Learning Semantic Place Labels from Occupancy Grids using CNNs

Robert Goeddel Edwin Olson

***Abstract*—The goal of this paper is to develop a robot with a grounded spatial vocabulary. Such a vocabulary would allow it to give and follow directions, and would give it valuable additional information in aiding localization and navigation. We approach the problem by defining an ontology of space (including corridor, doorway, and room) and by creating a Convolutional Neural Network (CNN) that allows the robot to classify LIDAR sensor data accordingly. In particular, we propose a CNN architecture that performs comparably or better than existing methods based on engineered features. Training CNNs can be fickle; we describe several specific aspects of our approach that are important for good performance in this task.**

I. Introduction

Environmental structure can encode key semantic information about the type of place a robot is in. Corridors typically consist of long, narrow open spaces, whereas doorways appear as short, narrow gaps in structure. Robots can exploit semantic information to improve on or add to their capabilities.

For example, knowing that a robot has transitioned from a room into a corridor adds extra context for a localization problem. By pairing semantic place knowledge with natural language, a robot can be given verbal instructions for navigation, such as "Follow this corridor until you reach an intersection, then take the hall on the left." Semantic information can also be used to segment the environment facilitating the creation of topological maps.

Given the ubiquity of range sensors in modern robotics, there is value in exploring methods for extracting as much information from them as possible. Though it is unlikely that 2D range data on its own is sufficient to reliably distinguish high level concepts such as specific classes of room (kitchen, living room, etc.), it has proven useful for more general place recognition tasks [1], [2], learning to distinguish between classes such as "doorway", "corridor" and "room." Much of the previous work in place recognition focuses on carefully hand-engineering features for the task at hand, often deriving them from 2D or 3D range data.

We propose that CNNs are particularly well suited to the task of place classification based on 2D range data. Past methods have examined classification methods built around statistical features extracted from the raw range returns. However, it is standard practice to convert sets of range returns into occupancy grids for mapping and navigation tasks. Occupancy grids are analogous to grayscale images describing local structure. We hypothesize that, just as CNNs are adept at learning task appropriate features for classifying objects in images, they will be similarly adept at identifying features in occupancy grid "images" relevant to classifying place types.

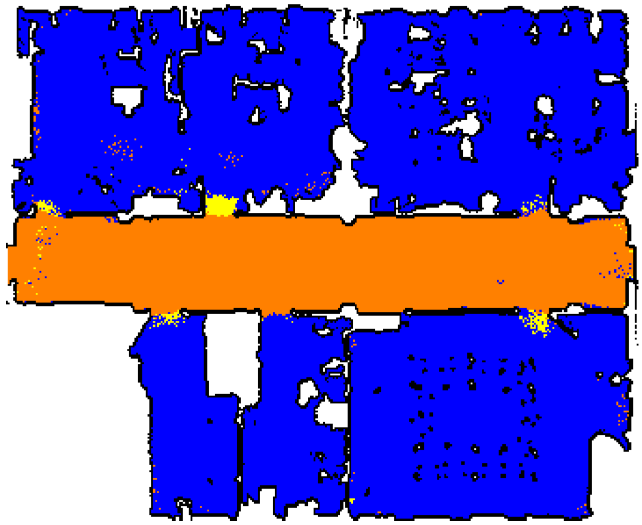


Fig. 1: CNN-based classification results on the fr79 dataset. Blue denotes a room, orange a corridor, and yellow a doorway.

This approach can be used for general map annotation tasks, as seen in Fig. 1, and is also well suited for processing live data streams from robots, as might be desirable when following directions in an unknown environment.

In this work, we demonstrate that CNNs can, in fact, be used effectively to apply semantic labels to places based solely on 2D range data. In particular, on the classes "room" and "corridor", we obtain accuracy above 92%. We also analyze the strengths and weaknesses of the system. Our contributions in the work include:

- We propose and evaluate a specific network structure to improve performance with regards to inter-class confusion.
- We experimentally demonstrate that CNNs perform comparably to or better than previous work based on a well-known dataset.

II. Related Work

Semantic place categorization adds information about structure to robotic systems in addition to providing grounded place labels in aiding human interaction with robots. It has been of particular interest to the topological mapping community, since the identification and recognition

This work was supported by DARPA D13AP00059.

