Semantic Place Categorisation and Object Detection using CNNs

Rune Grønhøj, Jan Kjær Jørgensen, Guilherme Mateus Martins, Jacob Krunderup Sørensen

***Abstract*—Place Categorisation and Object Detection are two distinct study fields in the broader field of robotics, which are currently being researched and every year there are advancements in the state-of-the-art methods and algorithms. In this paper we propose the integration of two previously released algorithms into a mobile platform. For Place Categorisation, a convolutional neural network (CNN) model using the Caffe framework is deployed through an open-source ROS[1] package and processed using DBSCAN[1]. For Object Detection, the algorithm You Only Look Once (YOLO)v3[2] using Darknet[2] is deployed and post processed to compute an object's position. The deployment of these algorithms is done using a TurtleBot3[3], augmented with a NVIDIA Jetson AGX Xavier[4], and two Intel RealSense cameras[5] The integration done in this paper allowed the TurtleBot3 to assign semantics to a metric map, even though the tests revealed that the perspective of the robot offset the results, causing the system to assign wrong semantics to the map. Furthermore, the pre-processed map ignores any constraints regarding the structure of the metric map, assigning at times semantics centroids in locations the robot cannot access. Finally, even though the robot understands the concept of association between objects and the semantics of a place, the fact that the robot cannot access some of these centroids, makes it impossible to detect objects, as wanted.**

***Index Terms*—Place Categorisation, Mobile Robotics, Semantic Mapping, Object Detection, Convolutional Neural Network**

I. Introduction

Fusing metric maps with semantic information, enables mobile robots to understand the environment beyond navigation and obstacle avoidance. The idea of *Semantic Mapping*[3] is to allow a robot to understand the environment around it on a higher level. Instead of just understanding the metrics of the walls and the obstacles in its surroundings, the robot obtains higher level information, such as objects and an understanding of the place it is seeing. This allows the robot to effectively perform smarter decisions while navigating, since it allows for a close understanding of the world surrounding it. One of the methods to assign semantics to a metric map is through *Place Categorisation*[4], which relies on general descriptions of the scene the robot encounters itself in.

Considering the information presented above, assigning semantics opens up a whole new world of possibilities that are not possible by simply using a metric map. A fetching task can, arguably, be optimised by providing the robot the knowledge to understand different layers of semantics, such as understanding that a cup belongs in a kitchen and after arriving to such a place it should begin searching for it.

However, simply locating and navigating to the desired room is not sufficient, in order to fetch the desired object, it needs to be detected when the robot arrives at the semantic place where it can be found. This task is a completely different field of robotics called *Object Detection* [5] and focuses on finding the position of objects, while understanding what the object is by classifying it.

This work aims to develop upon other semantic mapping methods, to showcase that such algorithms can be implemented in different systems and yield similar results, while being complemented by object detection algorithms. The main focus is to perform place categorisation with a pre-trained CNN, in order to enable semantic mapping, which augments a metric map. Essentially, the augmented map is used as the baseline for global path planning while fetching an object, since it allows the robot to navigate to places where specific objects are more likely to be located.

Section II contains a discussion on works related to *Semantic Mapping* and *Object Detection*. Section III gives an overview of the system integration, alongside software and hardware overviews of the robotic system. Section IV covers the experiments including a description of the specific robotic system, and results, based on the experiments. Lastly, section V concludes the paper and future work is discussed.

II. Related Works

The problems of augmenting metric maps with semantic information and detecting objects are widely studied areas of robotics, and a variety of approaches exist for either of the problems. In this section a short overview of related works is presented, in order to facilitate inspirations to this paper and identify the contributions of this paper.

