Autonomous Outdoor Building Navigation Using A Single Monocular Camera

JOHN S. SENG, California Polytechnic State University, San Luis Obispo, USA
LAURA H. MCGANN, California Polytechnic State University, San Luis Obispo, USA

In this work, we describe an autonomous robot system that navigates an outdoor building environment using color monocular images from a single camera. This system is able to avoid dynamic obstacles, such as pedestrians, and recognize its location in terms of which hallway it is located in. Using a multi-task convolutional neural network, the system processes images in real-time and produces predictions for 4 tasks: topological robot localization, driveable space classification, intersection detection, and hallway goal prediction. These predictions allow the robot to determine if an area is free of obstacles and allows the system to plan a safe, driveable path. We outline how training data is collected for each of the tasks, describe the overall neural network architecture, and cover what each network output head produces. We find the system can robustly traverse a limited outdoor building scenario at various times of day and lighting conditions.

Additional Key Words and Phrases: autonomous navigation, neural networks, visual place recognition, outdoor navigation, monocular camera, multi-task learning

1 INTRODUCTION

Robot navigation in and around building environments is useful in many use cases, including factory and human assistance applications. While there are still many challenges present in an indoor navigation scenario, an indoor setting often reduces the challenges of difficult lighting conditions and object appearance variability. In this work, we focus on the scenario of robot navigation around an outdoor building using a vision system. This environment provides for structure in the traversal map while giving variation in wall texture and lighting conditions. The system we propose navigates the region using images from a single monocular camera.

Using a single monocular camera as the only sensing modality provides both advantages and drawbacks. A primary advantage of the single camera is the reduced computational requirement when compared to a multi-camera setup. By using only one camera, our system is able to process video frames at 20Hz while meeting power requirements. There are several challenges though that need to be overcome with a single camera configuration. One challenge is the limitation of not having stereo vision in order to compute depth. This problem is overcome using a learned approach to estimate object depth and semantic driveability. Another challenge is robot localization without having a complete 360-degree view around the robot. Because the system uses a wide-angle lens and has a restricted operational area, it can successfully perform localization at a coarse level.

We consider an outdoor building environment to be one where there are sidewalks, connected concrete paths, and outdoor hallways for pedestrian traversal. The outdoor building environment we choose is a constrained environment and is the region around the computer science building at our university. This environment contains concrete pathways that are bounded by obstacles such as brick walls, handrails, or vegetation areas. Our system operates in the presence of obstacles such as pedestrians, chairs, and tables.

The paper is organized as follows: Section 2 outlines work that is related to this project. Section 3 describes the approach we take in solving the challenge of outdoor navigation. Section 4 outlines the

Authors' addresses: John S. Seng, California Polytechnic State University, San Luis Obispo, 1 Grand Avenue, San Luis Obispo, USA, jseng@calpoly.edu; Laura H. McGann, California Polytechnic State University, San Luis Obispo, 1 Grand Avenue, San Luis Obispo, USA, lmcgann@calpoly.edu.

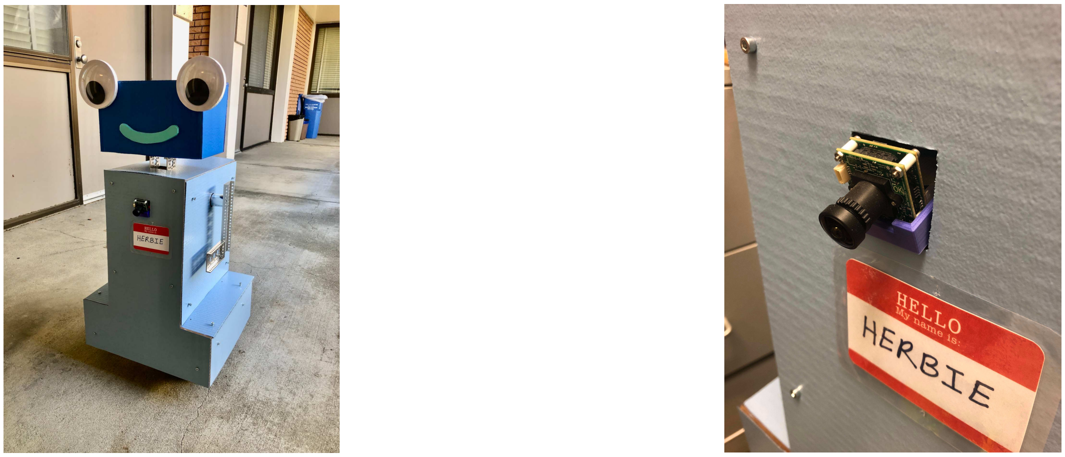

Fig. 1. Image of the robot with a closeup of the camera position.