The authors are with the Department of Computer Science and Engineering, The University of Michigan, Ann Arbor, MI {rgoeddel,ebolson}@umich.edu

of distinct places is critical to the functionality of topologically based systems. We broadly cluster approaches into two main groupings based on application: online labeling systems and map post-processing systems.

A. Online Labeling Systems

As the name implies, online labeling systems produce semantic place labels based on live sensor data from the robot. For example, one might combine range and camera data from a robot to attempt to classify the location the robot is currently in. These methods are well suited to tasks such as trajectory labeling or, more generally, operating in unknown environments.

For example, Mozos et al. present a place classification system using AdaBoost to distinguish places based on 2D range data [2]. The authors hand-construct a number of features describing the range data, allowing them to train a classifier to distinguish between rooms, corridors, doorways, and halls. Sousa et al., operating on a similar class of geometric features, show some success applying support vector machines (SVMs) to place classification, as well [3].

Shi et al. demonstrate a method for applying logistic regression to the same dataset and features as Mozos et al [4]. An interesting twist to this method is its ability to determine place categories for individual beams in a given scan, aiding in overall classification results.

Other recent works have investigated the use of 3D range and intensity data in addition to visual features acquired from a camera to distinguish between more specific place classes such as offices and kitchens [5], [6]. The rich features provided by these additional sensors allow high level knowledge (such as types of objects in a scene) to contribute to the semantic labeling process.

Hellbach et al. propose using non-negative matrix factorization on occupancy grid data to perform feature discovery [7]. They are able to achieve high rates of classification accuracy when using this as input for generalized learning vector quantization.

Online methods are not limited to range data. In addition to distinguishing spaces based on structural features, spaces may also be distinguished by the presence of particular objects in them. Several methods examine the process of building up a semantic hierarchy of spaces based on the presence of various objects in the scene [8], [9]. For example, a cluster of chairs around a table may be learned to denote a meeting space, while a table covered with books and a computer is a work space. Simultaneous occurrence of both suggests presence in an office. Models built based on relationships between objects in a scene have proven effective at identifying environmental context. These works are driven primarily by vision-based object recognition systems whereas the focus of this work is on the information encoded in 2D structural data.

Much like the works above, the method introduced in this paper is intended for use as an online method. Live occupancy grid data produced by the robot may be fed into the system, which returns a semantic label. The key insight in this paper is that CNNs, commonly used in the image classification domain, can also be applied effectively to occupancy grid data, eliminating the challenges of hand-designing features.

B. Map post-processing Systems

A variety of applications focus on post-processing metric maps to produce semantically annotated versions of the map, topological representations of the space, or both. They may employ strategies similar to online-labeling systems, or even directly use the output of such systems, but may additionally make use of spatial relationships between locations to help constrain the problem.

Beeson et al. employed extended Voronoi graphs (EVGs) constructed from 2D occupancy grids to identify places for the purpose of topological map building [1]. In this context, places can be distinguished based on the presence of various numbers of “gateways” and “path fragments” defined by the EVG. This can also provide some general discriminative ability about the types of places. For example, an intersection can only exist when there are multiple path fragments.

Friedman et al. introduce the Voronoi random field (VRF) which uses a Voronoi graph to generate a conditional random field (CRF), thereby incorporating spatial relationships into the classifier [10]. For example, labels will generally be locally consistent with their neighbors. Like Mozos et al., they also use AdaBoost to learn a classifier for different place types, later using the results from the AdaBoost classifier in the CRF. By taking advantage of the connectivity features, they are able to improve labeling consistency in addition to providing useful segmentations of the environment, which can be used to extract a topological representation of the space.

Mozos et al. follow up the original AdaBoost work by examining how further processing may improve the performance of the original system [11]. By applying some heuristics to correct initial labeling results, the authors are then able to segment the environment by region to produce largely accurate topological maps of several environments.

Additionally, it has been shown that associative Markov networks (AMNs) can be useful in improving classification results in the context of map-annotation [12], [11]. AMNs take advantage of the fact that labels are spatially correlated to produce improved classification results. For example, moving 10 cm often does not change the type of location that the robot is in.

