The Workshops of the Thirty-Second
AAAI Conference on Artificial Intelligence

Automatic Sign Detection with Application to an Assistive Robot

**Amlaan Shakeel, Peining Che, Xian Liu,
Yamuna Rajasekhar, John Femiani**
Miami University
510 E. High St
Oxford, OH, 45056
femianjc@miamioh.edu

Abstract

This paper explores automatic detection and classification of exit signs with the aim of enabling a service robot to assist the visually impaired with indoor navigation, inspired by a guide dog. The ultimate aim is to achieve autonomous indoor navigation using computer vision to identify navigational goals in an unfamiliar environment. In particular, we focus on the task of exiting a building by following exit signs that may include arrows that indicate where the next door or sign is located. The proposed method utilizes a deep learning framework, Faster Regional Convolutional Neural Network (Faster R-CNN), to classify and localize exit signs in real time. The Faster R-CNN model achieved competitive results on more sizable dataset[1] than existing approaches.

Introduction

Navigation may be one of the biggest challenges for the visually impaired which reduces their overall mobility. The most adopted solution is the white cane that allows a user to avoid nearby obstacles. In comparison to the traditional white cane, guide dogs have been found to be more successful in increasing mobility and independence of blind people (Whitmarsh 2005). The increased ease of mobility is simply because the guide dogs can see the path ahead of them and steer a person in the right direction, especially in an unfamiliar place. This leads to idea of a robotic assistant that potentially can do the same important things, i.e., see the surroundings to determine a path while avoiding obstacles to steer a person through its own motion.

Many research papers can be found on robotic assistance for people with physical disabilities such as in (Fukuda et al. 2011; Megalingam et al. 2014; Chung, Kim, and Rhee 2014; Peng et al. 2017; Monteiro et al. 2017; Wachaja et al. 2017). These robots are designed to detect and correct the posture of the person and in some cases provide directional force when the person is likely to fall. The idea of robotic assistance for the visually impaired, however, represents an opportunity for new development. This is because such a robot would need to be capable of autonomous navigation in an unfamiliar environment. The technologies required to achieve autonomy, such as computer vision, deep learning, and depth perception, have seen significant progress in recent years. One of the key challenges is indoor navigation, in which satellite-based Global Positioning System (GPS) is unavailable, so navigation relies on computer vision, depth perception or triangulation using wireless networks (Youssef, Agrawala, and Shankar 2003).

The co-robotic cane (Ye et al. 2016) is an example of active assistance that uses computer vision for pose estimation and object detection. A servo is attached to the bottom of the cane that can steer the user in a certain direction. The limitation of this device is that it requires a floor plan for navigation. The propsed approach aims to use signs posted in the building in order to navigate towards an exit, in much the same way that a sighted person might navigate in an unfamiliar environment. Another work (Lee, Chiu, and Zhuo 2013) presents a design for a social assistive robot and a proof-of-concept using a robotic setup consisting of a base, kinect sensor, laser range finder, SONAR and a PTZ camera. The prototype robot was capable of patrolling indoor environments by wall fallowing and obstacle avoidance. Simple colored markers are used to identify end goals. Although the design mentions the use of artificial neural networks to identify objects, signs and faces, it does not give any further details.

Our aim is to create a proof-of-concept for a service robot that can recognize objects in its surroundings and make sensible navigational decisions using computer vision and depth. Our priority is to interpret objects, such as signs, in order to determine a path just as a person would in a foreign environment. A fairly reliable type of indoor signage are "Exit" signs that aid navigation out of a building. Exit signs usually terminate at a specific door or an elevator. The objective of the robot is to analyze images in real time to recognize exit signs, follow direction as suggested by sign and stop in front of the final door or elevator. To achieve this we use a computer vision system capable of classifying and localizing objects and depth sensors to obtain the sign's 3D position relative to the robot.

Contributions

We make the following contributions:

- We demonstrate the feasibility of a system for locating and recognizing exist signs using state-of-the art deep

[1]Our dataset is available at http://teams.cec.miamioh.edu/vision/aaai18..

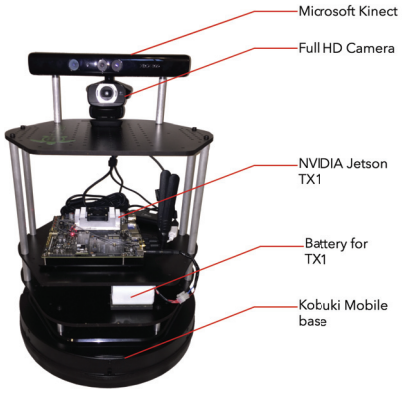


Figure 1: The prototype robot constructed for this work; the base is a TurtleBot, and an NVIDIA Jetson TX1 was used for GPU-accelerated inference. We used a Full HD camera to capture imagery at sufficient resolution to resolve exit signs.

learning techniques (Regional Convolutional Neural Network (R-CNN))

- We achieve competitive results on more sizable evaluation data than existing approaches for recognizing exit signs
- We provide a proof of concept of an actual robot developed to use this system to recognize exit signs and estimating distance to it.