neural network architecture and the training procedure used in our work. Section 5 covers the navigation policy used by the robot. Section 6 covers results and our experience with using this system. Section 7 concludes.

2 RELATED WORK

Computer vision is becoming a more common method of mobile robot navigation due to the rich sensor data available from a low cost sensor. Some earlier methods generated a visual memory - a collection of key image sets (visual paths) extracted from video captured during manual teleoperation - and then took in a target image, concatenating known visual paths to navigate to the end goal [1, 2]. While we similarly use a single, fixed camera to collect image data from video during manual operation and build a topological map, these works focus explicitly on natural landmark identification and matching to string paths together. Other works used landmarks with bearing-SLAM [6] or visual odometry, loop detection, and a modified ORB-SLAM2 algorithm [8] to navigate using a single camera, monocular and RGB-D, respectively. Key signboard images mapped to known locations via a trained convolutional neural network presents another method for using maps to localize [10].

Visual edge detection, combined with ultrasonic dynamic object detection, has shown promise in safely navigating in new environments and has been applied indoors [9]. More general feature extraction using Deep Belief Neural Networks showed successful obstacle avoidance outdoors [5]. This work similarly used the technique of distinguishing driveable ground to navigate, but the control function focused on keeping a mobile base in the center of a narrow lane without more global localization.

Neural networks have also been used for end-to-end (sensor data to robot motion command) mapping. Pfeiffer et al. [11] used a combination of reinforcement and imitation learning to learn optimal navigation policies in a given indoor setting, although this approach is limited to trained targets. Zhu et al. [17] presented a deep reinforcement learning approach that accepts a current state and a desired goal state image to predict the next best robot action in an attempt to be more flexible for new targets.

Yeboah et al. used transfer learning to develop more incremental control schemes: a deep convolutional neural network for semantic scene segmentation [15] and a two-branched, siamese CNN to simultaneously perform semantic segmentation (for global goal finding) and scene classification (for local scene placement) [16].

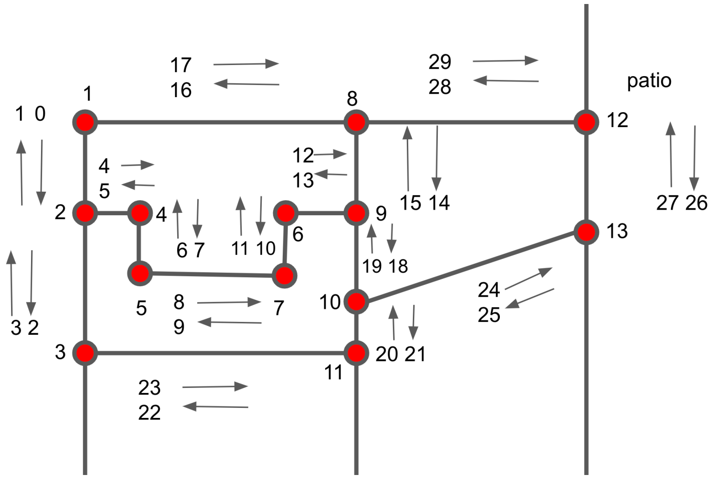

Fig. 2. Topological map of the outdoor robot environment. Hallways are represented as edges and intersections are represented as vertices. Numerical intersection and directional hallway IDs are shown.

3 SYSTEM OVERVIEW

For this system, we pose our robot navigation problem as a multi-task learning problem. We define 4 sub-tasks that need to be solved in order to successfully navigate an outdoor building environment: topological robot localization, driveable space classification, intersection detection, and hallway goal prediction. Topological localization is the task of determining the robot position in a topological map given a camera image. Driveable space classification is used to identify in which regions of an image the robot can drive and which regions are not semantically driveable. Intersection detection is a classification problem to identify if the robot has come upon an intersection region of the map. Hallway goal prediction provides a target location in the current image of where the robot should navigate towards. Figure 1 shows the robot and camera configuration.

