Robotic inference for floor objects

Yisi Zhang

Abstract—Autonomous robotic vacuums became the first robot involved in our daily life. It turns that a task as simple as cleaning the floor is not that easy. One challenge is to recognize the objects in view and make decisions for the navigation direction. With a camera implemented, real time inference is possible to recognize objects on the floor. This study tested the performance of two deep learning models, AlexNet and GoogLeNet, in classifying floor object images. Accuracy and inference time were compared between the two models. The results show that GoogLeNet is an optimal model to accomplish this task.

Index Terms—Robotic vacuum, inference, floor objects.

INTRODUCTION

AUTONOMOUS robotic vacuums are becoming part of daily life. A good robotic vacuum should be able to cover the entire floor area while performing vacuum cleaning. The robots are equipped with a set of sensors to detect obstacles, cliffs and dirt. To better recognize objects and to improve the mapping of the floor area, the new models such as the Roomba 900 series also include a camera system. The camera system can significantly improve the navigation performance when it is implemented with real time inference algorithms. Here are some examples for the advantages to have a camera system. Flat or very thin objects such as mats or wires are not easily detected by infrared sensors but can be captured by cameras. These objects tend to get entangled with the wheels and the brushes of the vacuum and sometimes can block the suction. By avoiding running into those areas, the robots are less likely to get stuck. Small objects sometimes are tricky for the robot to decide whether to push it aside or to avoid. Light, unimportant objects such as slippers may deserve being pushed away from the path so the robot does not need to take detours to reach certain areas. Delicate objects such as vase may be better to be avoided. With a camera, this kind of information can remarkably improve decision making for the robots. Another example that may hinder optimal navigation is the leggy objects such as chairs and tables. With the camera information, the robot can quickly decide if it can pass between the legs or not.

In this study, we used NVIDIA Deep Learning GPU Training System (DIGITS) to train models for recognizing floor objects, the models can be deployed on an NVIDIA Jetson board for real time inference. The models were trained to recognize four categories of objects commonly seen on the floor and important for navigation: slippers, wires, chair legs and mats. Two models, AlexNet [1] and GoogLeNet [2], were trained and compared in performance.

METHODS

Data acquisition

The training images were obtained through a Python interface with the laptop webcam. The laptop was placed on the floor. All objects were placed in front of the same background with slight variations depending on the angle and distance of the camera (Fig. 1A). Around 200 images were collected for each category with 3-6 different objects within the category (Fig. 1B). The data were RGB images with the size of 640x360. Each category of images were saved under the same folder with the name of the category.

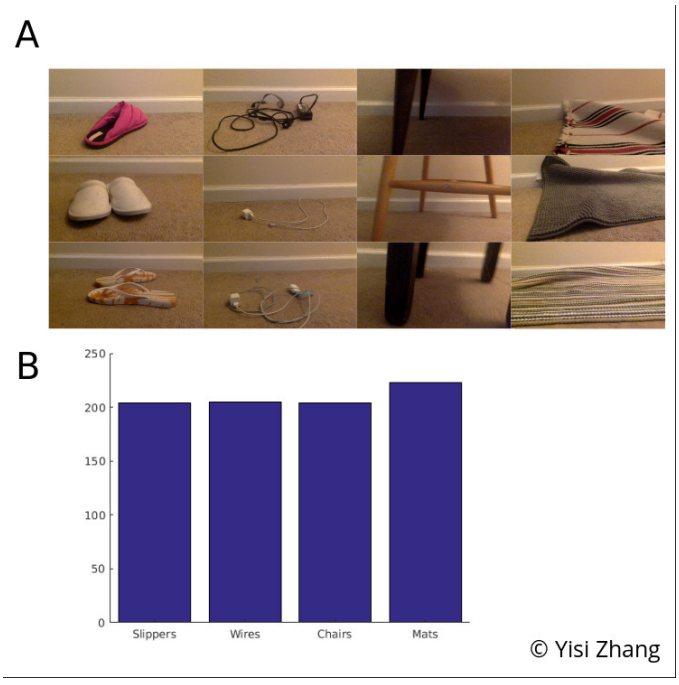

Fig. 1. Data collection. (A) Sample images from each category. (B) Data size summary.

Model training

25 percent of the images were used for validation (627 training data and 209 validation data). The models were trained with 30 epochs and 0.01 initial learning rate for stochastic gradient descent. Two network models were tested: AlexNet and GooLeNet. No modifications were made to the original setup of the DIGITS defaults.

We tested the methodology first on a set of pre-supplied data to distinguish photos of candy boxes, bottles and blank. Using GoogLeNet with a slight modification of the validation batch size to 8, the training converged fast towards 100% accuracy within 2 epochs with an initial learning rate of 0.01. We trained the model for 8 epochs. The evaluated accuracy was 75.4%. The mean inference time is 5.4259 $\pm$ 0.1065 ms (mean $\pm$ s.e.), well below 10 ms (Fig.2).

