Convolutional Neural Network Information Fusion based on Dempster-Shafer Theory for Urban Scene Understanding

Masha (Mikhal) Itkina and Mykel John Kochenderfer *
Stanford University
450 Serra Mall, Stanford, CA 94305
{mitkina, mykel}@stanford.edu

Abstract

Dempster-Shafer theory provides a sensor fusion framework that autonomously accounts for obstacle occlusion in dynamic, urban environments. However, to discern static and moving obstacles, the Dempster-Shafer approach requires manual tuning of parameters dependent on the situation and sensor types. The proposed methodology utilizes a deep fully convolutional neural network to improve the robust performance of the information fusion algorithm in distinguishing static and moving obstacles from navigable space. The image-like spatial structure of probabilistic occupancy allows a semantic segmentation framework to discern classes for individual grid cells. A subset of the KITTI LIDAR tracking dataset in combination with semantic map data was used for the information fusion task. The probabilistic occupancy grid output of the Dempster-Shafer information fusion algorithm was provided as input to the neural network. The network then learned an offset from the original DST result to improve semantic labeling performance. The proposed framework outperformed the baseline approach in the mean intersection over union metric reaching 0.546 *and* 0.531 *in the validation and test sets respectively. However, little improvement was achieved in discerning moving and static cells due to the limited dataset size. To improve model performance in future work, the dataset will be expanded to facilitate more effective learning, and temporal data will be fed through individual convolutional networks prior to being merged in channels as input to the main network.*

1. Introduction

Autonomously accounting for obstacle occlusion is an open problem for self-driving cars. Human drivers can anticipate possible hazards in blind spots caused by lack of visibility. A human driver can infer that a person standing by the road may begin moving or that a parked car may pull out onto the road. An autonomous vehicle should have the capability for similar logic and reactions. Dempster-Shafer Theory (DST) provides a decision-making strategy that addresses occlusion by modeling both lack of information and conflicting information directly [14]. DST can combine sensor information subject to uncertainty with semantic scene information obtained from a street-level digital map as in [14]. Sensor and digital map occupancy grids are fused to discern grid cells that contain potential hazards (both mobile and stationary) from cells that are navigable by the vehicle. This information is stored in a holistic perception grid, which allows for the perception system to anticipate areas where occluded hazards may appear. However, the approach heavily relies on several parameters that require manual tuning specific to the situation in order to achieve desired behavior in detecting static and moving obstacles [14].

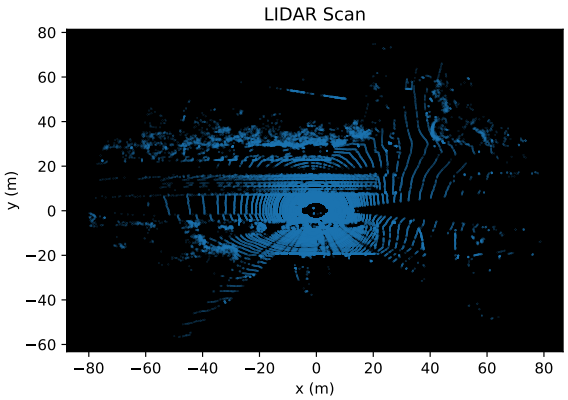


Figure 1: Example of LIDAR HDL-64E scan (top-down view) prior to pre-processing into an occupancy grid [6].

The proposed approach merges the semantic segmenta-

*Prof. Mykel Kochenderfer is the faculty advisor for this project in the Deparment of Aeronautics and Astronautics at Stanford University.

tion framework in [15] using a fully convolutional neural network (FCN) with the DST information fusion algorithm presented in [14] to increase the latter's robustness in discerning occupancy grid cells containing static and moving objects from navigable space. The inputs to the baseline DST algorithm in [14] are a LIDAR sensor grid containing LIDAR data, a geographic information system (GIS) grid containing semantic map data, and probabilistic occupancy grids which form the perception grid outputted by DST at the previous time-step. The input to the FCN is the set of probabilistic perception grids generated by the DST algorithm at the current and previous time-steps stacked in channels. The network outputs the updated perception grid for the current time-step, which is a cell-by-cell classification of the local grid according to its semantic segmentation as described in Section 4.

2. Related Work

A perception framework commonly depends on an occupancy grid built in 2-D, 2.5-D, or 3-D space [14, 21, 2]. This paper will focus on approaches dealing with 2-D occupancy grids due to their similarity in spatial structure to images, allowing for direct applicability of existing deep learning algorithms. One approach to scene understanding and sensor fusion employs DST as proposed in [14]. Kurdej et al. focus on the benefits of combining evidence in the form of an existing digital street-level maps and sensor data to naturally handle occlusion. A digital map occupancy grid and a sensor occupancy grid are combined to make decisions using DST as to which class a grid cell belongs to in a set of hypotheses (e.g. static, moving, infrastructure, etc.) thus forming a perception grid [14]. Kurdej et al do not cluster the grid cells into objects, in contrast to some Bayesian approaches as in [8], but rather facilitate perception based on classified grid cell information. The drawback to the algorithm proposed in [14] is that the approach relies on several parameters that require manual tuning to achieve desired behavior. For instance, the discounting factor determines how quickly information is discarded. The algorithm also relies on gains and increment/decrement step sizes that determine the speed with which a decision is made that an object is categorized as moving or static [14]. Manually tuning these parameters is not a robust solution since better optimization performance could be achieved algorithmically. Similarly to [14], [23] utilizes DST to fuse information from several sensors in order to perform obstacle detection. In [23], sensor information is discounted based on associations to obstacles from different sensor types, which leads to a biasing of the obstacle detections to more accurate sensor data. The requirements are also loosened on occupancy grid cell independence in [23] as compared to [14].