3.1 Topological Robot Localization

The map of the outdoor building is represented in a topological format where each physical hallway is assigned two identification numbers, one for each direction of traversal. A representation of the topological map is shown in Figure 2. The hallways are represented as edges in the topological map and the intersections are represented as vertices. Although the intersections are assigned numbers, the intersection IDs themselves are not used during local navigation. It is the hallway IDs that are used in global planning to determine a route from a starting hallway to a destination hallway. For the traversal area of our robot, the final map in our system contains a total of 30 hallway IDs corresponding to 15 different hallways, one ID per direction. Although our traversal map is quite structured with straight hallways, our localization system can function with a map containing curved and less clearly defined hallways if the start and end of the hallways are clearly defined. We discuss in Section 3.3 on intersection detection how to define these hallway start and end boundaries so the neural network can identify them.

In this system, robot localization is treated as a classification problem and therefore each hallway image must provide a unique visual perspective in order to be correctly classified. Because we are using a wide-angle camera (2.5mm M12 lens with a 130-degree horizontal field of view), the images captured of each hallway are unique enough to provide high classification accuracy. Captured images contain regions of the floor and ceiling (or sky), and this aids in localization.

The process of data collection involves recording images while manually driving the robot down a hallway and steering the robot up to 30 degrees from pointing straight down the hallway. These images are recorded with and without pedestrians in order to increase dataset diversity. Images where the robot is pointing more than 30 degrees away from the primary hallway axis are not collected, and when these images are presented to the trained neural network, they often result in a low confidence output from the network. We apply a filtering system in order to ignore low confidence predictions. Sample output of this localization prediction task can be seen in Figure 3. In these sample images, the numbers in the upper right corners indicate the predicted hallway IDs with the corresponding confidences: hallway 1 (left) and hallway 29 (right), both with a confidence of 1.000.

3.2 Driveable Space Classification

The driveable space is the obstacle-free area in front of the robot where it is safe for the robot to drive. An area where a robot cannot drive may be a region where the robot physically cannot drive (such as a wall or closed door), or it may be a space where a robot should not drive, even if it is physically possible (such as on grass or under a handrail).

In order to identify driveable space, we divide the input image into 64 column slices (10 pixels wide by 360 pixels high). For each of the column slices, the point where the driveable space ends (starting from the bottom of the image to logically follow the robot driving forward) is labeled. This y-coordinate of the end of the driveable space is converted into a single floating-point value between 0 and 1 representing the fraction of the space that is driveable. This labeling process allows the driveable space in each image to be expressed as 64 floating-point numbers (one for each column slice). Sample network output is shown in Figure 3 where the points are connected with a yellow line. Any region below the yellow line is considered driveable space. The labeling process itself is achieved by drawing a polygon outlining the free space starting from the bottom of the image and progressing toward the top. Any semantically un-traversable regions are excluded from the polygon. These regions include areas such as stairs, small regions between pillars, and areas under tables.

Once the driveable space boundary has been predicted by the network, a conversion from the coordinates in image space into the 2-D world coordinates is performed. Because the region of robot traversal is relatively flat, we assume that all the boundary points are in the ground plane, and this allows a conversion using a pre-computed transformation matrix.

3.3 Intersection Detection

Intersection detection is the task of identifying if the robot is reaching the end of a hallway section and coming to an intersection. Once the robot has identified an intersection is present, a turn maneuver can be executed. The intersection detection problem is treated as a single output classification problem where the network should output a 1 or a 0 depending on whether or not there is an intersection immediately present in the incoming video image.

