Memory Constrained Camera Based Obstacle Detection for Autonomous Vehicle Navigation

Joel Afriyie, Lhito Camson, and Eric Manley
Department of Computer Science and Mathematics
Drake University
Des Moines, IA
joel.afriyie@drake.edu, lhito.camson@drake.edu, eric.manley@drake.edu

ABSTRACT

Machine Learning, a subfield of Artificial Intelligence, in which programs learn from data, has spurred the present day information revolution. These data learning algorithms are the central piece behind some of the world's most sophisticated software programs from stock market predictors, video recognition bots, robust text translations software, and self-driving cars. Evidently, in testing machine learning software for robust safety, it is imperative to create smaller scale devices to mitigate risk of failure and examine constraints. This work investigates using computer vision on autonomous vehicle navigation on a memory-constrained robot. The robot was able to navigate an interior space autonomously without bumping into obstacles using only a camera; furthermore, high classification accuracy was achieved by utilizing a memory-efficient logistic regression model.

INTRODUCTION

The decreasing cost of data storage and overall computing hardware has allowed for machine learning applications to become ubiquitous across all industries and societies. Moreover, the prevalence of machine learning has powered areas of research within classical AI subfields like Computer Vision, Natural Language Processing, information retrieval, and vehicle navigation to achieve better benchmarks. In regards to vehicle navigation, computer vision has recently become an important tool for autonomous vehicle navigation. Examples of this emerging technology include Waymo, Alphabet's self-driving car company [5], and Uber's automated driving fleet [6].

Background

Supervised machine learning algorithms attempt to build predictive models based on data of known examples of the target function being modeled [4]. There are many techniques that machine learning can utilize in order to create this model, and the creation of new algorithms remains an active area of research. When applied to computer-vision applications, the model is trained on labeled images of objects that need to be identified. When new images need to be classified, they are given as input to the previously trained model, and the model predicts which objects appear in the image.

For example, a programmer can develop and train a machine learning model to be able to distinguish faces in a video feed, such that a security camera will be able to pick out a specific person in a crowded area. The programmer would have fed their model several images of the targeted person's face in different settings, as well as faces of non-targeted people. Through the learning process, the model would have eventually learned how to distinguish between the targeted person's face, and other faces. This is just one example of how Machine Learning is being used, but there are many more in use, and more that are currently being researched.

Some popular usages of machine learning that have been accomplished in the popular ImageNet Large Scale Visual Recognition Challenge 2016 are: object detection with provided training data, object localization, object detection from video, and scene classification [3].

METHODOLOGY

For this research, a Raspberry Pi B model with a camera, servomotor, RC wheels, and 1 gigabyte of RAM was utilized. The first step of the work was collecting images to amass a training data set. This was done by using the PiCamera library written in Python3; moreover, a Python script was created to manually maneuver the robot. The initial sample was 640 images, but the final dataset that was used for a majority of the experimental models comprised of 841 images of various interior spaces.

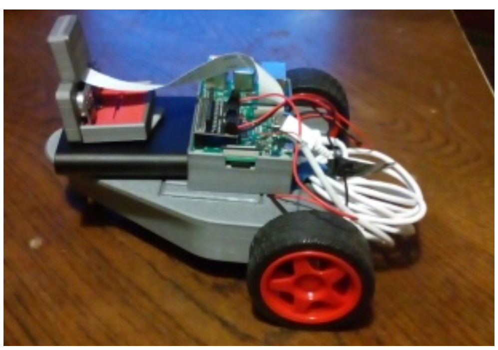

Figure 1: a picture of the Raspberry Pi robot

By using a limited robotic device, there were some modifications of the problem that were employed to make full use of the limited resources. First and foremost, in the state of the art of obstacle detection utilizing computer vision, an image, taken from the camera, is fragmented into several pieces. This allows the robot to employ complicated logic on deciphering, for example, on moving left or right. However, this robot has no such logic due to the limitations of memory. Instead, simpler logic was used for the robot. If the robot is in a space where there are no obstacles then it will travel in a straight line; otherwise, the robot will spin clockwise to avoid the obstacle in its path. This approach created a binary classification problem in which the robot is tasked with classifying an open space or an obstacle.

Initially, a convolutional neural network model was employed on the collected images. A CNN model was chosen because it has been shown to perform well on image classification tasks, in particular for the ImageNet competition [2], and it is an active research area for applications in autonomous vehicle navigation [1]. The CNN network, with an input of 3 RGB channels with an image height and width of 256 is fed into a convolutional layer feature map of 32 units with stride a 3x3 stride. This is then fed into a max-pooling layer of size 2x2. After the max-pooling layer, the aforementioned layers are used once again. Finally, a convolution layer of 64 units with stride 3x3 and a max-pooling layer of size 2x2 are utilized. The final feature map is then flattened into a vector consisting of 64 units (or 64x1 vector). Dropout is applied onto the vector (dropout is a way of preventing over-fitting [8]), which then feeds into the classification output where 1 is for an open space image and 0 for an obstacle.