Recently, several works have investigated convolutional neural networks (CNN) as a direct means to perform sensor fusion. In [18], the authors fuse data from stereo cameras with a 6-layer FCN framework to predict a disparity map utilizing the KITTI [6] dataset for training. The resulting algorithm is robust to obstacle occlusion. In [5], RGB and depth information was passed through a two-stream CNN separately to successfully perform object recognition. The two streams were unified with fully connected layers. DST has previously been used in perception as a pre-processing information fusion step to a CNN to achieve both semantic image labeling as in [25] and object detection and classification as in [16]. [25] presents a custom, 4-layer CNN, while [16] utilizes a pre-trained VGG-16 network for each sensor.

There have also been some recent work in scene segmentation utilizing LIDAR occupancy grids and deep learning. Since LIDAR datasets have started to emerge for public use only recently, utilizing deep learning techniques on LIDAR data is an active area of research. LIDAR 2-D occupancy grids provide a parallel with pixel-image data, since both are a 2-D representation of spatial information that can be stacked into channels. [22] investigates some common CNN architectures pre-trained on the ImageNet dataset such as AlexNet, GoogLeNet, VGG-16 to classify cells into road types. [22] determined that using networks pre-trained on images was advantageous as compared to training custom architectures from scratch. [7] utilizes LIDAR occupancy grids to discern hallways from rooms in a building with a 5-layer CNN architecture. [3] uses a deep FCN with 12 convolutional layers to provide semantic labels for the grid cells discerning the road from the rest of the environment. This algorithm outperforms the state-of-the-art on the KITTI dataset. The advantage of FCNs is the minimized number of parameters required and the ability to maintain the spatial representation of the input throughout training. [3] utilizes dilation to achieve a larger receptive field within the network, aiding in the segmentation task.

In this paper, the classical FCN image semantic segmentation approach proposed in [15] is merged with the information fusion algorithm presented in [14] to improve the performance of the DST algorithm in discerning occupancy grid cells containing both static and moving objects from navigable space. The generation and pre-processing of the dataset used to train and test the network is described in Section 3. The algorithmic approach to information fusion and segmentation utilizing DST and FCN is outlined in detail in Section 4, including the specific architecture of the FCN. The experimental results are presented and discussed in Sections 5 and 6.

3. Dataset and Features

The KITTI tracking dataset [6] was augmented for use in information fusion as per Kurdej's framework in [14]. Four driving sequences were chosen for training (140 examples),

two for validation (48 examples) and two for testing (64 examples). The augmented dataset consists of a GIS grid containing the semantic map information, a sensor grid containing HDL-64E Velodyne LIDAR data, and the labeled perception grid segmentation. A sample of the raw LIDAR data prior to processing into an occupancy grid is shown in Figure 1. Each grid is created for a single ego vehicle GPS coordinate which is obtained either every 1 s or 2 s depending on the driving sequence [6]. The grids have dimensions of $85.4m \times 85.4m$, with the ego vehicle in the center. Given a discretization of $0.33m$ per grid cell, each grid is of size 256×256 cells.

The data for the GIS grids was obtained from the OpenStreetMap database and processed with the QGIS software [17, 19]. Each grid cell is categorized into the classes: building, road, or intermediate space. The map is assumed to have high accuracy, although there is evidence that localizing with OpenStreetMap and GPS alone is not sufficient [24].

The data from the HDL-64E Velodyne LIDAR obtained as part of the KITTI tracking dataset was used to create the sensor occupancy grids [6]. The grids are categorized into free, occupied, and unknown space. A simple form of ray-tracing is performed where all space between a measurement and the physical sensor is considered free. In order to classify road measurements as 'free space', the RANSAC algorithm was used to segment out the estimation to the ground plane as part of data pre-prossessing [1].

The perception grid classifies each grid cell into five classes: navigable, non-navigable, moving, stopped, or building. Objects within the KITTI tracking dataset were classified as 'moving' if their global location with respect to the first obtained GPS point in a driving sequence changed from one measurement to the next by a distance of more than $10cm$ [2].

Examples of GIS and sensor grids, as well as their corresponding perception grid is shown in Figures 2, 3, and 4. The dataset contains an imbalanced class distribution with only 0.09% static cells and 0.26% moving cells within the training set.

4. Methods

This section is organized as follows: the DST framework is briefly described in Section 4.1 followed by the outline of the FCN architecture and the loss for optimization in Section 4.2.

[1]The RANSAC algorithm used: https://github.com/falcondai/py-ransac.

