a rectified depth image thanks to the disparity image. This grayscale image displays the distance from every pixel to the object it detects in the real world. It has been transformed and projected into the same plane as the rectified RGB image (used previously on the Neural Net and the visual segmentation) and is sufficient and necessary to generate a point cloud.

The resulting binary mask from the visual methodology is applied to the rectified depth image. Since both images have been rectified to match each other, this operation is allowed as every element of one image corresponds to the same element of the other image. Considering the camera has been previously calibrated, its intrinsec parameters are known. Thus, a point cloud can be generated. The derived point cloud will display the door handles 3D information precisely but might incorporate some impurities which are partly removed by cropping the point cloud around the range of 80 to 120 cm in height (because that is where handles usually are). Even though it is uncommon, some other elements might have survived the process. These will be removed in a posterior process when the handle point cloud generated by the described visual segmentation method and the one generated from plane extraction are merged together. This is explained in section III-C.

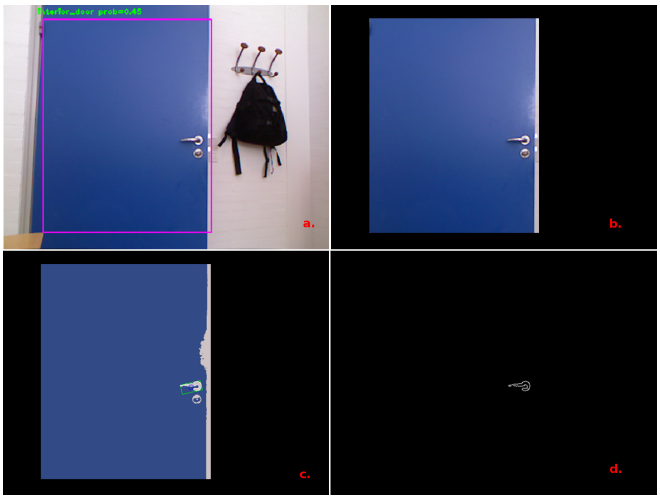


Fig. 4: **Method 1: Visual segmentation**. **a.** ROI selection from the output of the CNN (with class name and probability written on top). **b.** Cropped rectified RGB image. **c.** K-means clustering of ROI with rotated rectangles (green) and centers (blue) drawn over those contours that fit into certain specifications. **d.** Final binary mask showing only the contour corresponding to the handle, to be applied later to the depth image to generate the handles point cloud.

B. ***Method 2.*** *Planar model extraction*

As seen in the previous section, a point cloud can be easily generated given a depth image. In the preceding method, the door handle was extracted through segmentation in the RGB space to create a mask which was then applied to the depth information to obtain the point cloud. In this new method however, the full point cloud of the environment is generated from the original ROI, given by the neural net. Then, a plane model segmentation is applied to it.

This method focuses primary on point cloud manipulation via the *PCL* library as a secondary way of extracting the handles point cloud. To apply the following planar extraction, the full environments point cloud could be used, as done in [15], but that would use more information than necessary and definitely slower the processing time. Hence, to allow real time processing, the point cloud needs to be down-sampled. Applying a voxelized grid approach would result in a lower resolution of all areas (including the door handle) which is the opposite of what is intended. To solve this issue, the initial binary mask from the ROI given by the CNN is applied to the original depth map so that the generated point cloud only incorporates points inside the ROI and, thus, only reflects the door and handle themselves. The number of points (resolution) of the selected region is kept the same but everything else is deleted, meaning faster processing time. Once again, a pass through filter is applied to remove all those points out of 80-120 cm in hight region in which handles are usually found.

The strength of this approach resides in the planar segmentation of the point cloud, that is, finding all those points that support a plane model. If the ROI provided by the CNN is

of a high degree of precision, the generated point cloud will only be that of the door. In spite of this, sometimes, small parts of the neighboring walls or the environment behind a semi-open door are also represented, but their removal will be dealt with, once again, in the method fusing process III-C. Therefore, a planar model is extracted from the point cloud, the one that is supported by the vast majority of points, which will represent the door surface. Those points that fit into the plane model given a certain threshold (3-4 centimeters) will be considered as *inliers*; and those that fail to follow the model will be *outliers*. Being able to obtain a model of the door allows for robots to navigate closer to the door if they are too far away to detect the handle. This will be necessary since, as shown in the experimental results (IV), the door handles become harder to detect after the 1.5 meter distance mark. The coefficients of this plane are known, hence, the orthogonal vector to the surface can be derived. This is essential information for future opening of doors so that the robot knows in what direction to push/pull.

