Technical University of Denmark

Door and cabinet recognition using convolutional neural nets and real-time method fusion for handle detection and grasping

Maurin, Adrian Llopart; Ravn, Ole; Andersen, Nils Axel

Link back to DTU Orbit

DTU Library
Technical Information Center of Denmark

Door and Cabinet Recognition Using Convolutional Neural Nets and Real-Time Method Fusion for Handle Detection and Grasping

Adrian Llopart, Ole Ravn, Nils. A. Andersen
Department of Electrical Engineering
Technical University of Denmark
Lyngby, Denmark
e-mail: adllo@elektro.dtu.dk

Abstract—In this paper we present a new method that robustly identifies doors, cabinets and their respective handles, with special emphasis on extracting useful features from handles to be then manipulated. The novelty of this system relies on the combination of a Convolutional Neural Net (CNN), as a form of reducing the search space, several methods to extract point cloud data and a mobile robot to interact with the objects. 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. 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). A mobile robot for grasping the handle. The system assumes no prior knowledge of the environment.

Keywords-convolutional neural network; image processing; pointcloud processing; mobile robot; door recognition

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, further research can still be conducted to achieve full autonomy in the exploration of 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. Several approaches have been explored to solve this situation; however, most of them have focused only on the detection of doors and not interacting with them. These techniques incorporate the usage of either images ([1], [2] and [3]) or depth data ([4] and [5]).

Some approaches do indeed deal with the manipulation of door handles, but also present some limitations: [6] requires a prior semantic map; [7] directly detects handles using a 2D sliding window classifier but assumes the robot has already detected a door and is right in front of it. [8] can easily be considered *state-of-the-art* due to its methods of detecting doors and handles and manipulating them; however the generation of the necessary data takes 10 seconds plus some extra computation time to finally be able to open a door. The pipeline presented in this paper will challenge these limitations.

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 to unknown and dynamically changing scenario whilst, simultaneously, reducing computational time. This paper presents a framework that takes advantage of a CNN (similarly to the one proposed in [9]) to generate a ROI, which will significantly reduce computational time, and false positive rates in later stages of the system. Thus, the main contribution is a real-time integrated approach to the detection and manipulation of doors and cabinets, with no prior knowledge of the environment.

To achieve this, rectified RGB and Depth images will be obtained from a Kinect sensor. A Convolutional Neural Net, previously trained over several hundred door and cabinet images from *ImageNet* [13], 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. Then, 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 will be derived. Finally, a mobile platform will grasp the handle. The entire process is shown in Fig. 1.
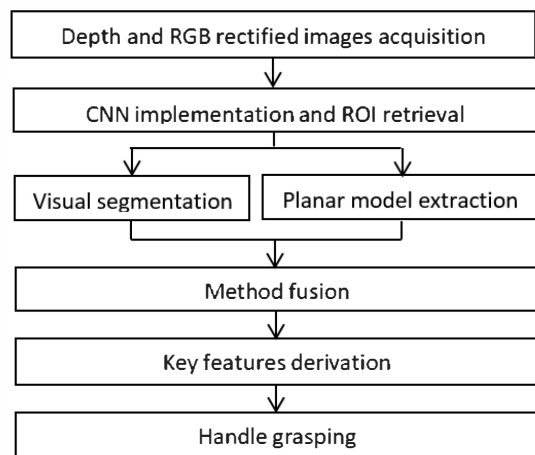
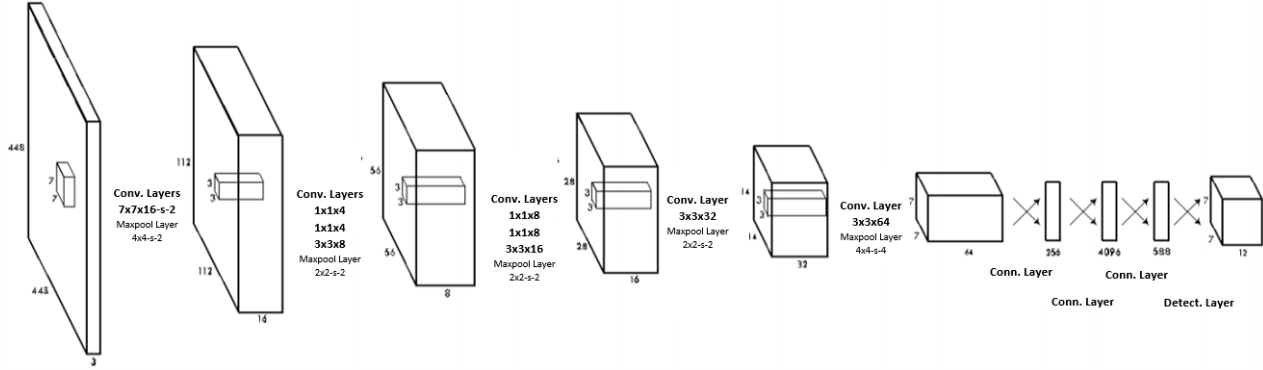Figure 1. Pipeline of the proposed method

Figure 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.