[2]Conversion to global coordinates from GPS coordinates: https://github.com/utiasSTARS/pykitti.

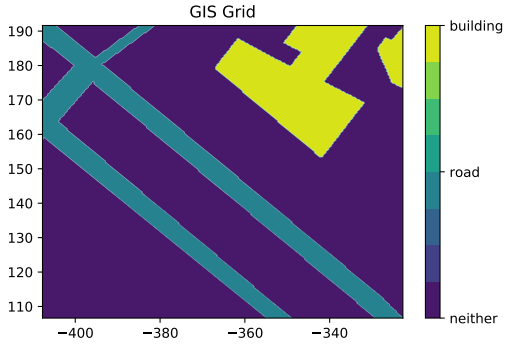


Figure 2: Training data example of a GIS occupancy grid plotted as a contour plot.

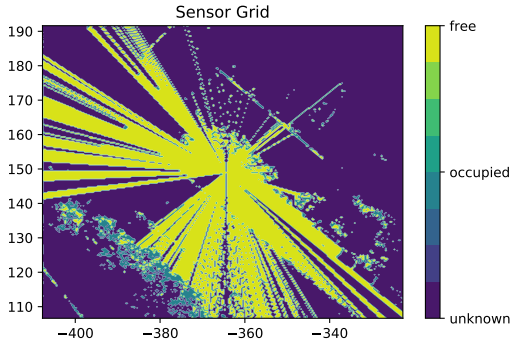


Figure 3: Training data example of a lidar occupancy grid plotted as a contour plot.

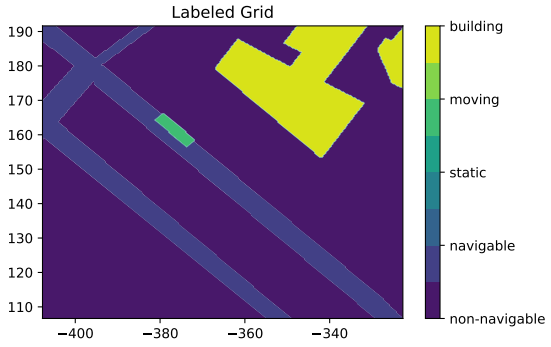


Figure 4: Training data example label of a perception grid plotted as a contour plot. The figure shows a moving truck approaching an intersection.

4.1. DST Information Fusion

The algorithm proposed in [13, 14] is chosen as the baseline comparison for the proposed FCN-DST information fu-

sion approach. DST takes as input the current sensor grid and GIS grid as well as the perception grid at the previous time-step. The algorithm combines the information utilizing a Dempster-Shafer combination rule to produce an updated perception grid. DST works with belief masses associated with sets of events rather than probabilities of singleton events. DST can directly model lack of information by assigning mass to the set of all possible events. These masses can then be converted to traditional probabilities using the concept of pignistic probability. Further details on the information fusion procedure are provided in the Appendix.

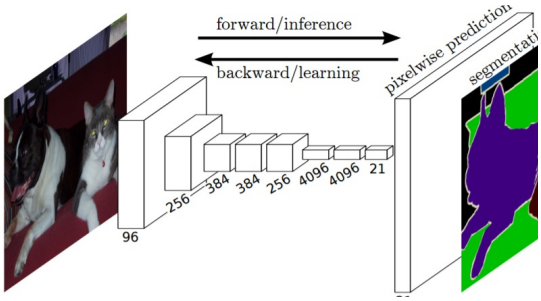


Figure 5: FCN architecture presented in [15].

4.2. FCN-DST Information Fusion

The grid sensitivity of the DST framework is optimized for segmentation performance by passing its output through an FCN. The architecture of the FCN [3] is based on the model presented in [15] as shown in Figure 5. An FCN consists of only convolutional layers, maintaining the spatial information for segmentation. In [15], the architecture begins with the 16 convolutional layers and ReLU activations of a VGG-19 network pre-trained on images, interspersed with pooling layers. These are followed by 3 de-convolutional layers separated by dropout for regularization, and then 2 more convolutional layers, ending in a softmax layer. Due to the small dataset size, it was imperative to initialize the VGG layers with pre-trained weights. Nevertheless, the entire architecture was trained on the dataset as the nature of LIDAR data is substantially different from that of RGB images.

To create compatibility between the occupancy grid dataset considered in this paper and the FCN architecture, several minor adjustments were made. The depth of the last set of layers was changed from 21 to 5 to accommodate the number of classes in the perception grid labels. The input to the FCN is the set of probabilistic perception grids generated by the DST algorithm at the current and previous time-step stacked in channels (10 channels in total). The perception grid updates within DST are accumulated over time; hence, the previous time-step output contains the accumulated time-history data. By providing the current and previous DST perception grids as inputs, the FCN should have sufficient information to learn the temporal and spatial information necessary to classify moving and static cells effectively. To make the 10-channel DST occupancy grids compatible with the VGG network, which expects RGB images as inputs, an additional convolutional layer was added at the start of the architecture to reduce the input channel number to three.