This method does sometimes provide false positives since not all *outlier* points correspond to the door handle. For instance, depending on the robots position relative to the door or cabinet, the ROI extracted from the CNN might not be tight enough and surrounding walls could appear in it. To solve this problem, both methods, visual segmentation and planar model extraction, will be merged.

C. Fusing both methods

The problem with the first method (visual segmentation) is that items located on top or around the door plane might be detected as false positives when applying the k-means clustering and will appear in the resulting point cloud. The main issue with the second method (planar model extraction) is that all points that do not fit into the plane model are kept in the point cloud. Hence, even though both methods obtain accurate and robust results, for specific cases, false positives are generated and must be dealt with.

A solution that removes false positives whilst keeping the accuracy of both methods is easy to implement. Similar to a bitwise AND operator where the output of the operation is true if, and only if, both inputs are also true simultaneously; the application of a similar approach to all points in both point clouds results in a highly precise detection of only handles. A comparison is done to see if a certain point exists simultaneously in the point clouds generated by the two methods If, indeed, it is located in the same position at the same time, in both point clouds, then the point is kept as a good estimate of the handles location. If not, it is removed. With this simple process, the imperfections of each method are overcome. All the objects (like posters, signs or even locks) clustered by the visual method will be removed because those points will be considered as *inliers* in the second method. Likewise, the possible errors of the second method will be removed because they will not appear in the resulting point cloud of the first. An evident example of this is when the ROI is not extremely precise and parts of the environment behind or around the door will show up inside the ROI. Even though, the second method will output them as true since they are *outliers* of the plane model, the first (visual) method will reject them. This is because the contour of those parts will be too large and certainly too vertical to be able to be recognized as handles, as shown in Fig. 5.

The outcome of merging both methods is a consistent, precise and robust estimation of the handles point cloud.

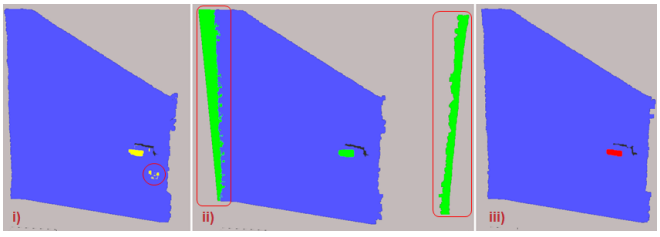


Fig. 5: **Method fusion.** **i)** Handle point cloud (yellow) obtained from the visual segmentation method (lock is included in the result, which is not desirable). **ii)** Handle point cloud (green) derived from the planar model extraction method (walls are included in the result but need to be removed). **iii)** Final handle point cloud (red) after merging both methods (all errors from each method have been solved).

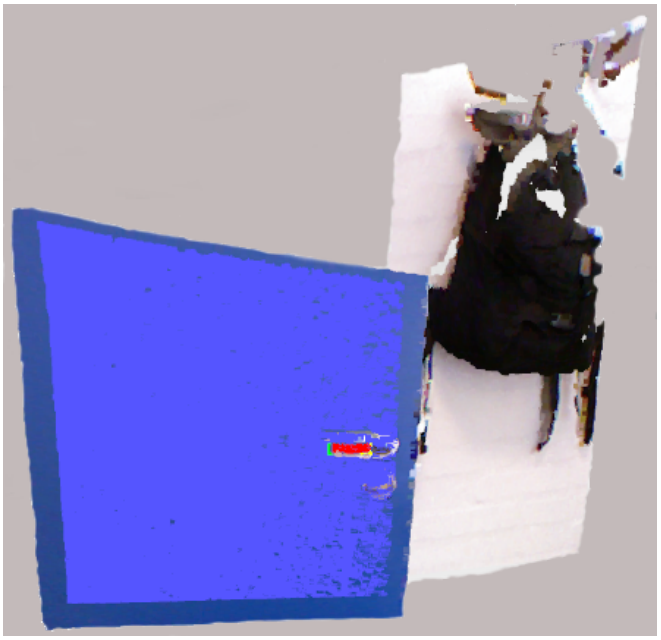


Fig. 6: The handle (red) and door plane (clear blue) point clouds obtained after using both methods are layered on top of the full environments point cloud. This visual segmentation method for this case is seen in Fig. 4 and the method fusion corresponds to Fig. 5. The green and yellow points on the left and right side, respectively, of the handle point cloud are some of the key features required for future robotic manipulation.