Detecting intersections in our topological map is somewhat akin to the problem of visual place recognition at a very fine-grained level. To treat our intersection detection problem as classification, we need to clearly denote the presence of an intersection. For each intersection, we select a horizontal line (an intersection marker) on the ground that separates the intersection from the adjoining hallway. At several locations along our path, there are dividing lines in the concrete from which we select certain ones as intersection markers. Once the line is selected, we then place the front of the robot along that line and

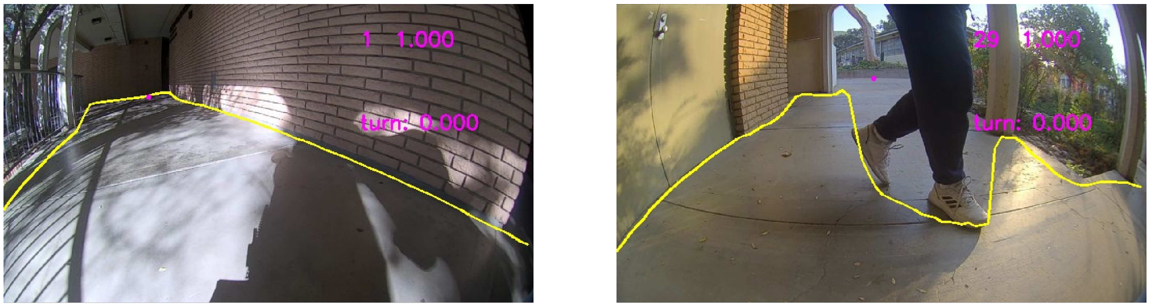

Fig. 3. Sample images from the monocular camera labeled with the driveable area boundary (yellow), the goal (magenta dot), the hallway localization (top magenta label), and the intersection detection (middle magenta label).

specify that when the robot reaches that location (as well as any area slightly after that location) an intersection should be detected. In order to create a buffer zone to prevent false detections, we classify images where the robot is 40cm away from the horizontal line (having not reached the intersection marker yet) as non-intersections. Any robot location further than the 40cm distance is also considered a non-intersection. We allow the neural network to interpolate any images between the horizontal line and the 40cm distance away. In practice, we find the neural network is able to detect these intersection boundaries with high accuracy and usually within 5cm.

3.4 Hallway Goal Prediction

When presented with an input image, the goal finding portion of the neural network outputs a single coordinate pair (2-dimensional value) corresponding to the end or 'vanishing point' of the hallway in the image. This point represents the predicted goal, providing a target for the robot to track towards when driving along a physical hallway. In Figure 3, the magenta point at the end of each hallway is the predicted goal location.

The training data for the hallway goal network component consists of input images and manually labeled 2-dimensional coordinates representing the goal in each image. When collecting training data, it is important to have the predicted goal be the same whether or not there are obstacles present. The prediction should be the same even in the presence of obstacles as we rely on the local planner to navigate around the obstacle while the goal prediction is used by the global planner.

4 NEURAL NETWORK DESIGN AND ARCHITECTURE

The neural network architecture for our system needs to offer fast inferencing time while producing correct predictions for the 4 tasks. Our problem lends itself to networks designed for mobile architectures which satisfy real-time inference demands while providing high accuracy. In this section, we describe our system neural network architecture.

4.1 Network Architecture

The neural network backbone used for our work is based on the EfficientNet-Lite [14] convolutional neural network architecture. Although this network is based on the EfficientNet [13] architecture, we include layer simplifications for faster inference on mobile devices. We specifically select the pre-trained EfficientNet-Lite0 model, and while it is the smallest model of the EfficientNet-Lite family, we find the performance satisfactory. With the final ImageNet classification layer removed, the EfficientNet-Lite0

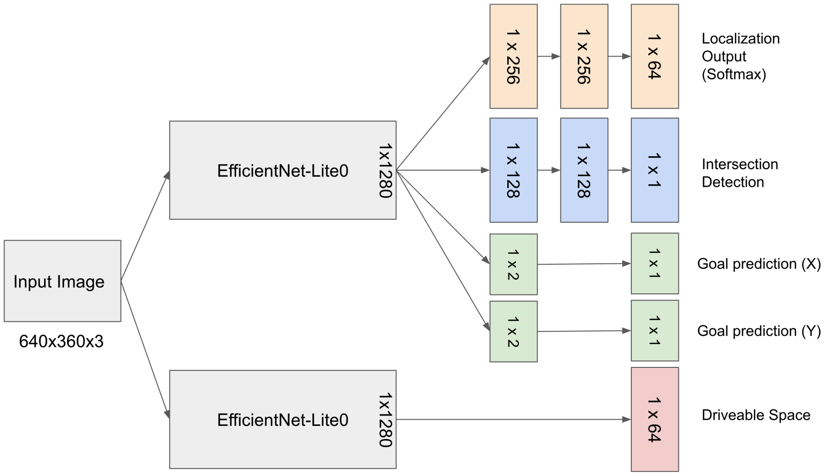

Fig. 4. The neural network architecture used in our system.

backbone has a total of 3.37M parameters. The input to the model is a 640x360 image with 3 color channels, and the overall architecture is shown in Figure 4. In total, the entire neural network has 7.42M parameters.

