Undergraduate Thesis

CurbNet: Semantic segmentation of curbs and curb cuts from street imagery

Yvan Putra Satyawan

Examiner: Prof. Dr. Wolfram Burgard
Advisers: Jannik Zürn

Albert-Ludwigs-University Freiburg

Faculty of Engineering

Department of Computer Science

Chair for Autonomous Intelligent systems

July 22$^{\text{nd}}$, 2019

Writing Period

04. 20. 2019 – 07. 22. 2019

Examiner

Prof. Dr. Wolfram Burgard

Second Examiner

?

Advisers

Jannik Zürn

Declaration

I hereby declare, that I am the sole author and composer of my thesis and that no other sources or learning aids, other than those listed, have been used. Furthermore, I declare that I have acknowledged the work of others by providing detailed references of said work.

I also hereby declare, that my thesis has not been prepared for another examination or assignment, either in its entirety or excerpts thereof.

____________________________ ____________________________

Place, Date Signature

Abstract

(TODO: Write this.)

Acknowledgments

First and foremost, I would like to thank...

- advisers
- examiner
- person1 for the dataset
- person2 for the great suggestion
- proofreaders

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

Semantic scene segmentation is a popular research topic in the field of computer vision, and especially important for autonomous vehicles. The ability to semantically understand a scene is especially important for autonomous vehicles and robots to safely navigate an environment. Generally, most implementations attempt to segment road surfaces but in this thesis, we propose the segmentation of curbs and curb cuts to allow safer sidewalk navigation.

The Europa project has resulted in the Obelix robotic platform, which has already been demoed to successfully perform pedestrian navigation [1][2]. We propose to add to this platform the ability to detect curbs and curb cuts using semantic segmentation. The Obelix platform is the result of a joint project to build a robotic platform capable of robotic navigation [1]. **(TODO: Add short description)**.

Our goal is thus to implement a computer vision algorithm capable of the semantic segmentation of curbs and curb cuts using a single camera image. To do so, we will implement a convolutional neural network (CNN) with a traditional encoder-decoder architecture. We will also include prior knowledge to the training, since we can assume that the camera setup for the Obelix robot will remain relatively similar throughout its lifespan.

We will begin by discussing the motivation behind this thesis, followed by a disucssion of related works and the background. Then the approach will be discussed in detail along with the experiments and results. Finally, a discussion of potential future research will be presented followed by the conclusion.

1.1 Motivation

(TODO: Do this.)

2 Related Work

CurbNet uses semantic scene segmentation to identify curbs and curb cuts. As such, related works can be divided into two categories: semantic segmentation and curb detection. The following is a discussion of relevant related works.

2.1 Semantic Scene Segmentation

There are many works in the field of semantic scene segmentation in recent years, both discussing object scene segmentation and road segmentation. The field of semantic segmentation using trainable neural network models started in 1989 with the pioneering work of Eckhorn et al. and their paper describing how the visual cortex of a cat functions and its implications for network models [3]. This early method used a pulse coupled neural network, which produced synchronous bursts of pulses, effectively grouping the neurons by phase and pulse frequency, which can then be analyzed for feature extraction.

In the same year, Y. LeCun et al. developed the first algorithm to use backpropagation and convolutional neural networks to identify and classify images [4]. Their paper titled "Backpropagation Applied to Handwritten Zip Code Recognition" proposed that using convolutional filters directly on an image input and training using backpropagation could reliably classify images into predetermined classes. This was the first network architecture that took a normalized image as input and returned the image class directly as an output. LeCun et al. also showed that using backpropagation to learn the convolutional filter coefficients performed significantly better than hand selected coefficients. This pioneering work set the stage for modern day image classification and segmentation algorithms using CNNs and automated learning.

"Very Deep Convolutional Networks for Large-Scale Image Recognition" by Karen Simonyan and Andrew Zisserman was one of the earlier works to show that a deeper network setup could result in very high accuracy image classification [5]. Their setup improved upon the use of convolutional neural networks by proposing instead to use small (3×3) convolutional filters and changing the depth to 16-19 layers. This network is now commonly known as VGG16 - VGG19, the number representing layer depth. This simple change in architectural designed resulted in their architecture securing first and second place in the ImageNet Challenge 2014 localization and classification tracks respectively. Unfortunately, these very deep networks tended to overfit on smaller datasets and were difficult to train.

"Deep Residual Learning for Image Recognition" by Kaiming He et al. proposed a solution to this problem [6]. They reformulate the layers as learning residual functions and use the layer inputs as references. This architecture is now commonly known as ResNet. By doing so, they were able to empirically show that their residual networks are easier to optimize and have lower complexity despite being up to eight times deeper than VGG networks. ResNet was able to obtain a 28% relative improvement on the COCO object detection dataset compared to previous methods [7].