The objective of a deep neural network is commonly taken as the cross-entropy loss. Since the segmentation output is of occupancy grid dimension, a modified cross-entropy loss is used, where the loss is averaged over all the cells in a grid. The loss is also weighted to resolve the class imbalance. The loss equation is as follows [15, 9]:

$$\text{loss} = \frac{1}{N \times 256 \times 256} \sum_k \sum_i \sum_j -\log\Big(\frac{e^{f_{y_{k,i,j}}}}{\sum_l e^{f_l}}\Big) w[y_{k,i,j}], \tag{1}$$

where N is the batch size (k is the corresponding iterator), i, j sum over the spatial dimensions, f represents the softmax scores, and y is the correct class label. The weights w for each class are computed according to the formula introduced in [4]:

$$w_c = \text{median\_freq}/\text{freq}[c] \tag{2}$$

where *freq* is the number of times a class appears divided by the total number of pixels in images that contain the class within the training set.

5. Results

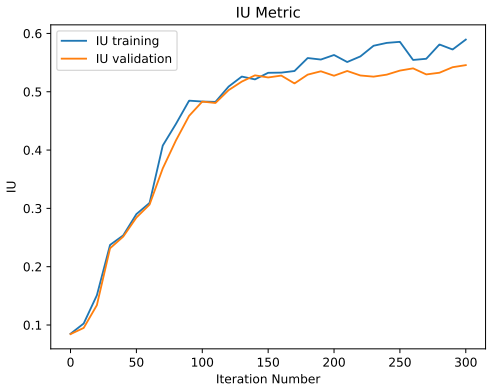
The FCN architecture described in Section 4.2 was implemented in the TensorFlow open-source framework [1]. The effectiveness of the approach is measured in reference to the DST baseline using the mean intersection over union (IU) metric over each of the classes. IU is often used for semantic segmentation to directly account for class imbalance. The IU metric is computed as follows:

$$\text{IU} = \frac{\text{TP}}{\text{FP} + \text{TP} + \text{FN}}. \tag{3}$$

The neural network parameters were tuned to optimize the IU metric. The batch size was chosen to be 32 to balance noise reduction in the loss update with reasonable computation time for each iteration (300 iterations took approximately 45 minutes to run on an NVIDIA GPU). The learning rate was optimized such that the loss was not decreasing too quickly at the start and not too slowly across iterations. The training loss curve is shown in Figure 6, which indicates that an appropriate learning rate was selected. In an

[3] Starter code for the FCN was obtained from: https://github.com/shelhamer/fcn.berkeleyvision.org.

(a)



(b)

Figure 6: (a) Loss profile over iterations. (b) IU metric profile over iterations.

attempt to prevent overfitting, the keep probability in the dropout layers during training was set to 0.1. Figure 6 depicts that overfitting was nevertheless still present, since the validation loss curve diverges from the training loss curve due to the limited dataset size. Table 1 summarizes the tuned hyper-parameters for the FCN.

learning rate	keep probability	batch size	filter size
1e-4	0.1	32	7

Table 1: Tuned model hyper-parameters based on IU metric.

To optimize the parameters within the FCN, the commonly used *Adam* algorithm is chosen [10]. *Adam* adapts learning rates to each parameter in the optimization, while employing the concept of momentum to arrive at a solution more efficiently [11]. The recommended hyper-parameters for the *Adam* optimizer were used: $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e-8$ [10].

Table 2 presents the results obtained for the IU metric on the validation and test sets. The tuned FCN architecture achieved a slightly higher mean IU than DST alone in both the test and validations sets of 0.546 and 0.531 respectively. Although the individual class IU values were higher (except for the 'building' class), the network still showed poor static and moving object detection performance. Figure 7 shows the confusion matrices for the classification task on the training, validation, and test sets. Although the training set confusion matrix shows favorable performance in predicting the static and moving classes (high values on the diagonal), the FCN does not generalize well in these categories on the validation and test sets. This is further shown in Table 3 in the accuracy, prediction, and recall metrics. Note that despite the high accuracy, overfitting is indicated by the lower precision and recall results in the validation and test sets. Figure 8 portrays the moving category probabilistic DST occupancy grid (a channel in the input to the FCN) alongside the predictions made by the network and the expected labels for an example within the validation set.

	FCN-DST val	DST val	FCN-DST test	DST test
Navigable	**0.895**	0.854	**0.839**	0.775
Non-Navigable	**0.931**	0.904	**0.923**	0.786
Building	0.882	**1.00**	0.903	**1.00**
Static	**0.00928**	0.00135	**0.000701**	0.000558
Moving	**0.0108**	0.0140	**0.00787**	0.00120
Mean	**0.546**	0.539	**0.531**	0.512

Table 2: Results table: IU metric values for each class in the validation and test sets.