II. CONVOLUTIONAL NEURAL NET

The initial step of the presented method is the use of CNN as a preliminary approach to recognize, detect and segment a ROI out of a full image.

The vital use of a CNN will increase responsiveness and performance since it will significantly reduce the amount of data that needs to be processed in later steps of this framework. To achieve this, the proposed CNN must be extremely fast in its entire process, optimized and reliable, generating correct bounding boxes with a high level of precision around those objects it has been trained to detect. Hence, a unified architecture that predicts bounding boxes and class probabilities directly from full images is required: The YOLO Detection System [10].

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. With a GeForce GT 730 and 2 classes, the Net reaches 14 FPS. These two classes are doors and cabinets (the latter also includes drawers and lockers). The images are selected from *ImageNet* [13], with a total of 510 images for the doors, and 420 for the cabinets. These images were resized to a maximum of 448 pixels. The training took 3 days considering the learning ratios and hardware.

Figure 3. Resulting bounding boxes for several cabinets and doors. Their respective door and handle point clouds are depicted in Figure 8

Some results of the bounding boxes found by the CNN can be 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.

III. HANDLE´S POINT CLOUD GENERATION

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

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 given ROI, outputs an image containing only two colors; ideally the one corresponding to the surface of the door and the one of the handle. The goal is to create a clear differentiation between door plane and handle, from which a mask can be extracted. One of the main false positives that this method may generate, is the detection of the lock beneath the handle.

To limit the detection specifically to handles, the contours (obtained by applying a Canny Edge Detector) are evaluated in size, form and orientation. 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 Figure 4.

To obtain the handles point cloud, a rectified depth image is required which is obtained directly form a Kinect sensor and the ROS *Indigo* interface. It has been transformed and

projected onto 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. Thus, the resulting binary mask from the visual segmentation is applied to the rectified depth image and knowing the intrinsec parameters of the camera, the handles point cloud can be generated.
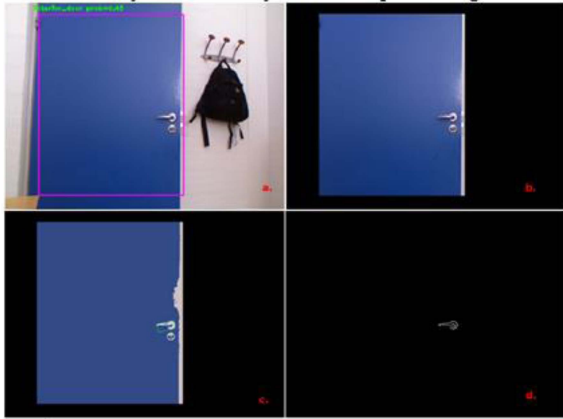

Figure 4. Method 1: Visual segmentation. a. ROI selection from the output of the CNN (with class name and probability written on top). b. Cropped rectified RGB image. c. K-means clustering of ROI with rotated rectangles (green) and centers (blue) drawn over those contours that fit into certain specifications.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

The derived point cloud will display the door handles 3D information precisely but might incorporate some impurities. In the case of doors, this noise is 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 obtained by the described visual segmentation method and the one generated from plane extraction are merged together. This is explained in section III-C.

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 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 [11] 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 [8], 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 reduced. 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 height region in which handles are usually found for doors.

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. Therefore, a planar model, that is supported by the vast majority of points, is extracted from the point cloud, 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 (V), the door handles become harder to detect after the 1.5 meter distance mark.

This method does sometimes provide false positives: small parts of the neighboring walls or the environment behind a semi-open door might also represented, but their removal will be dealt with, once again, in the method fusing process.