"Fully Convolutional Networks for Semantic Segmentation" by Jonathan Long et al. took these classification networks and added fully connected layers to take the encoded features and use it for semantic segmentation [8]. This paper laid out a novel insight to the problem of image segmentation as nothing more than a dense image classification problem. The proposed network architecture is now commonly referred to as FCN. In essence, they proposed that image segmentation is nothing more than image classification on a per-pixel basis. As such, previously developed CNNs for image classification could be used and indeed, they implemented their model based on VGG16. Their network based on VGG16 was able to outperform competing state-of-the-art approaches on the PASCAL Visual Object Challenge dataset by a relative margin of 20%.

DeepLab v3+ was proposed in "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation" by Liang-Chieh Chen et al. and improves on FCN by refining the segmentation, especially along object boundaries [9]. DeepLab v3+ became the basis of the network we used for CurbNet. Using pyramid pooling and an improved encoder-decoder architecture, DeepLab v3+ achieve 89.0% accuracy on the Cityscapes dataset [10].

2.2 Curb Detection

Research into curb detection is quite plentiful, with even one work from ETH Zürich being made specifically for the Obelix robotic platform [11]. Curb Detection for a Pedestrian Robot in Urban Environments" by Jérôme Maye, Ralf Kaestner, and Roland Siegwart used the LIDAR sensors that Obelix has to map the world around it. Using this point cloud data, a virtual representation of the environment can be computed and horizontal planes from the scene extracted. By detecting sudden changes in the vertical position of horizontal planes, a curb can be implicitly identified. This relies on the assumption that curbs take the form of vertical planes connecting two horizontal planes. Unfortunately, this work does not take into account curb cuts, which may not necessarily form a significant vertical height difference and are usually sloped.

The only work we were able to find that specifically address curb cuts was "WalkNet: A Deep Learning Approach to Improving Sidewalk Quality and Accessibility" by Andrew Abbott et al. [12]. This paper discusses the use of a deep neural network to classify images in which curb cuts existed. Their goal was the use of Google Street

View data to map which intersections in a city already had curb cuts and which didn't. This data would then be supplied to city governments to provide relevant information regarding sidewalk accessibility and quality. Unfortunately, this work only classified images which contained curb cuts and the neural network architecture they used was not described in depth. As such, we were unable to use any part of their research, despite being one of the few papers published regarding curb cuts.

Thus far, there seems to be no works related directly to the goal of this thesis; the semantic segmentation of curbs and curb cuts using computer vision.

3 Background

The basis of CurbNet is semantic scene segmentation using convolutional neural networks. As such, it is imperative that the reader has an understanding of the principles behind semantic scene segmentation and convolutional neural networks. This chapter on the background of this work will discuss semantic scene segmentation, machine learning with artificial neural networks and convolutional neural networks, curb segmentation, loss functions, and network optimizers.

3.1 Semantic Scene Segmentation

Unlike image classification, which predicts what an entire scene is, semantic scene segmentation is the processing of an image and assigning class labels to each individual pixel [13]. This allows for finer details and adds the ability to locate and classify multiple objects in a scene. For example, the image in figure **(TODO: add image)** has been segmented and the result is figure **(TODO: add segmented)**. Each pixel of the image has been assigned an associated class, in this case, **(TODO: add classes segmented)** This allows a computer or program to understand what objects are in the image it is shown.

By segmenting an image in this way, the program can interpret the scene or its environment semantically similarly to the way a human might. For example, by receiving the segmented image, a program can identify that there are line markings on the road and that there is a vehicle in front of it. The segmentation of images in this way is essential in many robotic implementations as it allows further higher level processing of the scene.

In our case, being able to segment and identify curbs in a scene would allow Obelix to map the location of the surrounding curbs and curb cuts, allowing for the safe traversal from sidewalk to street level via curb cuts.

Over the past few years, state-of-the-art methods in image segmentation have relied entirely on deep learning using CNNs to achieve better results.

3.2 Artificial Neural Networks

Artificial neural networks are a computing system inspired by biological neurons. Neural networks are comprised of neurons which are capable of taking any number of numerical inputs and outputs a numerical value. Mathematically, a single neuron is a time dependent function. The function of a neuron j receiving input $p_j(t)$ and producing output $o_j(t)$ is composed of the activation $a_j(t)$, potentially a threshold Θ_j, an activation function f that returns the activation at time $t+1$, and an output function f_{out}. The activation $a_j(t)$ can also be considered the neuron's state and is dependent on the time parameter t. The threshold Θ_j, if it exists, is fixed unless a learning function changes it. The activation function f calculates $a_j(t+1)$ given $a_j(t)$, Θ_j, and the network input $p_j(t)$ and can be defined as:

$$a_j(t+1) = f\left(a_j(t), \Theta_j, p_j(t)\right) \tag{1}$$

The output function f_out computes $o_j(t)$ based on $a_j(t)$ and is defined as:

$$o_j(t) = f_out\left(a_j(t)\right) \tag{2}$$

The output function is usually simply the identify function $f(x) = x$. Many activation functions exists and are used including the identity function and the rectified linear unit (ReLU), defined as:

$$f(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases} \tag{3}$$

Between each neuron in the network are connections which transfer the output of neuron i to neuron j. Each of these connections are assigned a weight and potentially a bias term and is computed by the propagation function to provide a neuron its input $p_j(t)$. This typically is defined as:

$$p_j(t) = \sum_i o_i(t) w_{ij} + w_{0j} \text{ , where } w_{0j} \text{ is the bias, if it exists.} \tag{4}$$

Learning occurs by using an algorithm to modify the parameters of the neural network.