A. Semantic Mapping

As previously mentioned, *Semantic Mapping* is a broad area of robotics and there are several possible approaches to solve problems. Some authors use methods relying on the objects found in a scene, such as [6], where the distribution of the objects within a scene is used to compute the semantics of the place. On the other hand, *Semantic Mapping* can be performed by focusing specifically on semantics of the general context of a location, *Place Categorisation*, as presented by [7]. The authors of this paper considered the `Places205` CNN for *Place Categorisation*, while expanding the classifier

[1]Robot Operating System
[2]Machine learning framework for neural networks written in C.
[3]Educational and research mobile robotic platform developed by ROBOTIS.
[4]Embedded computer designed for AI and robotics.
[5]Specifically built for Computer Vision applications.

with a random forest classifier, which adds additional classes, without having to retrain the CNN. It requires no environment-specific information and can easily be expanded, as shown by the previously mentioned author. Though, there are some authors that use a fusion of the concepts mentioned above and infer semantic information by multiple cues, with both the knowledge of objects in the scene and the its general context, such as [8], by using a 3D mesh to generate a projection of a scene, and render images from different poses around the scene. These recreated images are then used as input for a retrained `ResNet-50` CNN, that categorised the scene. Their proposed system also takes into consideration the detected objects in a setting, by comparing them with a probability distribution map relating objects with different place categories. In [9] a Kinect camera is used for place categorisation for classification of five specified rooms. A Support Vector Machine (SVM) classifier is compared to the use of a random forest classifier using a small image data set. Promising results are shown, considering the data set and the methods applied.

B. Object Detection

Object Detection is a complex task with a mountainous amount of problems, which require a multitude of different algorithms. For example, the works performed by [10], where a Probabilistic Object Detection approach is introduced, in order to estimate the position of bounding boxes and object classes.

Another problem currently faced is that, in order to perform object detection in real time, there is a need to create lightweight and precise algorithms. This introduces a problem with the detection of objects at different scales since it can be quite computationally heavy, which [11] attempts to solve by replicating the classical image processing scale pyramid, by taking advantage of the structure of CNNs in order to create a multi scale feature map. A similar approach is used by [12], as a response to the singular dimension feature map from state of the art works by [13]. However, as of 2018, [2] has also included scale invariance into the design of the algorithm.

Another aspect we looked into is the two types of approaches currently used by a large amount of state of the art algorithms. The first is the two-stage approach and the second the one-stage approach. Each of these have their own drawbacks and advantages, such as the former being more precise than the latter, at the cost of being more computationally heavy. Examples of two-stage algorithms are the works performed by [14] and [15] on the Fast R-CNN and Faster R-CNN, to which [13] responded with YOLO, which is a one-stage approach algorithm to the detection of objects. Furthermore, as mentioned before [12] proposed Single Shot multibox Detector (SSD), also a one-stage algorithm, as a response to YOLO.

For the problem at hand, YOLOv3[2] using the Darknet framework is closely considered due to the broad community support surrounding the algorithm and the fact that it is open-source software. Furthermore, its online speed and accuracy while performing object detection and classification are remarkable compared to some of its contemporaries [2]. YOLOv3 works by dividing the frames acquired by the camera into small sub-regions, in which bounding boxes are fit to best describe the location of an object, while simoultaneously classifying regions of the image. YOLOv3 is trained on data with 80 classes, and is composed by 53 convolutional layers.

Inspired by the works presented above, this paper present the following contributions to the field of semantic mapping and place categorisation:

1) We present semantic mapping implemented on a lightweight system, with an embedded computer.
2) We show how to process a semantic map, and use it for autonomous navigation.
3) We demonstrate that semantic maps, can be useful when combined with object detection methods, for finding specific objects in the environment.

III. System Overview

This section presents the overall software and hardware architectures developed. The overall architecture of the system is presented in figure 1, showing how the software and hardware elements relate to each other, as well as presenting the different elements of the software architecture. As can be seen from the figure, mapping is done offline. This is meant in the sense, that the metric map is performed using manual control of the robot. This metric map is then used to perform semantic mapping, again using manual input. Finally, the map is processed, before being used for navigation.