Because the driveable space detection task is critical to avoid obstacles and collisions, the highest accuracy possible is desirable for this network. To achieve this, we dedicate one EfficientNet-Lite backbone for this particular task. The other three tasks share a separate EfficientNet-Lite backbone.

4.2 Network Output Heads

For each of the tasks, we use a distinct output head connected to the appropriate backbone. The resulting network uses hard parameter sharing [4]. Each of the output heads is tuned to provide good performance, and the output head hidden layers are run through a LeakyReLU activation function.

For the topological localization output head, the 1280 outputs of the EfficientNet-Lite backbone are successively fed into two layers of 256 neurons. The final layer of this output head is 64 neurons where each neuron corresponds to one of the possible hallway identification numbers. We select 64 for possible future expansion of the topological map. This final layer is fed into a softmax activation function which produces a predicted hallway ID with the highest output (confidence) value.

The driveable space output head consists of 64 neurons directly connected to the output of a dedicated EfficientNet-Lite0 backbone. We find that using the dedicated backbone provides the best performance for this task. Additionally, adding extra hidden layers between the backbone and the output layer led to overfitting and thus were not needed.

For intersection detection, the output head is structured in a similar manner to the localization task except the hidden layers contain only 128 neurons and, the final layer is a single neuron output that is fed through a sigmoid activation function.

We find that the goal prediction task is highly sensitive to overfitting if there are too many parameters in the output head. For this task, there is a single hidden layer for each of the two dimensions of the goal prediction. This hidden layer comprises of two neurons, and we find that increasing the number of neurons in this layer even slightly results in worse performance due to overfitting.

Dataset	Num. Training Images	Num. Test Images	Test Error
Topological Localization	6459	920	1.1%
Driveable Space Classification	6844	363	0.8%
Intersection Detection	7776	1217	1.0%
Hallway Goal Detection	6230	936	2.1%

Table 1. Neural network dataset sizes and test errors.

4.3 Network Training and Inference

For the multi-task network with the three output heads, we begin by freezing the pre-trained backbone and training the weights of the output heads. After the weights of the three output heads have converged, the full training process begins where the weights of each output head and backbone are updated. The training proceeds by: unfreezing the backbone, freezing two of the three output heads, and updating the weights of the current head and the backbone. Once the gradient update step is complete, the current output head is frozen and the next output head is selected for training. When updating the weights for each output head, the weights of the EfficientNet-Lite backbone are also updated.

We train the model in PyTorch using the AdamW optimizer with an initial learning rate of .003 with cosine decay and warm restarts [3]. The neural network training runs for 200 epochs. The size of each of the task datasets is shown in Table 1. For the driveable space prediction and intersection detection, we apply image augmentations to the training set that vary image brightness and noise levels but do not distort the image itself. For the topological localization and hallway goal predictions, we apply the previous augmentations along with distortions on the image. On the hallway goal predictions, we adjust the goal position for any distortion that is applied.

In order to reduce the amount of data that needs to be labeled, we use an active learning [7] approach to data collection. For our active learning process, we train the model four times using different random initializations and then pass a new set of unlabeled data through the four models. For an unlabeled data item, if the models produce an output with high variance, then the image is added to the training dataset. This approach allows the dataset to contain difficult examples while minimizing the number of labeled images.

At runtime, the image captured from the camera is in 1920x1080 UYVY format and is downsampled and converted to a 640x360 RGB image. This downsampled image is passed through the network. The final inference time for the model on a Jetson AGX Xavier in the 10W power mode is 40ms. We do not do any quantization of the weights and instead leave them in the trained FP32 format.