D. Extraction of key data from final door handle point cloud

Once the final handle point cloud is derived, some key features need to be determined to allow robotic manipulators to interact with them. These key features are: the orthogonal vector to the door plane (calculated previously), the direction in which the handle has to be turned, the point around which the handle turns and a possible grasping position.

The handle turn orientation (either clockwise or anti-clockwise) is determined by knowing if the handle is closer to the right edge of the door (thus the turn needs to be anti-clockwise) or to the left (clockwise). To achieve this, the center of the contour that represents the door handle is found and the distances to both edges of the ROI are calculated to see which one is smaller. This is done during the visual segmentation method.

The handles turning point will be the edge point in the Y plane of the point cloud, in one or the other direction, depending on the handles turn orientation (clockwise means said point is on the right side of the point cloud and viceversa). The possible candidates for these turning points are shown in Fig. 6, 7a and 7b .

Finally, the grasping position is left to the robots interpretation. The centroid of the point cloud and the principal direction are derived. All points are then projected onto said vector. This results in a handle axis with all points from the point cloud projected onto it. Thus, the robot is able to infer the location of grasp by choosing any of the points. The centroid is, however, highly recommended, but there are other possibilities too.

IV. The Experimental Results

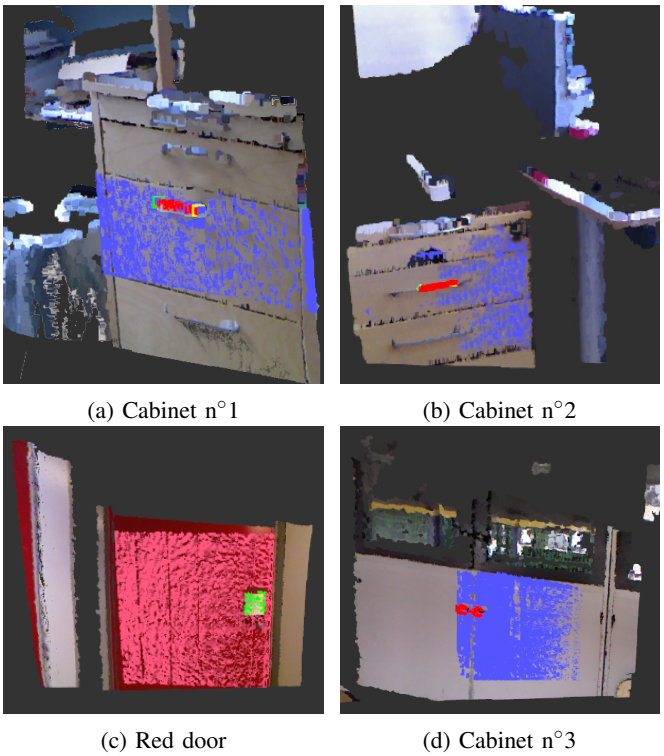
The hardware used for the door recognition via CNN is as follows: Intel Core i7-6700 CPU @ 3.40GHz with 8Gb of RAM and an NVIDIA GeForce GT 730 GPU with CUDA. The performance and speed of the neural nets can be improved with a more advanced setup. For the training step, a batch size of 64, a momentum of 0.9 and a decay of 0.0005 were used.

The proposed method was tested for accuracy and consistency on door handles. To do so, several door and robot positions were evaluated: the robot was positioned in front of a closed door, a semi-open door (35°) and open door (70°), at ranges of 0.5, 1 and 1.5 meters. For all cases, the CNN was able to detect consistently the object and provide a tight bounding box around it. It was observed that the depth camera was not able to pick up and differentiate precisely points from the door plane and from the handle after the 1.5 meter mark.