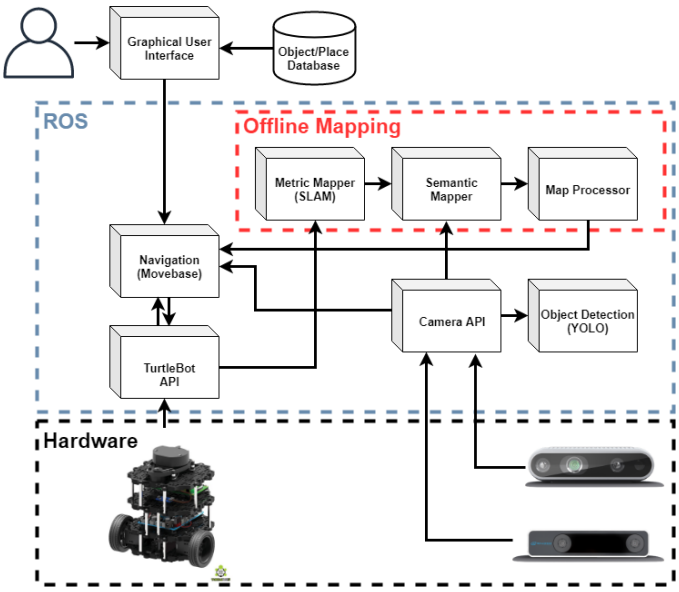

Figure 1. Overall system architecture, showing the relation between the hardware and software.

In addition, the figure shows that a user, aided by a GUI, selects an object, which the robot will then try to find. This is done by the robot assuming a certain object can be found at a certain semantic class, which the robot uses as its goal for the global path planning. During the motion period, the robot detects and avoids objects, using a local planner. Finally, after reaching the goal of the path planner, the robot performs object

detection by using YOLOv3, which uses the visual input from the camera.

A. Semantic Mapping

The data acquisition for Place Categorisation in this paper is performed by using the ROS package written by [7], alongside their paper. As previously mentioned, the package uses a CNN based on the `Places205` network, using the Caffe framework. In our instance, the package uses custom metric maps, which are then used to create custom semantic maps. These custom metric maps are acquired by performing SLAM with Gmapping, using however, visual odometry instead of the odometry performed by the wheel encoders of the TurtleBot3.

B. Processing of Semantic Map

The output from the semantic mapper is noisy and is not directly useful for navigation, hence some processing is needed. The fundamental information which is to be extracted from the semantic map are clusters of the mapped rooms and centroids of said rooms.

Multiple methods to achieve the desired output exist, such as the k-means algorithm and mean shift clustering. However, they are limited by the fact that the number of clusters has to be known beforehand. This prerequisite does not work in this case, since we want to build a flexible future proof system. Therefore, the DBSCAN algorithm presented in [1] can be used to achieve our goal. It takes eps and minPoints, which can be tuned for our case, where minPoints is a measure for how small a cluster can be and eps, relates to how dense the data has to be to be considered as clusters. Additionally, smaller clusters are considered as noise, which will help remove noisy areas from the semantic map, as wanted.

Although DBSCAN is used, it does not give the cluster centroids. Since these are wanted, they are found by taking the mean of the clusters, based on Euclidean distance. The metric map, semantic map, processed map and cluster centroids can be seen on figure 2.

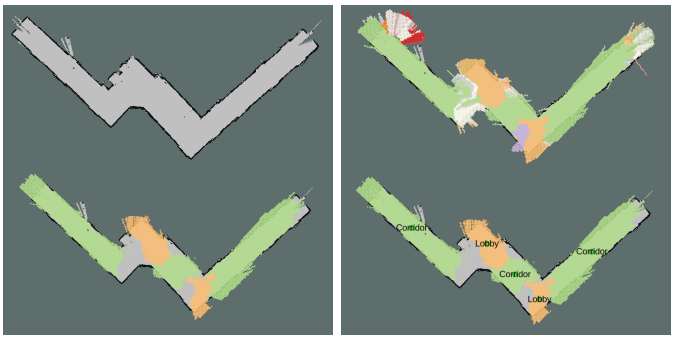

Figure 2. Processes of creating and processing semantic map. First metric mapping, then adding semantics, clustering the semantics and lastly finding centroids.