Our proposed method could be used as input to many of these systems. However, we do not investigate data post-processing methods for improving our results in this paper.

III. Place Recognition Targeting Interactive Systems

Our target application for the semantic labels produced by our system is an interactive robot in an indoor environment. Interactions include instructing the robot to perform tasks such as delivery or finding an open meeting room, or the reverse, as when a robot describes a scene to the human. The

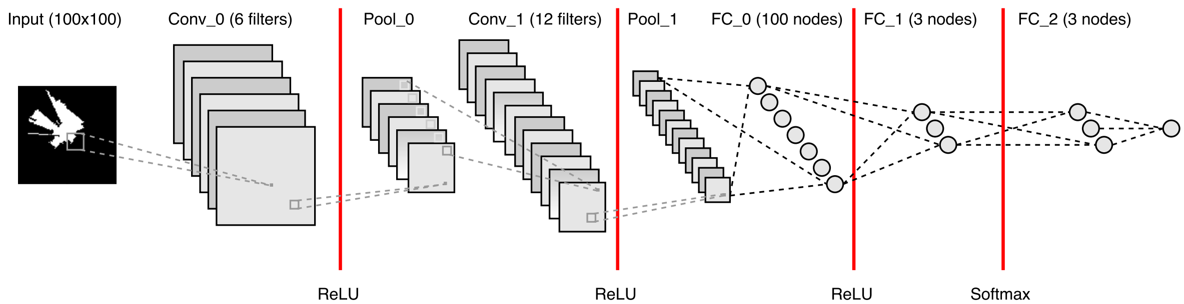


Fig. 2: The input gridmap is fed through several convolution/pooling layers before the results are classified by a multilayer perceptron.

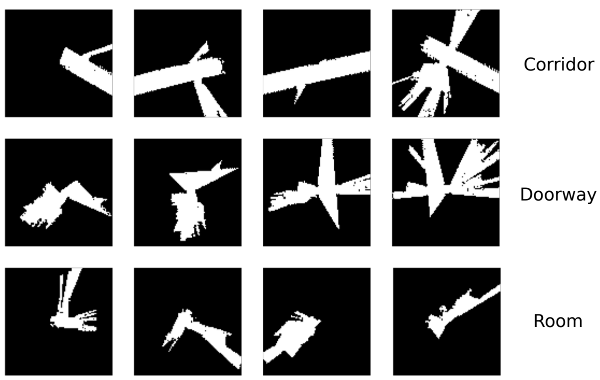


Fig. 3: Example occupancy grids created from single LIDAR scans for three place classes. These serve as input to the CNN. Black un-carved space, while white denotes carved free space. We assume un-carved by default, carving out free space based on range returns from our 2D LIDAR. This allows observations falling outside the fixed range of the occupancy grid add information to the image.

environment may not be known in advance, so it is important that the robot can produce reasonable classifications in unknown domains from as little as a single viewpoint. For this reason, we only use live sensor data as input to the classification system (e.g. range measurements and pose). The system should also be robust to viewing angle, producing consistent labeling for the same XY location regardless of robot orientation.

The input is a fixed-size occupancy grid centered on the pose of the robot such that the robot is facing down the X-axis. As can be seen in Fig. 3, our occupancy grids are created by carving out free space, as opposed to a more traditional format in which all space is assumed free until structure is observed. To eliminate "striping" in the maps caused by gaps between LIDAR strikes, we also carve out space between consecutive strikes by interpolating between them.

If space carving were not employed, range returns falling outside the range of our fixed-size occupancy grids are lost, adding no information to the system. Space carving allows us to retain some information about distant range returns, even if our local occupancy grid is small.

One hazard of this representation is that of scale; e.g. corridors come in many shapes and sizes. Without sufficient training data across scales, however, the CNN is unlikely to capture the important scale-invariant features of some classes. We incorporate a scaling step (detailed further below) into our training procedure to mitigate this issue.