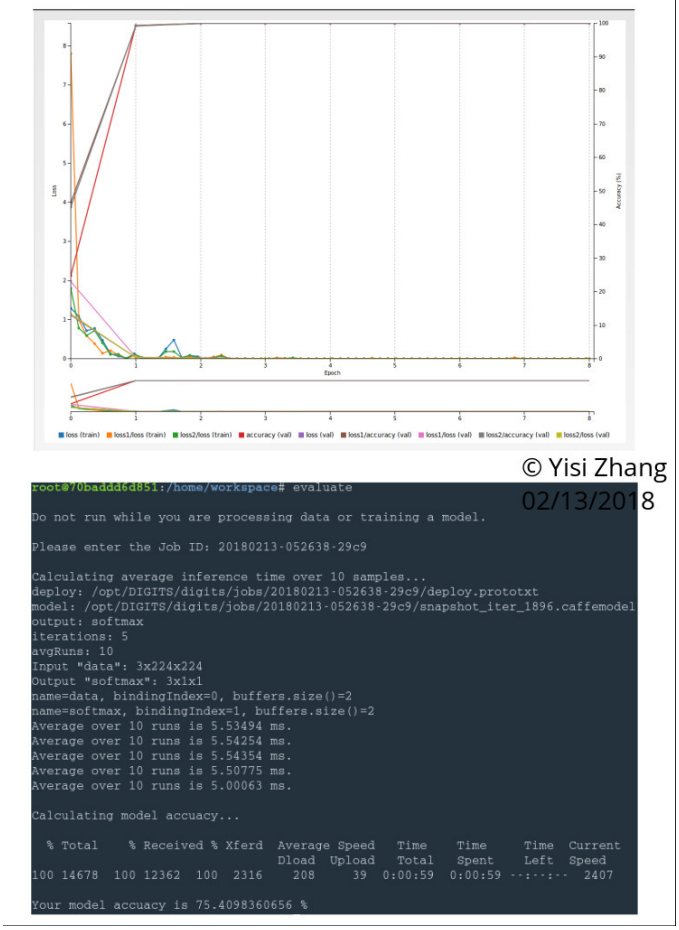

Fig. 2. Inference results of supplied data using GoogLeNet.

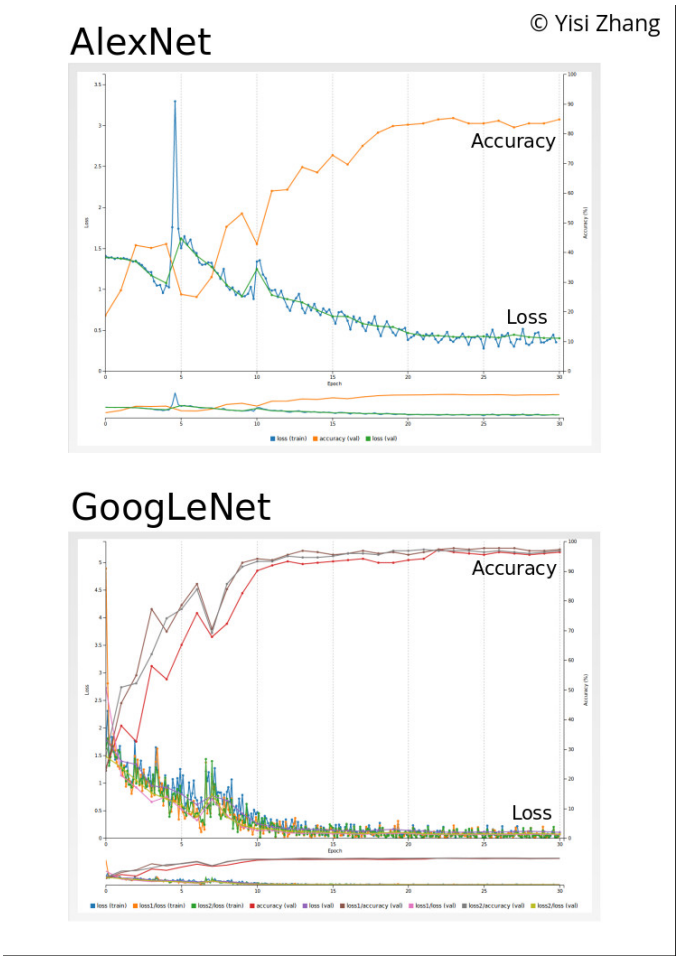

Fig. 3. Training processes of AlexNet and GoogLeNet models.

1 Results

Both models under investigation achieved a $>$ 80% validation accuracy within 30 epochs and reached a plateau. The GoogLeNet outperformed AlexNet in the number of epochs to reach 75% accuracy: it took about 8 epochs for GoogLeNet but 17 epochs for AlexNet. GoogLeNet also achieved a higher accuracy of 96.43% at the 30th epoch compared to AlexNet's 84.82% (Fig. 3).