C. Object Detection

The object detection in this project is performed by deploying [2]'s YOLOv3. As previously mentioned, this algorithm assigns bounding boxes to different objects in a frame, allowing it to not only classify the object inside the bounding box, but to also track its position. The classification element of YOLO is important for our integration, since it allows the robot to understand it is facing the object it is supposed to find. However, this would not be enough, since there will be more objects placed in a room besides the object of interest. Therefore, the second aspect of YOLO is paramount on locating the object, since the location of the bounding box gives us a good estimation of the object's centroid, which is then used while locating it and relating it to the external coordinate frame of the camera. This estimation can be used in a frame transformation from the previously mentioned camera frame, into a map frame, which, at the current state of development, allows the user to understand where the object is located. This object detection is triggered by a ROS service, when the robot finishes navigating to the room the object of interest is located.

D. Robotic System

We use an augmented TurtleBot3 (Burger) by integrating a Jetson AGX Xavier in place of the Raspberry Pi, which is shipped as the default with the TurtleBot. Furthermore, an Intel RealSense T265 is equipped onto the Turtlebot3 as the T265 camera is Intel's V-SLAM tracking camera, that through visual SLAM and an IMU localises itself in 3D space. This is used as the odometric input for our system, as it showed better results than the odometry available from the wheel encoders. In addition, an Intel RealSense D435 is used for visual input for the place categorisation and object detection. Furthermore, the mobile robot is equipped with a laptop power bank, to make it truly wireless, when navigating. Furthermore, it was also found during the integration that the Jetson only contains one USB 3 port and one usable USB-C, which proved to be challenging due to the need to integrate the two RealSense cameras, the laser scanner, the control board of the TurtleBot, a Bluetooth Xbox controller, and finally a Wi-Fi dongle, since the Jetson does not contain an integrated communication board. All of these components are interfaced through USB.

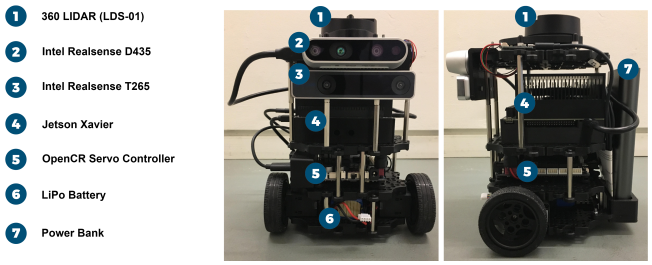

Figure 3. The fully assembled TurtleBot with the cameras mounted on the front and the battery mounted on the back. List of hardware components is included, with numbers.

IV. Experiment, Evaluation and Results

Three experiments are conducted. The first one is concerned with creating a semantic map and comparing it to a manually

created ground truth, based on our previous knowledge of the area. The second test relates to how well the robot can navigate to the centres of the detected rooms. Finally, the third test explores the capabilities of the object detection and determines how well, the desired object can be detected in a cluttered scene.

A. Place Categorisation

To evaluate the performance of the place categorisation on the system, 2 different areas have been mapped with both SLAM and semantic mapping, by having the robot perambulate the area. The results are compared to manually created semantic maps, based on an online map service. For this experiment to be successful at least 2 rooms of each area has to be classified correctly compared to the manually created semantic map.

As an exception, this test is conducted using the wheel odometry of the robot, since the ROS package is designed to use it, while crashing the system if visual odometry is used.

As an initial step to perform the test, a ground truth of the test areas was acquired as Test area 1(Left) and Test area 2(Right), as shown in figure 4, in order to be able to compare the results computed by the CNN and clustering algorithms.

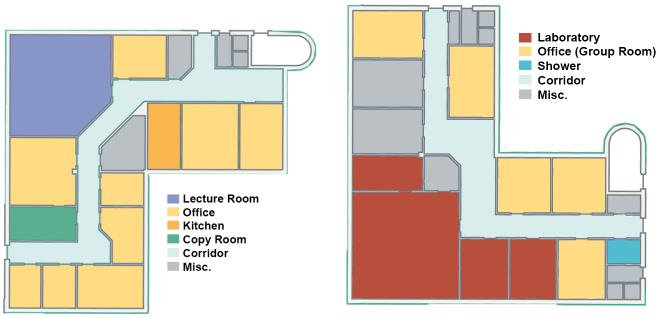

Figure 4. Ground truth map of the testing areas. Test area 1: Left. Test area 2: Right