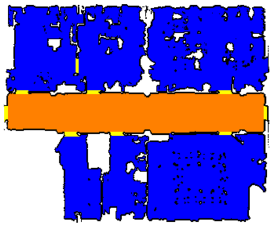
A. Classifying Places With CNNs

We hypothesize that the features relevant to distinguishing classes of places in occupancy grids are similar to those learned by the CNNs employed in image-based object classification tasks, therefore CNNs will also prove useful in this domain. CNNs are known to learn increasingly complex and specific features by combining results from previous convolutional layers [13]. While CNNs containing upwards of 20 layers can be found in the literature [14], we achieved good performance with relatively small networks.

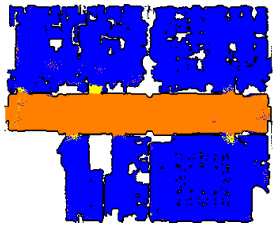
We employ a network structure loosely based on LeNet-5 [15]. Input is fed through a stacked pair of {convolution, ReLU, mean pooling layer} groupings. The feature maps are then fed into a fully connected ReLU layer before being collapsed into a probability distribution-like output across class labels by a fully connected Softmax layer. One final fully connected layer is applied to this distribution to produce a single label as the classifier's output. Our network structure can be seen in more detail in Fig. 2.

The final fully-connected layer after the softmax plays a critical role in classification. Using the LeNet-5 network structure, the penultimate softmax layer outputs were "scene contains a" detectors. The input occupancy grids often capture several classes in one image, especially in the case of doorways, from which both a room and a corridor may often be visible. As a result, it is not uncommon to see a situation when both the room-detecting and doorway-detecting portions of the network activate simultaneously.

This confusion often manifests in a split probability distribution, where the system is split between two or more classes. Perhaps because doorways are small compared to the other classes, this signal in tends to be overwhelmed by the nearby presence of rooms and corridors. The presence of this confusion can actually be a source of information, though. For example, the presence of simultaneous corridor, room, and doorway signals may, in fact, be indicative the



(a) fr79 test – hand-labeled

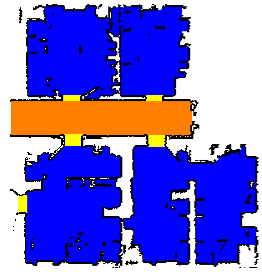


(b) fr79 test – CNN

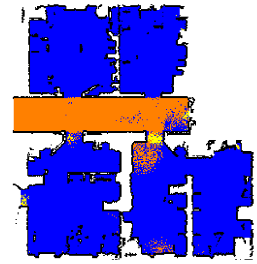
Fig. 4: Hand-labeled test data for the fr79 dataset (a) compared to labels produced by our CNN (b). The other half of building fr79 was used to train the CNN, while training data from fr52 was used as a validation set to prevent overfitting.

	Room	Corridor	Doorway
Room	98.3%	1.1%	0.6%
Corridor	0.7%	99.1%	0.2%
Doorway	24.1%	60.4%	15.5%
Overall Accuracy			97.8%

TABLE I: Classification confusion matrix, with the true values listed by row, for fr79 test. Our method performs well on room and corridor classes, but underperforms on doorways.



(a) fr52 test – hand-labeled



(b) fr52 test – CNN

Fig. 5: Hand-labeled test data for the fr52 dataset (a) compared to labels produced by our CNN (b). The other half of building fr52 was used to train the CNN, while training data from fr79 was used as a validation set to prevent overfitting.

	Room	Corridor	Doorway
Room	97.8%	2.1%	0.1%
Corridor	7.4%	92.3%	0.4%
Doorway	51.7%	32.4%	16.9%
Overall Accuracy			95.3%

TABLE II: Classification confusion matrix, with the true values listed by row, for fr52 test. Accuracy on corridors decreases compared to that seen in fr79 test. This can likely be overfitting the model to the limited training data. In particular, note the difference in structure between the training example of a dead-end in a hallway compared to that seen in the test set, where the majority of corridor errors can be seen.

doorway is the true class label, as this confusion only occurs when the robot is in a doorway.

The last fully-connected layer is designed to capture information based on these patterns in the final probability distribution. For example, the layer can override a strong room signal in favor of a weaker, but significant, doorway signal. Experimentally, we found that adding this final fully connected layer resulted in a nearly 33% increase in correct doorway classifications.