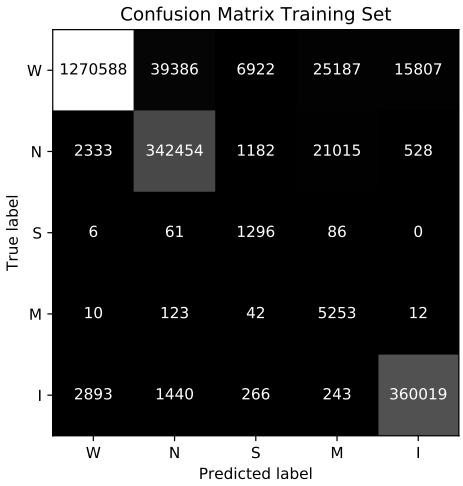
	Accuracy	Precision	Recall
FCN-DST train	**0.944**	**0.616**	**0.943**
DST train	0.888	0.599	0.598
FCN-DST val	**0.950**	0.569	0.592
DST val	0.851	**0.603**	**0.593**
FCN-DST test	**0.934**	0.433	0.471
DST test	0.827	**0.599**	**0.618**

Table 3: Average accuracy, precision, and recall values for the five classes in the training, validation, and test sets.

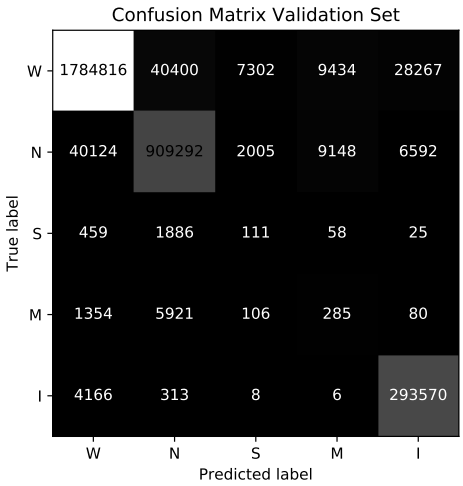
6. Discussion

The small dataset size of 140 training examples contributed to the relatively poor generalization performance in the 'static' and 'moving' classes observed in Table 2 and Figure 6. The deep FCN was able to overfit the training set, but the learned model was not sufficient to make effective predictions on the validation and test sets. The measurement frequency of 1 or 0.5 Hz may have been too low to effectively discern dynamic obstacles. Expanding the dataset to include higher frequency measurements would likely result in better generalization and classification performance.

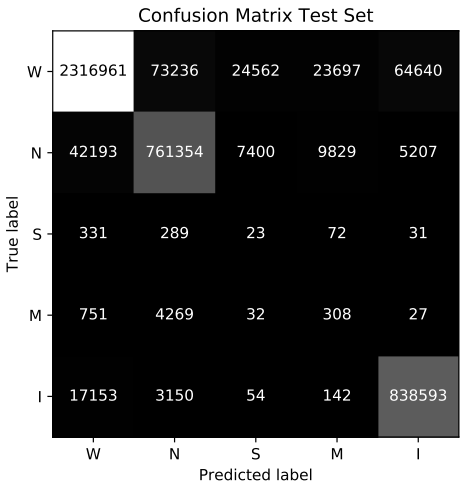
Prior to utilizing RANSAC plane fitting to filter out LIDAR points that returned occupancy measurements from



(a)

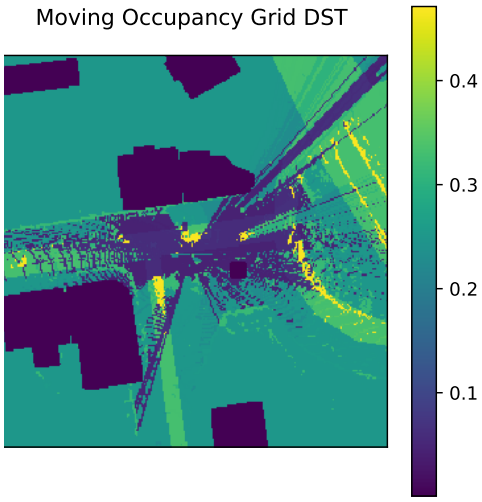


(b)

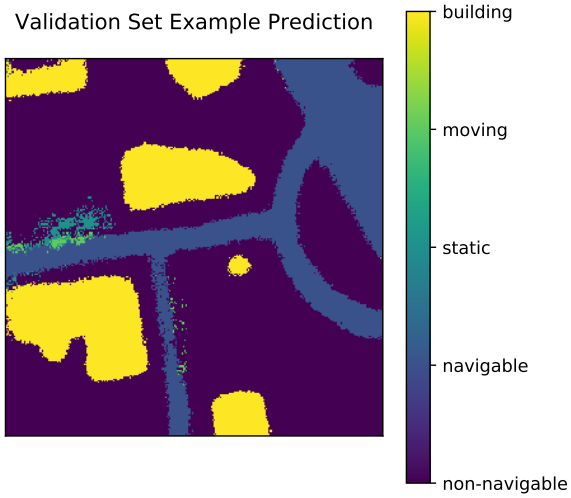


(c)

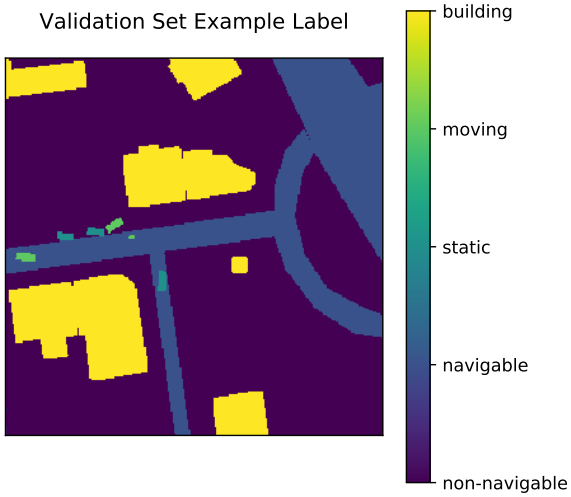
Figure 7: Confusion matrices for (a) the training set (batch of 32), (b) the validation set, and (c) the test set output of the FCN. The definition of the symbol labels on the axes is provided in the Appendix.