Place Categorisation is performed on Test area 1 with figure 5 presenting the results of the semantics. During the tests, we had access to the corridor, the lecture room and the kitchen presented in the ground truth. Observing the figure containing the results shows that the robot is capable of classifying most of the corridor correctly, except for a small corner on the top right side of the map, which is classified as a lobby. At the same time, the lecture room is classified as a corridor, while the kitchen is classified partly correctly, and partly as a lobby. At the moment, even though this test did not fully meet the success criteria, there are some interesting aspects to it, which are reflected upon during the discussion.

The second *Place Categorisation* test is performed on Test area 2 with figure 6 showing the results of the semantics. during the tests, we had access to the corridor, one office and the a shower presented in the ground truth. Compared to figure 5, the test presented in figure 6 yields similar results, where most of the corridor is classified correctly, except for the large area in its centre, which is classified as a lobby. Meanwhile, the shower room is classified almost fully correctly, except for

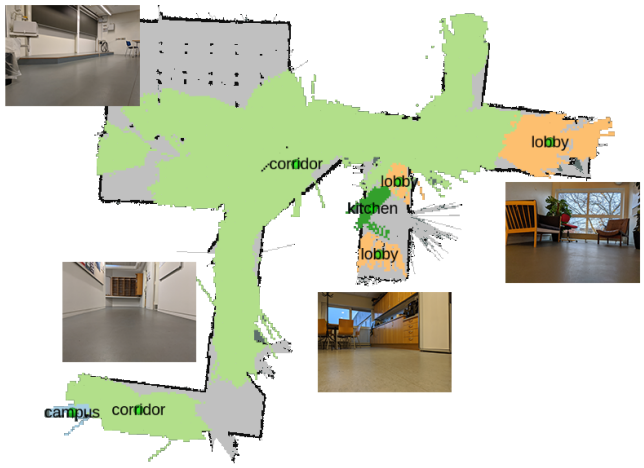

Figure 5. Semantic map result for test area 1

a small area on its corner which is considered as a campus. Finally, the office is classified as a corridor by the robot. The similar results of this test compared to the previous also create the possibility of a discussion, regarding the results.

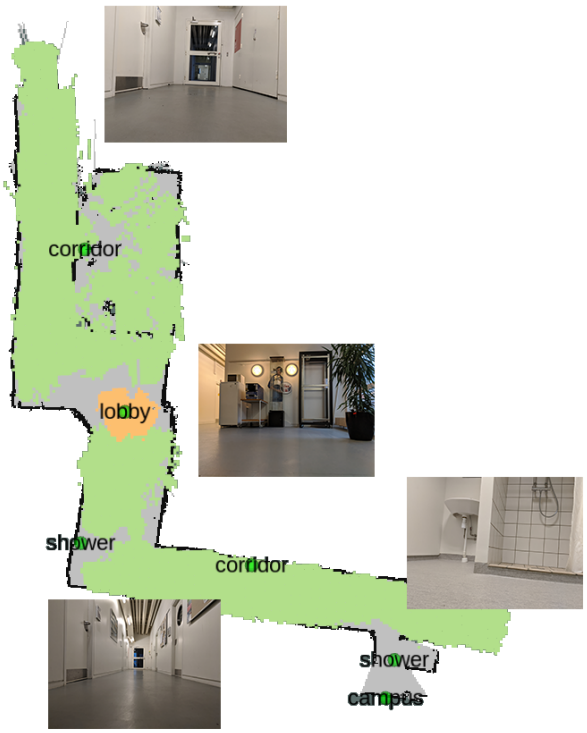

Figure 6. Semantic map result for test area 2

B. Navigation

To evaluate the performance of the robot's understanding of the semantics created for a metric map, the robot must be able to perambulate from its starting position to the centroid of a room. The robot must be able to gain this understanding according to the type of object a user wants the robot to find.