Methodology

Inspired by the recent development in computer vision, deep learning, and depth perception, we aim to create a proof-of-concept service robot that can recognize its surroundings objects and make sensible navigational decisions using depth information. We choose to use "Exit" signs as our main objective because they are reliable and abundant in indoor environments. In addition we imagine that quickly finding one's way out of a building is a useful and important task.

As a proof-of-concept we built a system using a TurtleBot with a Kinect sensor controlled by an NVIDIA Jetson TX1, shown in Figure 1. A Full High Definition (HD) camera and the Kinect sensor are places close together atop the prototype as they are used in combination to detect and localize exit signs. The Jetson controller board consists of 256 Graphics Processing Unit (GPU) cores that enables us to run the detection in real time. We were able to identify and localize signs on the prototype robot, and we were able to navigate based on exit signs while also avoiding obstacles.

Our classification and localization pipeline starts with RGB color images from a Full HD camera and registered depth images (RBG+D) from a kinect sensor. We use the RGB image to obtain object bounding boxes and classifications. The relative distance is computed by using the 2D position of the object on RGB image and estimating its distance as the minimum value in a corresponding, but much lower resolution, registered depth image. This is assuming the signs are the most foreground object visible within the

Table 1: Classes in Dataset with examples

Class	Example	Class	Example
Exit sign		Exit sign, left arrow	
Exit sign, right arrow		Exit sign, both arrows	
Stairway sign		Elevator sign	
Elevator door		Door	

estimated bounding box. We note that this is a coarse estimate and an instance segmentation approaches like Mask-RCNN (He et al. 2017) could give more precise estimates, however an approximate depth estimate suffices because a service robot only requires the direction of the nearest sign in order to navigate effectively.

A R-CNN (Girshick et al. 2013) is a combination of object region proposals and a Convolutional Neural Network (CNN) for the purpose of object classification and localization at the same time. The system extracts around 2000 region proposals in a given input image and uses a Convolutional Neural Network (CNN) trained for feature extraction on each region. It then classifies each region using an Support Vector Machine (SVM) with confidence values. Faster Regional Convolutional Neural Network (Faster R-CNN) is an enhanced version of R-CNN that shows improved object classification result and faster convergence during training. A flowchart of our computer vision pipeline is shown in figure 2.

CNNs require a large set of images along with annotations. Although the original Faster R-CNN was trained on ImageNet (Deng et al. 2009), a large enough image dataset is still required to refine the model for new tasks. Therefore, we collected about 8000 images across 8 classes of objects shown in Table 1 by taking pictures and videos in a variety of buildings and by scraping the web. Images were annotated using LabelImg (Tzutalin 2013) and our video data was annotated using the VATIC annotation tool (Vondrick, Patterson, and Ramanan). The annotations are XML files in PASCAL VOC format(Everingham et al. 2010).

A major portion of the dataset was created by capturing images and videos of buildings on Miami University campus at Oxford, Ohio. Images scraped from the web added a wider variety of exit signs. Images from the web account for about 350 images in the dataset.

Videos were captured while moving through the corridors of the building. This allowed us to capture signs from multiple angles simulating the way the moving robot would capture them. Images sliced from annotated videos account for about 7200 out of 8000 images in the dataset.

Evaluation

The model was evaluated using on nearly 800 holdout images, which is about 10% the size of training dataset. The