B. Training Procedure

Neural networks are notoriously difficult to train [16]. The power to discover relevant discriminatory features based on the data also leads to a well known tendency to overfit the training set. To mitigate the problem of overfitting, we incorporate a validation set into the training procedure. At the end of each epoch, the current network is tested against the validation set. If classification performance has improved, the current network parameters are saved, and training continues. If, after a certain number of epochs, no improvement has been seen, we revert to the best parameters based on the validation set and terminate the training procedure. Final test results are then produced from a third, independent test set.

To further increase the generality of the features learned, we address the impact of potential biases or limitations in the training data. For example, corridors come in a variety of widths, but if the training set only contains one example corridor of a fixed width, this information is lost. To augment the variety in training examples, we randomly apply X and Y scale transformations between 85% and 115% to training examples during runtime. The dataset used in this paper already contains a wide variety of randomly oriented viewpoints, so we found that introducing further random rotations to the data did not noticeably change performance.

If there are widely imbalanced numbers of training examples for each class (as there are in our datasets, since doors and even corridors take up much less space than rooms), the network can be biased towards detecting the more prevalent classes. For example, doorways account for only 2.3% of labeled examples in one of our training sets. Unsurprisingly, this causes the resulting CNN to overwhelmingly bias results towards the dominant class, “room.”

One method for addressing this phenomenon is random reselection of training examples from the smaller pools of class examples, also known as oversampling [17]. This operation may be repeated during a training epoch until each example from the largest pool of class examples has been seen once. If we had 1000 training instances of rooms, 500 of corridors, and 100 of doorways, each room example would be presented once, each corridor twice, and each doorway ten times. In conjunction with our runtime data augmentation, this increases the number of “unique” class examples. We found that this mitigated the impact of class imbalance for our dataset. For multi-threading purposes, data may also be split into mini-batches in a similar manner.

Additional strategies for avoiding overfitting include incorporating dropout and/or regularization to the network during training. These methods can force the network to use a more diverse array of its possible inputs, rather than overwhelmingly preferring a few, high-weight inputs. We found that incorporating dropout had erratic effects depending on the training data used, sometimes dramatically improving results on the doorway class, for example, other times significantly worsening it. As a result, we present results without dropout, which were more consistent overall.

IV. Results

We evaluate our system on the dataset[1] used by Mozos et al. in [2]. The dataset consists of thousands of hand-labeled, 360 degree LIDAR scans generated from maps of three separate buildings. Scans were gathered every 5 cm with random robot orientations. For each building, the scans are divided into a training half and a test half depending on which side of the building they were collected from. Mozos et al. present results for each building separately, training their algorithm on the training half and evaluating on the test half. We present building-specific results for our method based on the same training/test partitions, in addition to cross-building results.

We present results for buildings fr52 and fr79, which each contain examples of the classes room, corridor, and doorway. An example composite image of hand-labeled training scans from the fr79 dataset can be seen in Fig. 4a.

We use an in-house neural network library written in C to construct our CNN. Training was performed without GPU acceleration on a machine containing 32 GB of RAM and 2 2.5 GHz Intel Xeon processors, giving a total of 24 hyper-threaded cores. Training time typically took between 8 and 24 hours, depending on the structure and parameterization of the network.

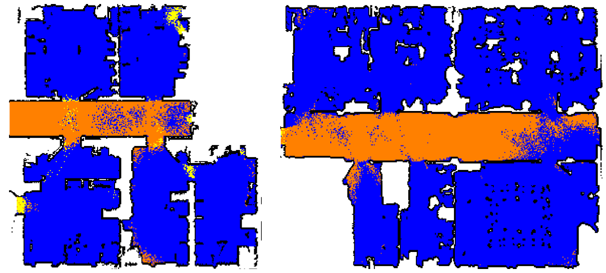
A. Choosing Network Parameters

When training a CNN, the choice of the size of the input and size and number of hidden nodes can impact network performance. We limited our input to 10 cm occupancy grids, which balanced capturing large areas while preserving fine-scale structure needed for doorways.