Object detection test	**Bottle**	**Cup**	**Cellphone**	**Fork**	**Book**	**Nothing**
Bottle	**9**	0	1	0	0	0
Cup	2	**8**	0	0	0	0
Cellphone	0	0	**9**	0	1	0
Fork	0	2	0	**7**	0	1
Book	0	2	0	2	**6**	0
Nothing	0	0	0	0	0	0
Success rate	90%	80%	90%	70%	60%	

Table I
EXPECTED RESULTS: CONFUSION MATRIX FOR OBJECT DETECTION TEST

E.g. a user inputs through the GUI created for the robot that it must find a laptop, and by accessing the database created for this project, it must understand that it can be found in an office. The robot must then navigate autonomously while localising itself by using the sensors at its disposal, as it must avoid obstacles on its way. In order for this experiment to be categorised as successful the robot must be able to successfully navigate to the correct room, and be able to understand that it currently encounters itself at the correct semantic location, in order to be able to trigger object detection.

Navigation test	**Corridor**	**Lobby**	**Corridor**
Corridor	0	0	0
Lobby	0	10	0
Corridor	0	0	0

Table II
CONFUSION MATRIX FOR NAVIGATION TEST

The navigation test results can be seen in table II. For the test the robot was first sent to the nearest corridor, then lobby and then finally to a corridor again. The robot went correctly to the lobby every time, however, due to the semantic map and clustering algorithm, the corridor centroid was placed inside what the robot believed to be a wall. This means that the robot was unable to make a plan, and navigate to the centroid, making the test unsuccessful.

C. Object Detection

To evaluate the performance of the detection of desired objects, after the robot has reached the correct semantic location, it must use the trigger given by the navigation step in order to launch object location. The object location must be able to distinguish between other objects in the scene and the object class which is aimed to be located. In order to achieve this, the robot must only consider objects which it classifies with over 90% confidence. Furthermore, when an object of interest is found, the robot must transform it from image frame-coordinates, into map coordinates, since at the focus of this project is not any type of manipulation, it should be able to provide the user with information of the location of the object. —Currently in development— The success criteria of this test is based on the capability of the robot being able to not only correctly detect an object, but also correctly converting its position into map coordinates.

The *Object Detection* test was performed on 5 different objects in the scene where the user chooses an object for the robot to locate as seen in figure 7. Table I shows the result of each object being detected in a confusion matrix.

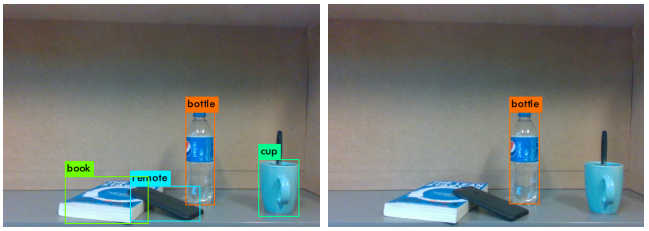

Figure 7. Detection of a water bottle, among other objects.

V. DISCUSSION AND CONCLUSION

Some of the acquired results from testing the system deviated from the initial expectations. One of the current hypothesis we have relates to the fact that previous uses of the network were performed at the height of a human [7]. In contrast, the camera mounted on the TurtleBot3 stands at around 19.7cm from the floor, which causes the field of view of the camera to only acquire information regarding lower parts of furniture. The fact that a large amount of the frames contain walls, might be causing the CNN to classify large chunks of the metric map as corridors. Additionally, we observed that raising the robot would yield different and at times better results. Furthermore, the results from figure 5 suggest that there is a small area of the corridor which was mapped as lobby. It is possible this happen since this region of the building is broader than the rest of the corridor and contains objects which are usually not found in corridors, but can be found in lobbies, such as sofas and plants. Even though the semantic information is inferred by a single cue regarding only the general context of the scene, the fact that these objects are present will affect the features found by the network and disturb the results. This is further justified by the similarly misguided results from the lecture room in this map, since even observing clean black boards seemed to offset the output from the CNN. However, by pointing the robot at a written black board, it is able to understand it is currently located at lecture room. Finally, the robot also seemed to be somehow uncertain after entering the kitchen, since it managed to correctly classify parts of the kitchen, however, it came to our understanding that some of the layout of this room could also be interpreted as a lobby out of context. Similarly to the test discussed above, the results presented in figure 6 show that the corridor was mapped mostly correctly except