(a)



(b)



(c)

Figure 8: Examples from the validation set showing: (a) the moving occupancy grid generated by DST, (b) the predicted labels from the FCN, and (c) the expected labels.

the ground, hand annotated labels from [26] were used. The latter was not a robust solution as the labels were not exhaustive and left considerable free space marked as occupied. The labels also limited the dataset size. The RANSAC algorithm slightly increased the capability of the network to discern moving and static objects, and improved the overall network performance enough to allow for more intensive dropout to decrease overfitting, while not sacrificing on the IU metric. This approach also loosens the limitation on the dataset set size allowing for future work to increase the frequency of received measurements to 10 Hz [6].

The general trend in the overfit region of the network was to lose performance in the classification of 'static' and 'moving' cells. From Table 3, the predictions decreased in both recall and precision. Nevertheless, due to the weighted cross-entropy loss formulation, the network does not assume that these classes do not exist, but rather makes incorrect predictions of occupied space in labeled free space. It is reasonable to expect that as the number of training examples with 'static' and 'moving' objects labels increases, the network will have more success in discerning these two classes.

The plots in Figure 8 show that in this validation set example, the FCN expected moving objects near the road, and static objects in the parking lot space matching the labeled obstacles. Hence, the network did learn some elements of the temporal and spatial structure of the data, despite the overfitting, showing merit for the proposed approach. Figure 8 also portrays some of the inaccuracies within the dataset itself. The DST moving occupancy grid shows a higher probability of moving objects in the right portion of the image, where none exist in the expected labels. The KITTI tracking dataset contains obstacle labels referred to as 'DontCare' regions which are ignored in the labeling process due to insufficient information for the human annotators to generate 3-D bounding boxes surrounding these obstacles [6]. Therefore, the plot of the 'moving' class DST occupancy grid conveys a relatively high probability of moving obstacles in the top right region of the grid, which corresponds to the ignored obstacles. It is interesting to note, that the network smoothes the edges of the 'building' regions in the grid due to dropout regularization, possibly accounting for any irregularities in the boundaries.

Furthermore, despite the loss curves in Figure 6 showing overfitting, the mean validation set IU continued to increase, suggesting that the cross-entropy loss is not representative of the IU metric. A method to utilize IU directly as a loss in a binary classification problem has been proposed in [20]. Extending this formulation to multi-class segmentation may improve the performance of the proposed approach. An attempt was made to use the negative of the IU metric as a loss directly by parallelizing the approximation to IU in [20] to the number of classes. However, this novel loss behaved in an unstable manor, requiring very precise parameter tuning which will be pursued in future work.

7. Conclusion

A method was introduced for optimizing the performance of a DST information fusion procedure for urban scene understanding with the use of a deep FCN. Despite the observed overfitting of the dataset, the FCN-DST framework outperformed the DST baseline in the mean IU metric reaching 0.546 and 0.531 in the validation and test sets respectively. However, little improvement was achieved in discerning moving and static cells due to insufficient data and a loss that was not representative of the evaluation metric.

To improve model performance in future work, the dataset will be expanded to include measurements at a frequency of 10 Hz increasing the number of examples to approximately 3200, facilitating more effective learning [6]. Additionally, further investigation into the multi-class IU loss will be performed such that the metric of interest is directly optimized. The approach may also benefit from passing the previous and current DST outputs through several convolutional layers independently prior to merging them into channels for input to the FCN.

Acknowledgement

Thanks to Professor Mykel Kochenderfer for the guidance and support provided to make this project possible. This course paper is a subset of the research performed for the Stanford Intelligent Systems Laboratory (SISL), and overlaps with the course project for AA222: Introduction to Multidisciplinary Design Optimization. The research project is sponsored by Ford Motor Company.

References

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Azim, A. and Aycard, O. Detection, Classification and Tracking of Moving Objects in a 3D Environment. *2012 Intelligent Vehicles Symposium*, 2012.

[3] Caltagirone. Fast LIDAR-based Road Detection Using Fully Convolutional Neural Networks. *CoRR*, abs/1703.03613, 2017.

[4] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *CoRR*, abs/1411.4734, 2014.

[5] Eitel, A., Springenberg, J.T., Spinello, L., Riedmiller, M., and Burgard, W. Multimodal Deep Learning for Robust RGB-D Object Recognition. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[6] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[7] Goeddel, R. and Olson, E. Learning Semantic Place Labels from Occupancy Grids using CNNs. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[8] Held, D., Guillory, D., Rebsamen, B., Thrun, S., and Savarese, S. A Probabilistic Framework for Real-time 3D Segmentation using Spatial, Temporal, and Semantic Cues. *Robotics: Science and Systems*, 2016.