Experiments were performed for images capturing 4×4 m through 10×10 m regions surrounding the robot. We found that selecting smaller regions often improved performance on doorways, but at the cost of distinguishing classes such as corridors and rooms. This result can intuitively be linked back to the scale of the places in question. Doorways occupy small areas, thus small occupancy grids filter out distracting information. Conversely, many parts of a corridor or room may only be adequately captured at a larger scale. As a result, we opted to focus on the largest, 10×10 m occupancy grids, as they seemed best suited for capturing relevant information about all classes.

Based on this input image size, convolutional filter sizes were set to 9×9 pixels. Additional filter sizes were not explored, as these settings seemed to work well. Six first-level features proved sufficient to capture low-level details in the occupancy grids. Variation in the number of second-level filters had only marginal impact on the results. Very large and very small numbers of hidden nodes in the fully-connected layer resulted in noisy results, but we found that using between 50 and 500 nodes produced similar levels of performance. Accuracy varied by less than 1% between the three network structures. We present results for 100 hidden nodes, as this resulted in the best overall performance across classes.

[1] http://www2.informatik.uni-freiburg.de/~omartine/place_data_sets.html



(a) fr79 CNN on fr52 test (b) fr52 CNN on fr79 test

Fig. 6: Test results for fr52 and fr79 based on training data from the other building. Though overall accuracy still exceeds 90%, performance decreases, likely due to overfitting building specific characteristics.

	Room	Corridor	Doorway
Room	97.0%	1.8%	1.2%
Corridor	16.6%	81.2%	2.2%
Doorway	29.7%	47.3%	23.0%
Overall Accuracy			93.2%

TABLE III: Percentage of proposed labels (column) compared to the true label (row) for fr52 test data, trained on fr79.

	Room	Corridor	Doorway
Room	98.8%	1.1%	0.0%
Corridor	24.4%	75.6%	0.0%
Doorway	73.9%	20.8%	5.3%
Overall Accuracy			92.4%

TABLE IV: Percentage of proposed labels (column) compared to the true label (row) for fr79 test data, trained on fr52.

With our implementation of this architecture, we are able to produce a classification of a single 100×100 occupancy grid in 7 ms on a consumer grade desktop equipped with an Intel i7-2600K processor clocked at 3.4 GHz and 8 GB of RAM.

B. Map Annotation

Though our system is designed for use online, it is easy to demonstrate its performance across viewpoints in an environment by using it in a map annotation task. We train two networks, one using the fr79 training set, validated against fr52 training set during training, and the other trained by reversing the sets. Our network structure can be seen in Fig. 2. Then, both networks are evaluated against the fr52 and fr79 test sets.

Classification results trained and tested on fr79 can be seen in Fig. 4, while results tested on fr52 can be seen in Fig. 5. Hand-labeled training sets were used for training and validation, while test sets were reserved for evaluation. Results generated by the CNN classifier are visualized for both the test and the training sets.

Accuracy on both test sets are comparable to or better than those seen in the literature, achieving 97.8% accuracy on fr79 and 95.3% accuracy on fr52 (see Table I and Table II for class specific breakdowns). In comparison, Mozos et al. report 93.4% accuracy on fr79 and 92.1% on fr52 with their sequential AdaBoost method [2]. Labels show large degrees of spatial consistency, in particular for rooms and corridors, although there is some speckled noise.

Doorways prove challenging to effectively classify. As documented in Sec. III-A, this can be attributed to the fact that views from doorways inherently contain other classes, as well as the small amount of training data generated by a class occupying such a small area. On the fr79 dataset, we are only able to achieve 15.5% accuracy on doorways in the test set. Mozos et al. only report training error for individual classes, but they, too, report the highest individual class error was seen for doorways.

Further investigation of the failures shows that a large portion of doorway detections register just beyond the bounds of the labeled regions. Given the subjective nature of the doorway annotations in the training and test data, these results could still be quite serviceable in practical applications. Additional training instances of doorways would also present better opportunity for learning general models.

C. Performance on New Environments

In this section, we examine how the classifiers trained in Sec. IV-B perform in new environments from which no training data has been seen. We would like to know if the internal representations of classes being learned are general enough to be used in novel domains. To this end, we evaluate the fr79 classifier on the fr52 test set and the fr52 classifier on the fr79 test set. Annotated map data for both tests can be seen in Fig. 6.