for two small areas. The shower on the left side of the map could be considered noise that bypassed DBSCAN, while the area mapped as lobby could have similar causality as in the previous test, since this part of the corridor is broader than the rest, and contains several objects that can be found in lobbies. One thing that came to our attention is that whenever a plant is shown to the robot, the confidence of the CNN that the robot is located in a lobby increases. As for the rest of the map, the shower area was mapped almost correctly, except for a small corner which was labelled as a campus, which seems to be noise, while the group room was fully mapped as a corridor, possibly as a consequence of the perspective of the robot. As for the navigation between different places, the robot complies with the need to move from its current position to the centroid of a cluster. Even though the robot managed to navigate to the lobby in figure 6, it did not manage to navigate to either of the corridors, since the centroids of the clusters have a tendency to not abide by the shape of the metric map, since they are computed considering data which is out of bounds for the robot. Therefore, this problem could be solved by considering the architecture of the metric map when assigning semantics.

The database used for associating objects and place semantics could be vastly improved by using learning strategies, it would be possible to teach the robot which objects are to be expected in which rooms. This would improve the flexibility of the system greatly, and in theory let the robot make better association based on semantic information. On a related note, it would be interesting looking into voice commands. Telling the robot "I want a plate", and having the robot pick out the semantic keyword "plate" and understand where it most likely will be able to find said item, based on a semantic map. Furthermore, one of the long term goals of this prototype is to integrate a manipulator into the mobile platform. The position of detected objects can then be transformed from the image frame to the end-effector of the manipulator, in order to grasp the object and retrieve it to the user.

This work presented three main contributions. The implementation of semantic mapping on a lightweight embedded computer system, the Nvidia Jetson AGX Xavier. We showed a method for processing the semantic map, that however faced some problems, that made it impossible to use for navigation, at times. Lastly, we demonstrate that semantic mapping and object detection can be combined for finding and detecting specific object in the environment, for a fetching task.

ACKNOWLEDGEMENT

We thank our supervisor Dimitris Chrysostomou for good discussions and guidance on the whole process.

REFERENCES

[1] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[2] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[3] D. F. Wolf and G. S. Sukhatme, "Semantic mapping using mobile robots," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 245–258, 2008.

[4] J. Wu, H. I. Christensen, and R. J. M., "Visual place categorization: Problem, dataset, and algorithm," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4763–4770, 2009.

[5] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, 1998, pp. 555–562.

[6] K. Charalampous, I. Kostavelis, F. Chantzakou, E. Volanis, C. Emmanouilidis, P. Tsalides, and A. Gasteratos, "Place categorization through object classification," in *2014 IEEE International Conference on Imaging Systems and Techniques (IST) Proceedings*, Oct 2014, pp. 320–324.

[7] N. Sünderhauf, F. Dayoub, S. McMahon, B. Talbot, R. Schulz, P. Corke, G. Wyeth, B. Upcroft, and M. Milford, "Place categorization and semantic mapping on a mobile robot," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 5729–5736.

[8] M. Brucker, M. Durner, R. Ambruş, Z. C. Márton, A. Wendt, P. Jensfelt, K. O. Arras, and R. Triebel, "Semantic labeling of indoor environments from 3d rgb maps," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1871–1878.

[9] O. Mozos, H. Mizutani, R. Kurazume, and T. Hasegawa, "Categorization of indoor places using the kinect sensor," *Sensors (Basel, Switzerland)*, vol. 12, pp. 6695–711, 12 2012.

[10] D. Hall, F. Dayoub, J. Skinner, H. Zhang, D. Miller, P. Corke, G. Carneiro, A. Angelova, and N. Sünderhauf, "Probabilistic object detection: Definition and evaluation," 2018.

[11] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016, pp. 21–37.

[13] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, pp. 1–10, 2015. [Online]. Available: http://arxiv.org/abs/1506.02640

[14] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.