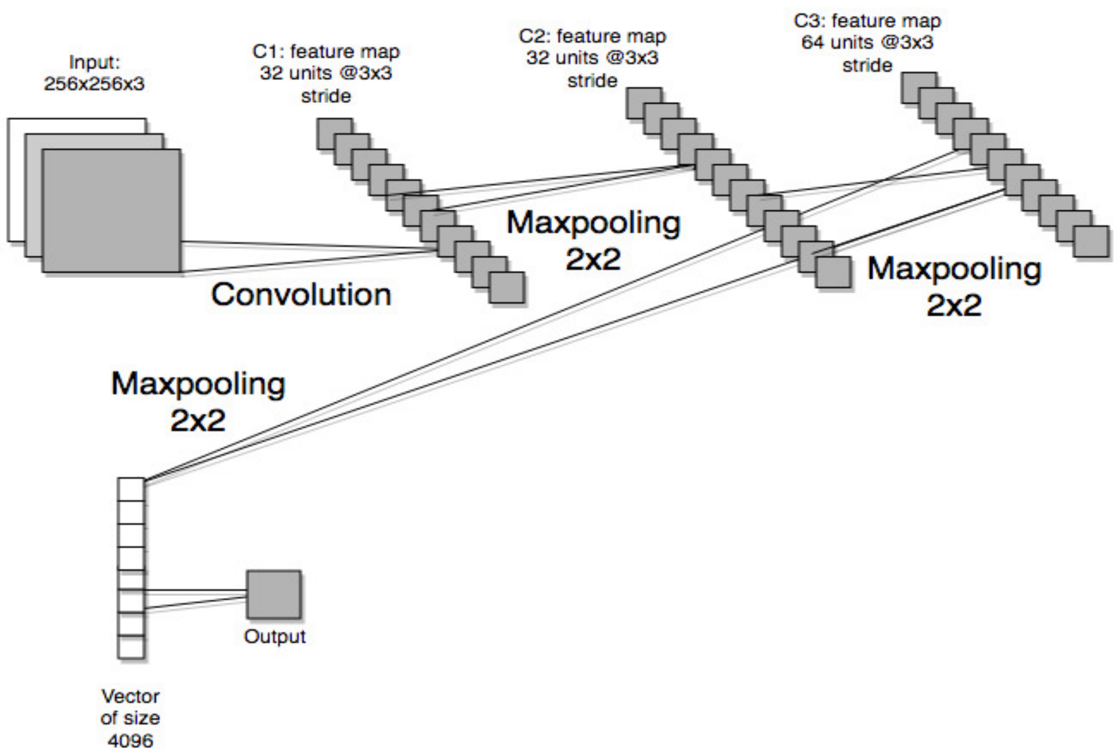

Figure 2: Diagram of Convolutional Neural Network with three feature maps

For several tasks and applications, a convolutional neural network achieves high classification accuracy. However, in this work, due to the limitations of the machine and the camera, it was difficult for the network to generalize well on different pictures; moreover, because of the small set of samples it was difficult for the neural network to converge. Another convolutional neural model was created that allowed for image transformation tricks to expand the amount of samples. Evidently, the use of different convolutional neural networks presented several problems for the robot. These include: the inability of the robot to hold all of the parameters of the neural network in RAM; obstacle and open space images sharing similar feature maps; and time to train on the images. Due to the lack of convergence and low accuracy, other models, such as a Support Vector Machine, were chosen.

A Support Vector Machine (SVM) was used to create a maximal margin on the decision boundary between the open spaces and obstacle images [10]. However, in order to fully utilize the SVM to its best capacity a kernel was employed on the images. A kernel maps inputs into a higher dimensional space and outputs a similarity between data points [7]. In this case, the use of a kernel enables the SVM model to more easily to decipher the dissimilarity between open and obstacle images. By using this method, the kernel removed most of the inseparability between the two classes; furthermore, this allowed for an improvement with regards to the classification accuracy. An SVM with a polynomial kernel, which is commonly known as a poly kernel, achieved the best classification accuracy out of all the models examined.