The same experiment was repeated for these three distance ranges but with a relative angle between door normal and robot of 30 degrees on both sides. To test the performance of the algorithm, each test was evaluated during 10 seconds and the number of attempts that provided a good, bad or unknown result were recorded. A good result is considered as the procurement of a handle point cloud at least as big as half of its surface. An unknown result is an attempt that fails to provide any type of point cloud. A bad result is a point cloud that does not strictly correspond to the handle. The full outcome is presented in Table I. Finally, the consistent extraction of a planar model corresponding to the door surface was evaluated for all tests.

It is evident that for ranges going up to 1.5 meters, the presented method is able to detect accurately door planes and its handle point cloud. As seen in the results, the further you move away from the door, the worse the performance is. This is mainly due to the following factors:

- Firstly, the small errors during the calibration process become more evident the further away you are from the door; this means that when rectifying depth and RGB images, the pixel-to-pixel matching will not be perfect, thus, generating a point cloud that is not correct. This is easily seen around the handle area, where pixels that have been successfully detected as handle are being represented on the point cloud as part of the door plane. For larger ranges than 1 meter, this problem is the main reason for attempts resulting in *unknowns*.
- Another factor is that the ROI that the CNN outputs



(a) Cabinet n°1 (b) Cabinet n°2

(c) Red door (d) Cabinet n°3

Fig. 7: Each image displays the corresponding handle point cloud from the ROIs extracted in Fig. 3. The handle point clouds are represented in red and the detected planar surface in blue (except for the case of the red door in which it is green and pink respectively). it is worth noticing how, with some simple minor tweaks to the algorithm (like what is the minimal distance between points for them to be considered as part of the same plane or size and orientation of the rotated rectangles around the segmented handle clusters), it can be easily adapted to non rectangular handles, as shown in the red door and cabinet n°3 images.

TABLE I: EXPERIMENTAL RESULTS

		HANDLE DETECTION											
		CLOSED DOOR (0°)				SEMI-OPEN DOOR (35°)				OPEN DOOR (70°)			
Range (m)	**Angle (°)**	**Good (%)**	**Unknown(%)**	**Bad (%)**	**Attempts**	**Good (%)**	**Unknown (%)**	**Bad (%)**	**Attempts**	**Good (%)**	Unknown (%)	**Bad (%)**	**Attempts**
0.5 m	-30°	70.6	23.5	5.9	17	66.7	28.6	4.8	21	$*^1$	$*^1$	$*^1$	-
	0°	90.5	9.5	0.0	21	100.0	0.0	0.0	16	100.0	0.0	0.0	16
	30°	81.0	19.0	0.0	21	69.2	30.8	0.0	26	68.2	31.8	0.0	22
1 m	-30°	88.9	11.1	0.0	18	93.8	6.2	0.0	16	$*^1$	$*^1$	$*^1$	-
	0°	57.9	36.8	5.3	19	87.5	12.5	0.0	16	61.9	23.8	14.3	21
	30°	55.6	44.4	0.0	18	84.2	10.5	5.3	19	80.0	20.0	0.0	20
1.5 m	-30°	18.8	81.2	0.0	16	18.8	62.4	18.8	16	$*^1$	$*^1$	$*^1$	-
	0°	33.3	60.0	6.7	15	41.2	47.1	11.7	17	14.3	71.4	14.3	14
	30°	11.8	76.4	11.8	17	30.0	60.0	10.0	20	22.2	66.7	11.1	18

$*^1$ In this position, the robot would be looking almost perpendicular to the door plane so the handle extraction was not possible.

might not be perfectly aligned with the door. In reality, the further away the robot is from the door, the looser the bounding box generally is. Hence, some parts of the surrounding image might be included inside the region of interest. Specifically in the case of walls next to a closed door, the visual segmentation might detect contours that meet the required specifications, and the method based on planar segmentation of the door will also accept those points. Therefore, some parts from the wall might be presented as belonging to the handles point cloud, thus resulting in an attempt classified as bad. The effects of this problem usually happen after the 1.5 meter barrier.

- Finally, as mentioned previously, after the 1.5 meter mark the depth sensor starts having issues differentiating precisely points from the handle and the door plane. This results in an evident increase in the number of *unknown* attempts since the second method is not able to produce a good handle point cloud. A good way to solve this problem is by being able to detect handles with specularity as presented by [1].