[9] Karpathy, A. CS231n: Convolutional Neural Networks for Visual Recognition, 2016.

[10] Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *ICLR*, 2015.

[11] Kochenderfer, M.J. and Wheeler, T. AA222: Course Notes, 2017.

[12] M. Kurdej. *Exploitation of map data for the perception of intelligent vehicles*. PhD thesis, Universite de Technologie de Compiegne, 2015.

[13] Kurdej, M., Moras, D., Cherfaoui, V., and Bonnifait, P. Controlling Remanence in Evidential Grids Using Geodata for Dynamic Scene Perception. *International Journal of Approximate Reasoning*, 55(1):355–375, 2014.

[14] Kurdej, M., Moras, D., Cherfaoui, V., and Bonnifait, P. Map-aided Evidential Grids for Driving Scene Understanding. *IEEE Intelligent Transportation Systems Magazine*, pages 30–41, 2015.

[15] Long, J., Shelhamer, E., and Darrell, T. . Fully Convolutional Networks for Semantic Segmentation. *CVPR*, 2015.

[16] Oh, S.-I. and Kang, H.-B. Object Detection and Classification by Decision-Level Fusion for Intelligent Vehicle Systems. *Sensors 2017*, 2017.

[17] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org. https://www.openstreetmap.org, 2017.

[18] Poggi, M. and Mattoccia, S. Deep Stereo Fusion: combining multiple disparity hypotheses with deep-learning. *2016 Fourth International Conference on 3D Vision*, 2016.

[19] QGIS Development Team. *QGIS Geographic Information System*. Open Source Geospatial Foundation, 2009.

[20] Rahman, A. MD. and Wang, Y. Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation. *ISVC*, 2016.

[21] Rieken, J., Matthaei, R., and Maurer, M. Toward Perception-Driven Urban Environment Modeling for Automated Road Vehicles. *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015.

[22] Seeger, C., Manz,18th International Conference on Intelligent Tra M., Matters, P., and Hornegger, J. Locally Adaptive Discounting in Multi Sensor Occupancy Grid Fusion. *2016 IEEE Intelligent Vehicles Symposium (IV)*, 2016.

[23] Seeger, C., Muller, A., Schwarz, L., and Manz, M. Towards Road Type Classification with Occupancy Grids. *IEEE Intelligent Vehicles Symposium 2016 Workshop: DeepDriving - Learning Representations for Intelligent Vehicles*, 2016.

[24] B. Suger and W. Burgard. Global outer-urban navigation with openstreetmap. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2017.

[25] Yao, W., Poleswkia, P., and Krzystek, P. Classification of Urban Aerial Data Based on Pixel Labelling with Deep Convolutional Neural Networks and Logistic Regression. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLI-B7, 2016.

[26] Zhang, R., Candra, S.A., Vetter, K., and Zakhor, A. Sensor Fusion for Semantic Segmentation of Urban Scenes. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

Appendix

The following is the Dempster Shafer information fusion formulation used to obtain the perception grids, which are then inputted into the FCN architecture. Only the outline of the approach is presented here; it is described in more detail in [14]. Dempster Shafer operates with masses on sets of events as opposed to probabilities on single events. The masses for all possible sets must add up to one, similar to a probability distribution. To make the DST formulation more efficient, we introduce the following notation:

$$F - \text{free space} \tag{4}$$
$$O - \text{occupied space} \tag{5}$$
$$B - \text{building} \tag{6}$$
$$R - \text{road} \tag{7}$$
$$T - \text{intermediate space} \tag{8}$$
$$N - \text{navigable space} \tag{9}$$
$$W - \text{non-navigable space} \tag{10}$$
$$I - \text{infrastructure} \tag{11}$$
$$S - \text{static obstacle} \tag{12}$$
$$M - \text{moving obstacle.} \tag{13}$$

The algorithm begins by defining a refinement of the sensor and GIS grids. The masses associated with the possible subsets of events in the sensor grid and in the GIS grid are translated to the perception grid 'frame of reference' as follows:

$$m_{SG}(\{F\}) = m_{SG}^{PG}(\{N, W\}) \tag{14}$$
$$m_{SG}(\{O\}) = m_{SG}^{PG}(\{I, S, M\}) \tag{15}$$
$$m_{SG}(\{F, O\}) = m_{SG}^{PG}(\{N, W, I, S, M\}) \tag{16}$$
$$m_{GIS}(\{B\}) = m_{GIS}^{PG}(\{I\}) \tag{17}$$
$$m_{GIS}(\{R\}) = m_{GIS}^{PG}(\{N, S, M\}) \tag{18}$$
$$m_{GIS}(\{T\}) = m_{GIS}^{PG}(\{W, S, M\}) \tag{19}$$
$$m_{GIS}(\{B, R, T\}) = m_{GIS}^{PG}(\{N, W, I, S, M\}). \tag{20}$$