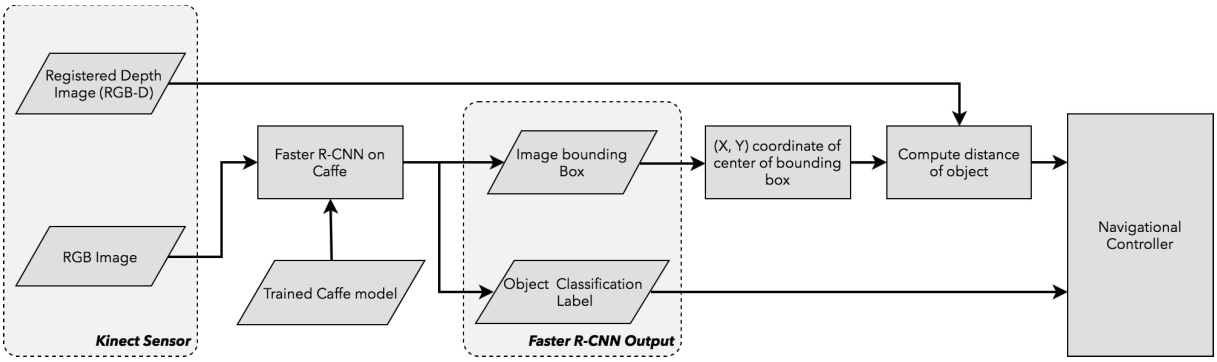


Figure 2: Computer vision pipeline using Faster R-CNN

Table 2: Results of model validation

Object Class	AveP (%)	Object Class	AveP (%)
Exit forward	89.14	Exit left	94.73
Exit right	91.94	Exit both	87.76
Stairs	76.02	Elevator	65.17
Elevator door	95.02	Door	89.03
		mAP	86.10

Table 3: Comparison against existing methods

Class	Sarna'11 (Success %)	Wang'11 (F-score)	Ours (F-score)
Exit Forward	80	94.74	89.66
Exit Left	NA	81.08	93.73
Exit Right	NA	90.47	93.33

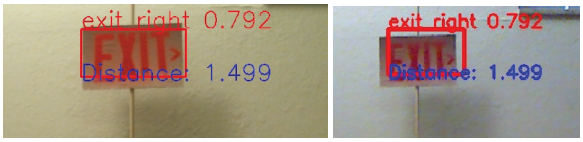
precision and recall for each object class are based on detections with over 50% Union (IoU) performed on the test data, and we provide the the Average Precision (AveP) for each of the classes. As a result, the AveP of each object class falls between 65% to 95%, and the mean Average Precision (mAP) is 86.0% as shown in 2.

For comparison, we identified papers that have explored the task of recognizing exit signs. The authors in (Aaron Sarna 2011) used 10-fold cross validation method to validate their trained model, but we did not use 10-fold cross validation method due to difficulties of training 10 R-CNN models. The only statistic provided in their paper that are useful for comparison is the success rate. Although success rate is not exactly the same as AveP, it is the closest metric to compare with.

The method employed in (Wang and Tian 2011) uses saliency maps and query patterns to perform detection. They report their results as F-scores, which we compare with ours[2] in Table 3.

The trained model was run on the NVIDIA Jetson TX1 using the Robot Operating System (ROS) framework and OpenCV (Bradski 2000) on a TurtleBot. This allowed processing images in real-time to Faster R-CNN as a part of the computer vision pipeline. For processing a Full HD image the average time duration is about 600ms (unoptimized python code). Example detections and depths are shown in figure 3.

[2]Based on 20 (forward), 20 (left) and 22 (right) images vs 347, 189, and 78 example images in ours



(a) HD Camera (b) Kinect

Figure 3: Sample output with the classification names with result of mapping bounding box coordinates from HD camera to Kinect camera Numbers beside the names are confidence scores. Below the bounding box is the distance of the objects from the camera in meters.

Faster R-CNN allows us to change the number of region proposals it will process. This gave an opportunity to reduce the execution time by limiting this number as long as there is no loss in accuracy. The loss in accuracy depends on the application. Since we are dealing with exit signs, we do not expect to see a large number of signs in the same frame. Based on our experiments using 3 exit signs visible in the frame, the accuracy started to drop at around 50 region proposals. At 300 region proposals the execution time was 789 ms. We reduced this execution time to about 600 ms by using 80 region proposals, which was about our observed threshold of 50 in our experiments.

Discussion

The most prominent errors were observed in the cases of directional exit signs, which we attribute to variation in the way directions are indicated on different signs; the most common error is to confuse a directional sign for "Exit forward". Errors also occur when the sign is small or unclear, the accuracy seems to increases when images are taken from a closer distance.

The computer vision pipeline is part of the robot navigation system shown in figure 4, which was implemented in C++ using the ROS library. In the absence of any other