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. Therefore, a comparison is done to see if a certain point exists simultaneously in the point clouds generated by the two methods. 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 Figure 5. The outcome of merging both methods is a consistent, precise and robust estimation of the handles point cloud. From this result, several handle features can be obtained.
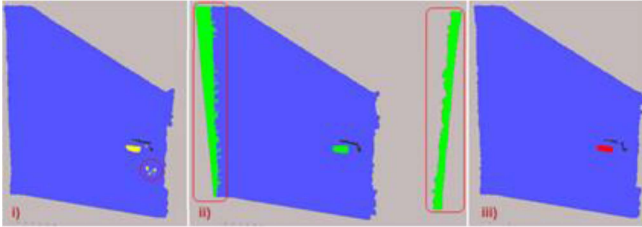

Figure 5. Method fusion. i) Handle point cloud (yellow) obtained from the visual segmentation method (lock is included in the result, which is not desirable). ii) Handle point cloud (green) derived from the planar model extraction method (walls are included in the result but need to be removed). iii)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

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 (as presented in [7]) to allow robotic manipulation. These key features are: the orthogonal vector to the door plane, the turning direction of the handle, the point around which the handle turns and a possible grasping position.

Since the coefficients of the door/cabinet plane are known, the orthogonal vector to the surface can be derived.

Figure 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 Figure 4 and the method fusion corresponds to Figure 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.

The handle turn orientation (either clockwise or anticlockwise) is determined by knowing if the handle is closer to the right edge of the door (thus the turn needs to be anticlockwise) or to the left (clockwise). To achieve this, the centre of the door handles contour 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 green and yellow in Figure 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 resulting 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. MOBILE ROBOT AND MANIPULATOR

A mobile platform (Figure 7) has been developed to put into practice the methods presented in this paper. It has been built as a combination of an *iRobot-ATRV-Jr*, an *UR5* robotic arm, a *Robotiq 2-Finger Adaptive Robot Gripper,* a *Robotiq Force-torque sensor* and a *Kinect* camera. The CNN mentioned before runs on al *GeForce GT 730M GPU* installed on an external computer that communicates directly with the robot. All components have been connected and share information via the *ROS Indigo* interface over Linux 14.04 (LTS) Trusty Tahr distribution.

Figure 7. Mobile robot grasping door and cabinet (drawer) handles

With the use of the *MoveIt* library [12], the manipulator achieves the grasping position (the centroid of the handle) and orientation, avoiding self collision and other obstacles. In practice, it was seen that the best grasping results were achieved when the goal position was set to be a few centimeters further away from the surface following the orthogonal vector to the door. Then the manipulator can slowly approach the handle following that same vector until the force sensor detects a contact with the door surface. Finally, closing the *Robotiq* gripper ensures a correct grabbing of the handle (Figure 7). Additionally, if the

detected handle is out of reach, the robot is able to navigate closer to the door first and then attempt the handle grasping.

V. 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 net 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 CNN proposed in this paper was tested on 210 images (241 doors and cabinets [N_d]) and achieved a true positive detection rate (TP) of 90.0% and a false positive rate (FP) of 6.64%. Comparatively, the proposed CNN has a higher detection rate of doors and cabinets than [2], [4], [5] and [9], but marginally lower than [3].

(a) Cabinet n°1   (b) Cabinet n°2
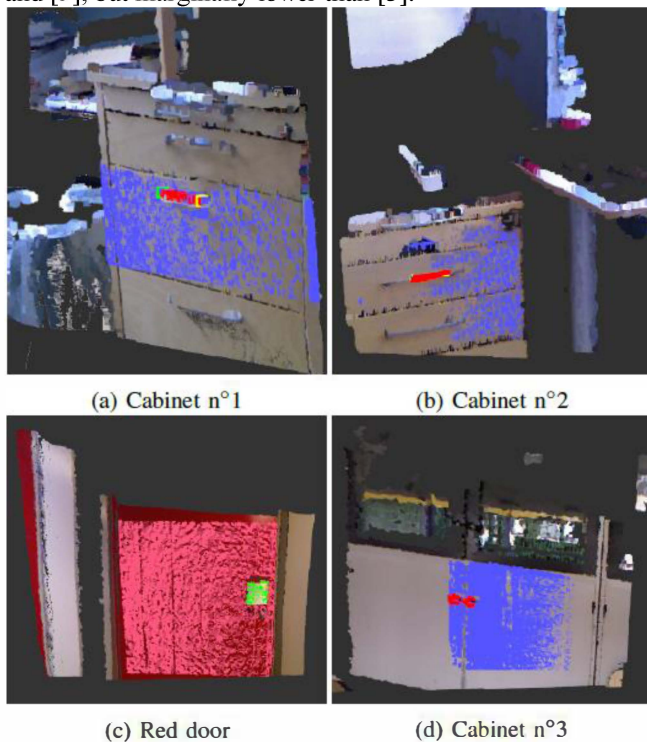
(c) Red door   (d) Cabinet n°3

Figure 8. Each image displays the corresponding handle point cloud from the ROIs extracted in Figure 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 (evaluation of size and orientation of the contours), it can be easily adapted to non-rectangular handles, as shown in the red door and cabinet n°3 images.