5 ROBOT NAVIGATION

Navigation for our system is based on the ROS Navigation Stack [**?**]. This mature navigation stack allows traversing from waypoint to waypoint given a global map and local sensor readings. For path planning, obstacles are represented in a local and global costmap. The global costmap is used to plan a path from the current robot location to the predicted hallway goal position in 3-D space. The local costmap is used for local planning around nearby obstacles, and we use a local costmap of 3 meters square. The driveable space prediction is projected into a top-down, bird's-eye view and then converted into a simulated 2-D planar laser scan which is placed in the local costmap. A visual representation of this process is shown in Figure 5. In our system, the global map starts out empty and is built at run-time. Global navigation of the topological map requires generating a list of connected hallways using a standard graph traversal algorithm.

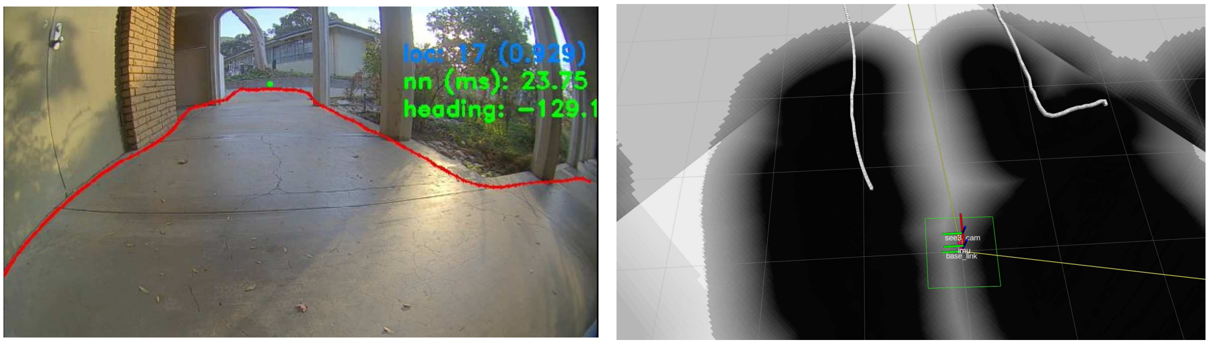

Fig. 5. Sample navigation images show the camera input (left) and the projected driveable space in world coordinates (right).

Once the robot has localized to a particular hallway, the task becomes to navigate from the current position to the end of the hallway. The hallway goal point is projected into the global map and a global plan is generated from the current position to this end point. Any dynamic obstacles will appear in the robot's local costmap, and the local planner will attempt to navigate around the local obstacle. In the current implementation, if a dynamic obstacle prevents the robot from progressing in the local plan, the robot will stop to avoid a collision and wait until the path becomes clear.

As the system proceeds down a hallway while navigating towards the hallway goal, the network may detect an upcoming intersection. The neural network only predicts the presence of an intersection and does not predict which turn directions are present. The topological map informs whether the robot is able to make a left turn, right turn, or may turn in either direction. Turn routines currently consist of driving towards the hallway goal for a fixed distance and then performing a turn in-place until the new hallway goal is detected. Turns do appear as a mechanical, turn-in-place maneuver, but it does guarantee the robot will be centered in the next hallway and that the hallway goal prediction will produce a highly probable result.

6 RESULTS AND DISCUSSION

This section covers the results of training the neural network and of operating the robot in the outdoor environment. It also discusses various failure modes.

6.1 Experience

Table 1 shows the results of training the multi-task neural network. The output of the driveable space task produces very accurate predictions even without a hidden layer in the output head. The training dataset consists of several images of pedestrians and does well in the presence of obstacles that have not been seen before, such as various tables and chairs.

The topological localization task has high accuracy, and this is in part because the training set contains a number of off-axis hallway images. In testing, the robot can be turned away from a hallway direction by up to 30 degrees and the robot will still maintain the same localization output.

The hallway goal prediction has the highest error rate of all the network tasks, and we find that it is very sensitive to whether pedestrians are present or not. As opposed to the localization task which can tolerate a larger amount of various dynamic objects that are present, the goal prediction task is more sensitive. We collect more training images with pedestrians in order to overcome this sensitivity.