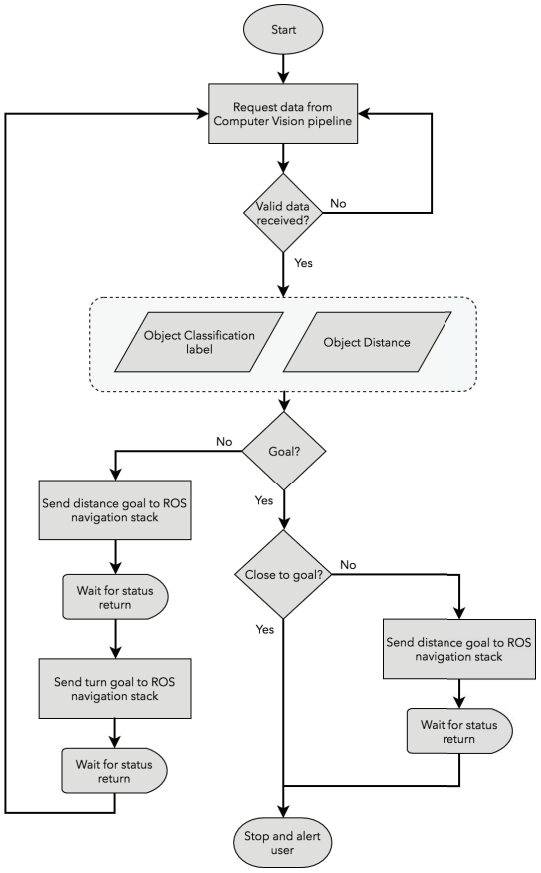


Figure 4: A flowchart for the navigational controller.

stimulus, the robot slowly scans the room. The computer vision system is used to recognize intermediate waypoints (exit signs, classified with directions) and final goals (exit doors or elevators). The controller receives data from the computer vision pipeline and determines whether the next target is a goal or a waypoint. If it is a waypoint, the robot uses an obstacle-avoiding path planning module that is part of ROS to position it self at the target location, and turns to face the direction indicated by the sign. If it cannot see the next waypoint, it resumes scanning the room.

Conclusion

In conclusion, we trained a Faster R-CNN model to identify signs needed for an active assistance robot on a dataset containing about 8000 images with annotations. A robotic prototype was constructed to run the model, and the model performance is competitive with related work. In future work we plan to evaluate the robot's performance in exiting real buildings, to try improved RGB-D sensors, and construct smaller and faster robot or haptic tool.

References

Aaron Sarna, Michael Oleske, A. H. 2011. Robot navigation through exit sign detection.

Bradski, G. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Chung, I. Y.; Kim, S.; and Rhee, K. H. 2014. The smart cane utilizing a smart phone for the visually impaired person. In *2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE)*, 106–107.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

Everingham, M.; Van Gool, L.; Williams, C. K. I.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision* 88(2):303–338.

Fukuda, T.; Di, P.; Chen, F.; Sekiyama, K.; Huang, J.; Nakajima, M.; and Kojima, M. 2011. Advanced service robotics for human assistance and support. In *2011 International Conference on Advanced Computer Science and Information Systems*, 25–30.

Girshick, R. B.; Donahue, J.; Darrell, T.; and Malik, J. 2013. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR* abs/1311.2524.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. *arXiv preprint arXiv:1703.06870*.

Lee, M. F. R.; Chiu, F. H. S.; and Zhuo, C. 2013. Novel design of a social mobile robot for the blind disabilities. In *Proceedings of the 2013 IEEE/SICE International Symposium on System Integration*, 161–166.

Megalingam, R. K.; Nambissan, A.; Thambi, A.; Gopinath, A.; and Nandakumar, M. 2014. Sound and touch based smart cane: Better walking experience for visually challenged. In *2014 IEEE Canada International Humanitarian Technology Conference - (IHTC)*, 1–4.

Monteiro, J.; Aires, J. P.; Granada, R.; Barros, R. C.; and Meneguzzi, F. 2017. Virtual guide dog: An application to support visually-impaired people through deep convolutional neural networks. In *2017 International Joint Conference on Neural Networks (IJCNN)*, 2267–2274.

Peng, H.; Song, G.; You, J.; Zhang, Y.; and Lian, J. 2017. An indoor navigation service robot system based on vibration tactile feedback. *Adv. Robot.* 9(3):331–341.

Tzutalin, D. 2013. Labelimg.

Vondrick, C.; Patterson, D.; and Ramanan, D. Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision* 1–21. 10.1007/s11263-012-0564-1.

Wachaja, A.; Agarwal, P.; Zink, M.; Adame, M. R.; Möller, K.; and Burgard, W. 2017. Navigating blind people with walking impairments using a smart walker. *Auton. Robots* 41(3):555–573.

Wang, S., and Tian, Y. 2011. Indoor signage detection based on saliency map and bipartite graph matching. In *2011 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)*, 518–525.

Whitmarsh, L. 2005. The benefits of guide dog ownership. *Visual Impairment Research* 7(1):27–42.

Ye, C.; Hong, S.; Qian, X.; and Wu, W. 2016. Co-robotic cane: A new robotic navigation aid for the visually impaired. *IEEE Systems, Man, and Cybernetics Magazine* 2(2):33–42.

Youssef, M. A.; Agrawala, A.; and Shankar, A. U. 2003. Wlan location determination via clustering and probability distributions. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003).*, 143–150.