It was observed that for all cases in which an object is detected via the neural net, a robust and precise model of its surface is obtained. In cases where the robot is far from the object, the detection precision will be lower. This is not significant because for all cases, the robot will obtain an estimated position of the object and can easily move closer to it for a more definite detection, if required.

Lastly, the algorithm was also evaluated in front of diverse types of drawers, cabinets and doors. Some of the results are shown in Fig. 7.

V. CONCLUSIONS

The proposed approach accurately recognizes doors and cabinets in dynamically changing environments by virtue of a trained Convolutional Neural Net. The state-of-the-art is pushed even further by using two different methods (visual segmentation and planar model extraction) and fusing them together to produce a precise and consistent point cloud of the handles. Specific key features can then be extracted for robotic manipulation.

REFERENCES

[1] T. Ruehr, J. Sturm, and D. Pangercic, "A Generalized Framework for Opening Doors and Drawers in Kitchen Environments," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Minnesota, USA, May 2012, p. 38523858.

[2] M. Quigley, S. Batra, S. Gould, E. Klingbeil, ..., and A. Y. Ng, "High-Accuracy 3D Sensing for Mobile Manipulation: Improving Object Detection and Door Opening," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 2009, pp. 2816–2822.

[3] S. Kim, H. Cheong, D. H. Kim, and S. K. Park, "Context-based Object Recognition for Door Detection," in *Proc. IEEE International Conference on Advanced Robotics: New Boundaries for Robotics (ICAR)*, Tallinn, Estonia, June 2011, p. 155160.

[4] U. Adar and L. Bayindir, "Door Detection Using Camera Images obtained from Indoor Enviroments ," in *Proc. IEEE Signal Processing and Communications Applications Conference (SIU)*, Vijayawada,India, May 2015, p. 20052008.

[5] C. Chen and Y. Tian, "Door Detection via Signage Context-based Hierarchical Compositional Model," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops (CVPR Workshops)*, San Francisco, California, USA, June 2010, pp. 1–6.

[6] M. M. Shalaby, M. A. M. Salem, A. Khamis, and F. Melgani, "Geometric Model for Vision-based Door Detection," in *Proc. IEEE International Conference on Computer Engineering and Systems (ICCES)*, Kuala Lumpur, Malaysia, 2014, p. 4146.

[7] X. Yang and Y. Tian, "Robust Door Detection in Unfamiliar Environments by Combining Edge and Corner Features," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops (CVPR Workshops)*, San Francisco, California, USA, June 2010, p. 5764.

[8] T. H. Yuan, F. Hashim, W. Zaki, and A. B. Huddin, "An Automated 3D Scanning Algorithm using Depth Cameras for Door Detection," in *Proc. Electronics Symposium: Emerging Technology in Electronic and Information*, 2015, p. 5861.

[9] S. M. Borgsen, M. Schoepfer, L. Ziegler, and S. S. Wachsmuth, "Automated Door Detection with a 3D-Sensor," in *Proc. Canadian Conference on Computer and Robot Vision (CRV)*, Montral, Quebec, May 2014, p. 276282.

[10] M. Derry and B. Argall, "Automated Doorway Detection for Assistive Shared-Control Wheelchairs," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013, p. 12541259.

[11] Y. Zhou, G. Jiang, G. Xu, X. Wu, and L. Krundel., "Kinect Depth Image Based Door Detection for Autonomous Indoor Navigation," in *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Edinburgh, Scotland, UK, Aug. 2014, p. 147152.

[12] E. Klingbeil, A. Saxena, and A. Y. Ng, "Learning to Open New Doors," in *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, Oct. 2010, p. 27512757.

[13] W. Chen, T. Qu, and Y. Zhou;, "Door recognition and deep learning algorithm for visual based robot navigation," in *Proc. IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Legian, Bali, Dec. 2014, pp. 1793–1798.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," May 2016, to be published.

[15] W. Meeussen, M. Wise, S. Glaser, S. Chitta, C. McGann, P. Mihelich, ..., and E. Berger, "Autonomous Door Opening and Plugging In with a Personal Robot," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, Alaska, USA, May 2010, pp. 729–738.