Overall accuracy on both test sets remains high, but does decrease compared to when a classifier trained on the other half of the same building is used. This is likely due to overfitting building specific characteristics. For example, the end of the corridor in the fr52 training data is very different from that in the fr79 test data. As a result, this region of fr79 is mostly misclassified as "room." Likewise, the doorways in both training environments are largely dissimilar from those in the test environments, making them even more challenging to learn.

The most challenging aspect of this dataset is the lack of diversity of "places" (e.g. specific instances of doorways). For example, though the variety of viewpoints provides robustness to orientation, there are 10 or fewer examples of doorways in either of the training sets. As a result, it is difficult to learn a model of doorway that generalizes well to new environments.

V. Conclusion

In this paper, we have demonstrated that CNNs can be used to great effect in learning semantic place labels from 2D range data. Results generalize well between environments, but could be improved by increased variety in the training data. Nonetheless, our method performs as well or better than existing work.

References

[1] P. Beeson, N. K. Jong, and B. Kuipers, "Towards autonomous topological place detection using the extended voronoi graph," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 4373–4379.

[2] O. M. Mozos, C. Stachniss, and W. Burgard, "Supervised learning of places from range data using adaboost," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 1730–1735.

[3] P. Sousa, R. Araújo, and U. Nunes, "Real-time labeling of places using support vector machines," in *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*. IEEE, 2007, pp. 2022–2027.

[4] L. Shi, S. Kodagoda, and G. Dissanayake, "Laser range data based semantic labeling of places," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 5941–5946.

[5] A. Pronobis, O. M. Mozos, B. Caputo, and P. Jensfelt, "Multimodal semantic place classification," *The International Journal of Robotics Research (IJRR), Special Issue on Robotic Vision*, vol. 29, no. 2-3, pp. 298–320, Feb. 2010. [Online]. Available: http://www.pronobis.pro/publications/pronobis2010ijrr

[6] O. M. Mozos, H. Mizutani, H. Jung, R. Kurazume, and T. Hasegawa, "Categorization of indoor places by combining local binary pattern histograms of range and reflectance data from laser range finders," *Advanced Robotics*, vol. 27, no. 18, pp. 1455–1464, 2013.

[7] S. Hellbach, M. Himstedt, F. Bahrmann, M. Riedel, T. Villmann, and H.-J. Böhme, "Some room for glvq: Semantic labeling of occupancy grid maps," in *Advances in Self-Organizing Maps and Learning Vector Quantization*. Springer, 2014, pp. 133–143.

[8] S. Vasudevan and R. Siegwart, "Bayesian space conceptualization and place classification for semantic maps in mobile robotics," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 522–537, 2008.

[9] P. Viswanathan, D. Meger, T. Southey, J. J. Little, and A. Mackworth, "Automated spatial-semantic modeling with applications to place labeling and informed search," in *Computer and Robot Vision, 2009. CRV'09. Canadian Conference on*. IEEE, 2009, pp. 284–291.

[10] S. Friedman, H. Pasula, and D. Fox, "Voronoi random fields: Extracting topological structure of indoor environments via place labeling." in *IJCAI*, vol. 7, 2007, pp. 2109–2114.

[11] O. M. Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard, "Supervised semantic labeling of places using information extracted from sensor data," *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 391–402, 2007.

[12] R. Triebel, R. Schmidt, Ó. M. Mozos, and W. Burgard, "Instance-based amn classification for improved object recognition in 2d and 3d laser range data," in *Proceedings of the 20th international joint conference on Artifical intelligence*. Morgan Kaufmann Publishers Inc., 2007, pp. 2225–2230.

[13] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 818–833.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014. [Online]. Available: http://arxiv.org/abs/1409.4842

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[16] J. V. Tu, "Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes," *Journal of clinical epidemiology*, vol. 49, no. 11, pp. 1225–1231, 1996.

[17] M. A. Maloof, "Learning when data sets are imbalanced and when costs are unequal and unknown," in *ICML-2003 workshop on learning from imbalanced data sets II*, vol. 2, 2003, pp. 2–1.