Note that it is assumed that if there is mass uncertainty regarding a GIS grid element, it all goes to the full uncertainty $\{B, R, T\}$ event, rather than to sets of pair events. For the simplified occupancy grid framework used in this paper, it is assumed that a LIDAR measurement has 0.8 confidence mass, and 0.2 uncertainty mass. The map is given high confidence at 0.995 mass and 0.005 uncertainty mass. Then Dempster's combination rule is applied on each cell:

$$m'^{PG}_{SG} = m_{SG}^{PG} \oplus m_{GIS}^{PG}, \tag{21}$$

where,

$$K = \sum_{\emptyset = B \cap C} m_1(B) \cdot m_2(C) \tag{22}$$
$$(m_1 \oplus m_2)(A) = \frac{\sum_{A = B \cap C} m_1(B) \cdot m_2(C)}{1 - K} \tag{23}$$
$$(m_1 \oplus m_2)(\emptyset) = 0. \tag{24}$$

To determine the dynamics of the environment, conflict masses for cells that have become free or that have become occupied are defined as follows:

$$m_{PG,t}(\emptyset_{OF}) = m_{PG,t-1}(O) \cdot m_{SG,t}(F) \tag{25}$$
$$m_{PG,t}(\emptyset_{FO}) = m_{PG,t-1}(F) \cdot m_{SG,t}(O) \tag{26}$$

where $m(O) = \sum_{A \subseteq \{I,U,S,M\}}$ and $m(F) = \sum_{A \subseteq \{N,W\}}$. Classifying a grid cell as static or moving is dependent on the accumulator ζ that stores temporal information. Four parameters are defined for accumulation: incrementation and decrementation steps $\delta_{inc} \in [0, 1]$, $\delta_{dec} \in [0, 1]$, and threshold values γ_O, γ_{empty}. These parameters were set as indicated in [14], [12] to: $2/3, 2/3, 6, 6$ respectively. The accumulator is computed according to:

$$\zeta^{(t)} = \min(1, \zeta^{(t-1)} + \delta_{inc}) \tag{27}$$
$$\text{if } m_{PG}(\emptyset_{FO}) \geq \gamma_O \text{ and } m_{PG}(\emptyset_{FO}) + m_{PG}(\emptyset_{OF}) \leq \gamma_\emptyset \tag{28}$$
$$\zeta^{(t)} = \max(0, \zeta^{(t-1)} - \delta_{inc}) \tag{29}$$
$$\text{if } m_{PG}(\emptyset_{FO}) + m_{PG}(\emptyset_{OF}) > \gamma_\emptyset \tag{30}$$
$$\zeta^{(t)} = \zeta^{(t-1)} \tag{31}$$
$$\text{otherwise.} \tag{32}$$

ζ provides a method for specializing the mass for M using the equation:

$$m'_{PG,t}(A) = S(A, B) \cdot m_{PG,t}(B) \tag{33}$$

where,

$$S(A \backslash \{M\}, A) = \zeta \qquad \forall A \subset PG \text{ and } \{M\} \in A \tag{34}$$
$$S(A, A) = 1 - \zeta \quad \forall A \subset PG \text{and} \{M\} \in A \tag{35}$$
$$S(A, A) = 1 \qquad \forall A \subset PG \text{ and } \{M\} \notin A \tag{36}$$
$$S(\cdot, \cdot) = 0 \qquad \text{otherwise.} \tag{37}$$

To model information aging, a discounting factor α is introduced:

$$m^\alpha(A) = (1 - \alpha) \cdot m(A) \qquad \forall A \subset \Omega \tag{38}$$
$$m^\alpha(\Omega) = (1 - \alpha) \cdot m(\Omega) + \alpha \tag{39}$$

where Ω is the complete uncertainty set. In experiment, the value of α was set to 0.9.

The last step of the fusion algorithm is to combine the previous perception grid with time discounting with the new information:

$$m_{PG,t} = m^{\alpha,'}_{PG,t-1} \circledast m^{'}_{SG,t}. \tag{40}$$

$\circledast$ is a modified combination rule suited for moving object detection, defined as:

$$(m_1 \circledast m_2)(A) = \sum_{A=B\cap C} m_1(B) \cdot m_2(C) \quad \forall A \subset \Omega \wedge A \neq M \tag{41}$$

$$(m_1 \circledast m_2)(M) = \sum_{M=B\cap C} m_1(B) \cdot m_2(C) + \sum_{\emptyset_{FO}=B\cap C} m_1(B) \cdot m_2(C) \tag{42}$$

$$(m_1 \circledast m_2)(\Omega) = \sum_{\Omega=B\cap C} m_1(B) \cdot m_2(C) + \sum_{\emptyset_{OF}=B\cap C} m_1(B) \cdot m_2(C) \tag{43}$$

$$(m_1 \circledast m_2)(\emptyset_{FO}) = 0 \tag{44}$$

$$(m_1 \circledast m_2)(\emptyset_{OF}) = 0. \tag{45}$$

Once the perception grid masses have been computed, a pignistic probability is defined to convert the masses to probabilistic values:

$$betP(B) = \sum_{A\in\Omega} m(A) \cdot \frac{|B \cap A|}{|A|}, \tag{46}$$

where $|A|$ is the cardinality of set A.