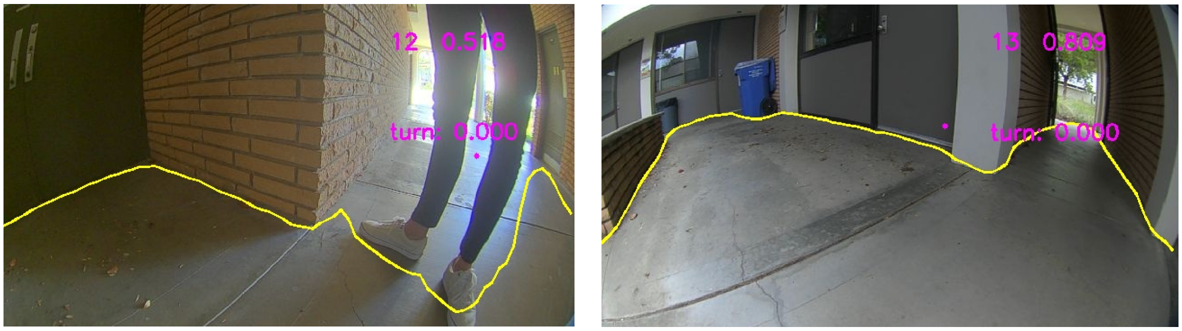

Fig. 6. Sample images of failure modes.

In terms of overall navigation success, we are able to run the robot continuously for approximately 1 hour at a time, generally without collisions or loss of localization. We are currently limited to the 1-hour time limit because we are using an RC-style lithium-polymer battery that is rated at 6000mAh. During these navigation runs, the robot is periodically directed to various hallway waypoints. These runs occur during student passing periods between class sessions when there is a higher density of pedestrians.

6.2 Navigation Failure Modes

Figure 6 provides two images of failure modes that can occur with the navigation system. In the left image, the driveable space, turn detection, and hallway goal outputs are all valid, but the localization is producing a low-confidence (.518) incorrect prediction (hallway 12 instead of hallway 17). To produce this image, we manually steer the robot towards a short hallway (left side of the left image) ending in a closed door. This type of scenario is not currently in the training dataset and thus leads to an incorrect localization prediction. In order to mitigate such scenarios, we apply a filter that samples the last 10 localization outputs and selects the localization prediction that is most common out of those 10.

The right failure image shows the hallway goal prediction when the robot is in the process of turning and is facing the area between two hallways. Because the hallway goal dataset is trained with many off-axis images, it can tolerate a significant amount of deviation from the actual hallway direction, but in this instance it is somewhat ambiguous which hallway is correct to proceed towards. During turn operations, the system does not follow the goal prediction and instead uses the localization output to determine when to stop turning.

7 CONCLUSION

In this work, we present a system that allows a robot to successfully traverse an outdoor building environment using a topological map. The map is divided into hallways and intersections, and these elements are represented in a topological format. Using a single monocular camera, the robot is able to make predictions using a convolutional neural network on 4 different tasks: topological localization, driveable space prediction, intersection detection, and hallway goal prediction. These predictions are made in real-time for each image that is captured.

The neural network has a multi-task network architecture and incorporates hard parameter sharing using an EfficientNet-Lite backbone network. Using training data that has been collected from this limited outdoor environment, the system is successful in autonomously operating in the presence of dynamic objects. We have observed that the system can operate continuously for 1-hour time segments while navigating through various hallways within the topological map.

REFERENCES

[1] Jonathan Courbon, Youcef Mezouar, Laurent Eck, and Philippe Martinet. 2008. Efficient visual memory based navigation of indoor robot with a wide-field of view camera. In *2008 10th International Conference on Control, Automation, Robotics and Vision*. 268–273. https://doi.org/10.1109/ICARCV.2008.4795530

[2] Jonathan Courbon, Youcef Mezouar, and Philippe Martinet. 2009. Autonomous Navigation of Vehicles from a Visual Memory Using a Generic Camera Model. *IEEE Transactions on Intelligent Transportation Systems* 10, 3 (2009), 392–402.

[3] Yimin Ding. 2021. The Impact of Learning Rate Decay and Periodical Learning Rate Restart on Artificial Neural Network. In *2021 2nd International Conference on Artificial Intelligence in Electronics Engineering* (Phuket, Thailand) *(AIEE 2021)*. Association for Computing Machinery, New York, NY, USA, 6–14.