Although the SVM with a polynomial kernel achieved the best accuracy, it was not used for the classification on the robot due to the size of the model. Other techniques, such as Principal Component Analysis (PCA), were employed on the SVM. PCA reduces the number of components of a piece data while at the same time maximizing the variance within the data. PCA techniques have been shown to work well on image classification tasks [9], and thus it was tried experimentally within this work. Unfortunately, the PCA did not lead to higher accuracy results so other models were examined. Finally, a logistic regression model was used because its performance was similar to the SVM model but its size was much smaller. This allowed the model to fit in RAM on the robot and still maintain a high level of predictive accuracy.

RESULTS

In our experiments, the simpler models performed better. The more complex model, the convolutional neural network without image augmentation techniques, achieved an accuracy of approximately 40%, which is less than a random flipping of a fair coin. The convolutional neural network with image augmentation achieved around 55% percent accuracy with a sample size of 640 images, and a bolstered 59% accuracy when trained with the full sample size of 841 images. The Support Vector Machine with a poly kernel achieved 75% accuracy on the full dataset, which was the best prediction model. However, it was not chosen due to it being 431 MB, which caused issues since the Raspberry Pi could not load the model into memory.

Two Support Vector Machine models were created that used an RBF kernel: one used principal component analysis (PCA), and the other did not. The Support Vector Machine model that used PCA achieved approximately 58% accuracy, whereas the one without achieved 55% accuracy. Finally, a Logistic Regression model achieved around 67% and the model size is 1.6 MB. Due to the second best classification accuracy out of all the models, and the small memory size, the Logistic Regression model was used as the predictive engine behind the autonomous robot.

Model	Classification Accuracy	Data Sample Size	Memory Of Model/Weights
Support Vector Machine with Poly kernel	75%	841	431 MB
Logistic Regression	67%	841	1.6 MB
Support Vector Machine with RBF kernel and PCA	58%	841	31 KB
Convolutional Neural Network with Image Augmentation	55%	640	29.8 MB
Convolutional Neural Network with Image Augmentation	55%	841	29.8 MB
Support Vector Machine with RBF kernel	55%	841	1.05 GB
Convolutional Neural Network	40%	640	29.8 MB

Table 1: Comparisons of accuracy, sample size, and memory of all experimental models in this work

CONCLUSION

Limitations of computational resources and data sample size presents interesting problems on the scope of autonomous vehicle navigation in the future. With this in mind, in regards to computationally limited robots and small data sets, it has been shown that simple classification algorithms are viable predictive engines in a computationally restrained environment. Although state of the art classification accuracy was not achieved, this work offers a baseline in regards to the lower bounds of performance of autonomous robots.

Future Work

The effectiveness of a Support Vector Machine has empirically been shown to offer a high precision of accuracy. To augment the power of a SVM approach and to lessen the memory overhead, one could make a server-client application. The robot sends a GET request along with a captured image, at timestamp *t*, to a server that contains the SVM and other memory intensive models. The server would then send a classification to the robot. This would allow for better models to be used. This would also present interesting research questions of the speed of the network needed for REST communication between the server and the robot for accurate behavior.

REFERENCES

[1] Hwu, T., Isbell, J., Oros, N., & Krichmar, J., A self-driving robot using deep convolutional neural networks on neuromorphic hardware, *Neural Networks, 2017 International Joint Conference on Neural Networks*, 635-641, 2017.

[2] Krizhevsky, A., Sutskever, I., & Hinton, G. E., Imagenet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems,* 1097-1105, 2012.

[3] Large Scale Visual Recognition Challenge 2016 (ILSVRC2016), ILSVRC2016. *http://image-net.org/challenges/LSVRC/2016/results*, retrieved March 02, 2018.

[4] Mitchell, T. M., Machine Learning. New York City, NY: McGraw-Hill, 1997.

[5] Lee B. T., Waymo self-driving trucks are hauling gear for Google data centers, *https://arstechnica.com/tech-policy/2018/03/waymo-moves-beyond-driverless-taxis-with-trucking-program/,* published on March 9th, 2018.

[6] Muoio D, We rode in Uber's self-driving car – here's what it was like, *http://www.businessinsider.com/uber-driverless-car-in-pittsburgh-review-photos-2016-9,* published on September 14th, 2016.

[7] Vapnik, V., Golowich, S. E., & Smola, A. J., Support vector method for function approximation, regression estimation and signal processing, *Advances in neural information processing systems*, 281-287, 1997.

[8] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R., Dropout: A simple way to prevent neural networks from overfitting, *The Journal of Machine Learning Research*, *15*(1), 1929-1958, 2014.

[9] Han, H., Cao, Z., Gu, B., & Ren, N., PCA-SVM-based automated fault detection and diagnosis (AFDD) for vapor-compression refrigeration systems, *HVAC&R Research*, *16*(3), 295-313, 2010.

[10] Cortes, C., & Vapnik, V., Support-vector networks, *Machine learning*, *20*(3), 273-297, 1995.