To test the accuracy of the trained models in recognizing the objects, we randomly selected a set of images from the data set with balanced size for each category. AlexNet achieved a top-1 accuracy of 86.12% and GoogLeNet achieved a top-1 accuracy of 96.0%. Among all the four categories, chairs and wires were predicted best, both were correctly classified above 90% using AlexNet and 100% using GoogLeNet. The worst recognized objects were the slippers, especially with the AlexNet model for only 68.63%. GoogLeNet significantly improved the recognition of slippers to 88.46% (Fig. 4).

Although the GoogLeNet model outperformed AlexNet in classification accuracy, it took longer in the inference time. The mean inference time for AlexNet is 4.2163 $\pm$ 0.0061 ms (mean $\pm$ s.e.) while it took GoogLeNet for 5.4124 $\pm$ 0.1394 ms (mean $\pm$ s.e.) (Fig. 5).

2 Discussion

Both AlexNet and GoogLeNet are convolutional neural network based models. AlexNet consisted of 5 convolution layers and 3 fully connected layers. GoogLeNet consisted of 22 layers deep CNN with a novel modular architecture dubbed "inception" which effectively reduces the number of parameters. Overall, both models achieve high classification accuracy and efficiency.

In this project, GoogLeNet performed significantly better in image classification. The deeper architecture allowed it to extract the features that better describe the objects. In this data set, the slipper images may be more diverse than other categories as it contains photos from 6 very distinct pairs of slippers with various colors, materials and styles. The top and bottom of the slippers were also different in color and shape. Thus, the "concept" of slippers given such variety

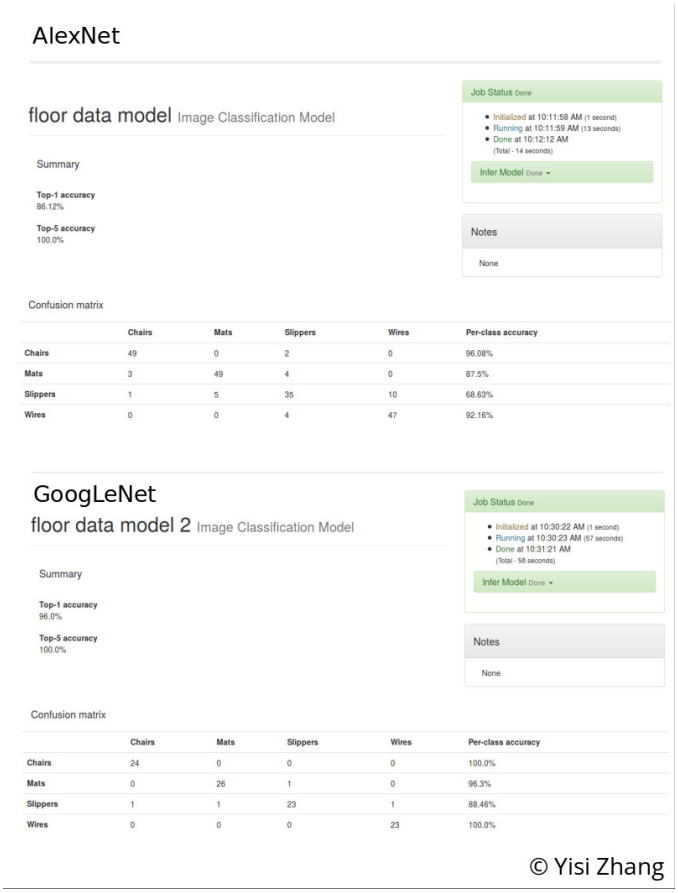

Fig. 4. Accuracy test.

Fig. 5. Inference time.

may require deeper networks to capture and this may be the reason for GoogLeNet to perform better. In a more complicated environment with changing background and lighting condition, a deeper network would be preferred to guarantee the accuracy. In this sense, the GoogLeNet is preferred to complete this task.

The inference time for AlexNet was significantly shorter than GoogLeNet by more than 1 ms. A short inference time is important for the robot to keep updated with the changing environment. Suppose the robotic vacuums move at a speed within the range of 0.1 – 1 m/s, 1 ms difference in inference time would yield a 0.1 – 1 mm difference in spatial resolution. This range should fall far below the spatial distribution of objects in a house. Thus, although the GoogLeNet was slower in the inference time, it should be sufficient for the floor object recognition task. Given the above two reasons, GoogLeNet model would be an optimal model to implement.

3 Conclusion / Future work

In conclusion, this study demonstrated the plausibility for the robotic vacuum to conduct real time object recognition with the deployment of a GoogLeNet model. Compared to other models such as the AlexNet, it achieved much higher accuracy. In addition, it completed inference within reasonable time. Thus, the object recognition feature should be ready to be commercialized in the autonomous robotic cleaning industry.

In the future, a semantic segmentation algorithm may further improve recognizing objects in a changing environment. With this feature implemented, multiple different objects within the same view can be identified and this is a situation closer to reality. Further work can focus on obtaining the training data for semantic segmentation and test the performance and inference time.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, 2012.

[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *Cvpr*, 2015.