[4] Zhiqiang Gao, Dawei Liu, Kaizhu Huang, and Yi Huang. 2019. Context-Aware Human Activity and Smartphone Position-Mining with Motion Sensors. *Remote Sensing* 11 (10 2019), 2531. https://doi.org/10.3390/rs11212531

[5] K.M. Ibrahim Khalilullah, Shunsuke Ota, Toshiyuki Yasuda, and Mitsuru Jindai. 2017. Development of robot navigation method based on single camera vision using deep learning. In *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*. 939–942. https://doi.org/10.23919/SICE.2017.8105675

[6] Dong Wook Ko, Chuho Yi, and Il Hong Suh. 2012. Semantic mapping and navigation with visual planar landmarks. In *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. 255–258. https://doi.org/10.1109/URAI.2012.6462988

[7] Yu Ma, Shaoxing Lu, Erya Xu, Tian Yu, and Lijian Zhou. 2020. Combining Active Learning and Data Augmentation for Image Classification. In *Proceedings of the 2020 3rd International Conference on Big Data Technologies* (Qingdao, China) *(ICBDT 2020)*. Association for Computing Machinery, New York, NY, USA, 58–62. https://doi.org/10.1145/3422713.3422726

[8] Ying-Ze Mu, Chao-Yi Dong, Qi-Ming Chen, Bo-Chen Li, and Zhi-Qiang Fan. 2020. Research on Navigation and Path Planning of Mobile Robot Based on Vision Sensor. In *Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence* (Tianjin, China) *(ICCAI '20)*. Association for Computing Machinery, New York, NY, USA, 519–524. https://doi.org/10.1145/3404555.3404589

[9] I. Ohya, A. Kosaka, and A. Kak. 1998. Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing. *IEEE Transactions on Robotics and Automation* 14, 6 (1998), 969–978. https://doi.org/10.1109/70.736780

[10] G. S.T. Perera, K. W.R. Madhubhashini, Dilani Lunugalage, D. V.S. Piyathilaka, W. H.U. Lakshani, and Dharshana Kasthurirathna. 2020. Computer Vision Based Indoor Navigation for Shopping Complexes. In *Proceedings of the 2020 4th International Conference on Vision, Image and Signal Processing* (Bangkok, Thailand) *(ICVISP 2020)*. Association for Computing Machinery, New York, NY, USA, Article 15, 6 pages. https://doi.org/10.1145/3448823.3448828

[11] Mark Pfeiffer, Samarth Shukla, Matteo Turchetta, Cesar Cadena, Andreas Krause, Roland Siegwart, and Juan Nieto. 2018. Reinforced Imitation: Sample Efficient Deep Reinforcement Learning for Mapless Navigation by Leveraging Prior Demonstrations. *IEEE Robotics and Automation Letters* 3, 4 (2018), 4423–4430. https://doi.org/10.1109/LRA.2018.2869644

[12]]ros Stanford Artificial Intelligence Laboratory et al. [n. d.]. Robotic Operating System. https://www.ros.org

[13] Mingxing Tan and Quoc Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 6105–6114. https://proceedings.mlr.press/v97/tan19a.html

[14] Mingxing Tan and Quoc V. Le. 2020. Higher accuracy on vision models with EfficientNet-Lite. arXiv:1905.11946 [cs.LG]

[15] Yao Yeboah, Cai Yanguang, Wei Wu, and Zeyad Farisi. 2018. Semantic Scene Segmentation for Indoor Robot Navigation via Deep Learning. In *Proceedings of the 3rd International Conference on Robotics, Control and Automation* (Chengdu, China) *(ICRCA '18)*. Association for Computing Machinery, New York, NY, USA, 112–118.

[16] Yao Yeboah, Cai Yanguang, Wei Wu, and Shuai He. 2018. Autonomous Indoor Robot Navigation via Siamese Deep Convolutional Neural Network. In *Proceedings of the 2018 International Conference on Artificial Intelligence and Pattern Recognition* (Beijing, China) *(AIPR 2018)*. Association for Computing Machinery, New York, NY, USA, 113–119. https://doi.org/10.1145/3268866.3268886

[17] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J. Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. 2017. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 3357–3364. https://doi.org/10.1109/ICRA.2017.7989381