Most of these erroneous results (FP) occur when blank walls next to the objects are considered positive. These results do not generally affect the posterior handle detection rates though, since usually no handles exist in that region. However, to safeguard completely against these errors, those detections with a lower confidence than a certain threshold are completely removed. This is done because it is better to have a higher omission rate than a FP rate,

especially considering that the proposed CNN has an update rate of 14Hz and so new images are processed constantly.

TABLE I. DOOR DETECTION

	Shalaby et. al [2]			Yang et. al [3]			Proposed CNN		
	N_d	TP	FP	N_d	TP	FP	N_d	TP	FP
Simple	55	100%	3.6%	55	98.2%	1.8%	63	98.4%	7.53%
Medium	94	82.7%	17.3%	93	91.4%	1.1%	107	95.3%	4.7%
Complex	63	68.2%	31.7%	56	85.7%	7.1%	71	74.6%	8.45%
Total	212	82.8%	26.4%	204	91.7%	2.9%	241	90.0%	6.64%

The proposed system was tested for accuracy and consistency on door handles. To do so, several door and robot positions (Table I) were evaluated (similarly to [4]): the robot was located in front of a closed door (0ª), a semi-open door (35ª) and open door (70ª), at ranges of 0.5, 1 and 1.5 meters. 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, unknown or bad result were recorded. A *good* result is considered when the obtained handles point cloud is 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 II.

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 is one of the reasons for attempts resulting in *unknowns*.

- The further away the robot is from the door, the looser the ROI that the CNN outputs generally is, meaning some parts of the surrounding image might be included inside it. The consequence is that 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.

- Lastly, around the 1.5 meter mark the hardware limitations of the sensor result in an evident increase in the number of *unknown* because the second method is not able to produce correct handle point clouds. Meeussen *et al.* [8] presented a system that is able of detecting handles with specularity.

The presented system achieves an average 0.96% error rate at handle detection for a distance of 0.5 meters, 3.11% for 1 meter and 10.55% for 1.5 meters. This clearly outperforms previous approaches at distances shorter than 1.5 meters. Kim *et al.* [1] only detected 50% of the tested handles; Meeussen *et al.* [8] obtained around a 60% success rate in handle detection; Ruehr *et al.* [6] reached a stunning 93.3% of true positives using only depth data but had prior knowledge of the door and cabinet locations; Klingbeil *et al.* [7] also reached 93.2% using only visual features. Hence, it is proven that fusing both approaches (visual segmentation and planar model extraction) allows for much better results than each individual method.

TABLE II. HANDLE DETECTION

Range (m)	Angle (°)	CLOSED DOOR (0°)				SEMI-OPEN DOOR (35°)				OPEN DOOR (70°)			
		Good (%)	Unknown (%)	Bad (%)	Attempts	Good (%)	Unknown (%)	Bad (%)	Attempts	Good (%)	Unknown (%)	Bad (%)	Attempts
0.5	-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	-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	-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

The average omission (*unknown*) rates are 14.65%, 20.66% and 65.65%, respectively. These are not very relevant because the fast update of data ensures results are obtained almost every time step. The average processing time of the end-to-end system (except the grasping) is of 1.8Hz; which is much faster than [5], since that approach does not detect handles, and [8], where just the data collection for one trial works at 1Hz.

It was observed that every time an object was detected by the CNN, a consistent model of its surface was obtained. In cases where the robot is far from the door/cabinet surface, the handle detection precision will be lower. However, the robot will always achieve 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. 8.

VI. CONCLUSIONS AND FUTURE WORK

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 further by using a combination of a CNN, two different methods (visual segmentation and planar model extraction) and their fusion to produce a precise and consistent point cloud of the handles. Specific key features can then be extracted so that the designed robot can grasp the handles and open/close the doors and cabinets.

In the future, the CNN will be trained to detect additional classes and various algorithms will be applied to segment the point clouds correctly. Finally, a force feedback control system will be designed to select correct grasping poses based on geometry of the objects and to manipulate them securely (i.e. opening doors).

REFERENCES

[1] 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.

[2] 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, Sept. 2014, p. 4146.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, June 2016, pp. 779-788.

[11] R.B. Rusu and S. Cousin, "3D is here: Point Cloud Library (PCL)", in Proc. IEEE International Conference on Robotics and Automation (ICRA), Shangai, China, May 2011.

[12] I. A. Sucan and S. Chitta, "MoveIt!", [Online] http://moveit.ros.org

[13] J. Deng, "Imagenet: A Large-Scale Hierarchical Image Database", [Online] http://image-net.org