Deep neural networks are so called due to having multiple "hidden" layers between the input and output. These hidden layers are not visible to the end user and their values are typically never accessed directly.

3.3 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a specific class of deep neural networks and have been found to perform exceptionally for image analysis, such as image classification and segmentation. The inspiration for CNNs come from biological processes to simulate the orgainization of the visual cortex in animals [14]. Convolutional layers are the core build blocks of CNNs. These convolutional layers consist of a set of learnable filters which are convolved across the entire input and computing the dot products between the different entries of the filter. This allows the layer to activate when it detects a specific type of feature at some point in the input.**(TODO: convolution image)** By using multiple convolutional layers, higher level features can be extracted and a feature map generated.

For semantic scene segmentation applications, the convolutional layers that make up the CNN are called the feature encoder. The output of these convolutional layers are then fed into one or more fully connected linear layers to produce a classification of each pixel in a secene. These fully connected layers are called the decoder. The resulting classification produced is a probability of each pixel being a certain class.

3.4 Curbs and Curb Cuts

Curbs are the edges of roads, or the separator between the road and some other area, usually a pedestrian sidewalk, and are usually stone or concrete. Curb cuts are "cuts" in the curb that allow a pedestrian sidewalk to have a gentle slope down to street level. Originally, these cuts were made to allow accessibility access, especially for those requiring wheelchairs, as to a wheelchair user, a curb is as good as a wall in terms of traversibility, as was discussed in the article "Curb Cuts" by Cynthia Gorney and Delaney Hall. These curb cuts first started appearing fifty years ago from the efforts of activist Ed Roberts and his push to make city streets more accessible. **(TODO: Include picture of a curb cut)**

3.5 Curb Segmentation

A discussion of the state-of-the-art of curb and curb cut segmentation would be moot, as there is no work we could find specifically discussing this topic.

There are quite a few papers regarding the identification and localization of curbs using LIDAR sensors, but none that specifically tackle the problem of curb cuts and none that use computer vision approaches. The challenge for curb segmentation is the relatively small size of curbs in relation to everything else in the scene. As such, there is a heavy class imbalance. With such an unbalanced dataset, it is possible for a neural network to simply optimize towards classifying no curbs at all, since this would produce a reasonably low loss.

Furthermore, the segmentation and differentiation between curbs and curb cuts using a single image with no depth information is quite difficult, as visually, curbs and curb cuts can seem very similar.

3.6 Loss Functions

The loss function l maps the output of the neural network $\hat{y}$ and the target y onto a real number and represents the "cost" of an output. The goal of learning is to minimize this cost value, known as the network loss. Effectively, the loss function measures the difference between the predicted value and the target. In our case, the target value is the ground truth labeling of an image and the output is the predicted labeling from the network. There are many different loss functions used, but for the purposes of image segmentation, the most commonly used function is cross entropy loss or weighted cross entropy loss.

Cross entropy loss is the loss function most commonly used for image segmentation networks. The weighted variant, known simply as weighted cross entropy loss, is the basis of the custom loss function we use for CurbNet and is defined as:

$$\text{loss}(x, class) = weight[class]\left(-x[class] + \left(\sum_j \exp(x[j])\right)\right) \tag{5}$$

where x is the predicted labeling, $class$ is the ground truth labeling, $weight$ is the weight given to each class, and j is the individual pixels in x.

4 Approach

The approach usually starts with the problem definition and continues with what you have done. Try to give an intuition first and describe everything with words and then be more formal like ‘Let g be ...’.

4.1 Problem Definition

Start with a very short motivation why this is important. Then, as stated above, describe the problem with words before getting formal.

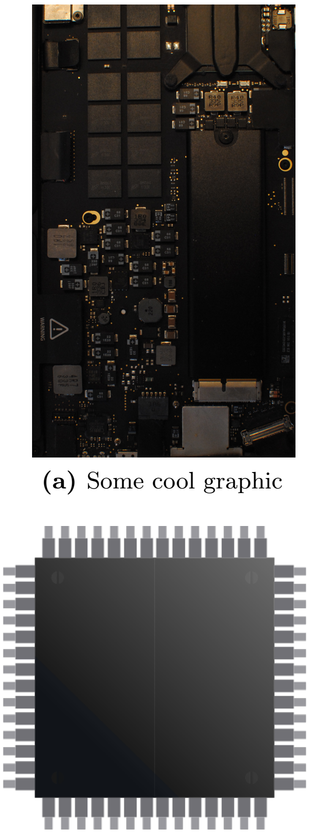
4.2 First Part of the Approach

4.3 N-th Part of the Approach

5 Experiments

Type	Accuracy
A	82.47 ± 3.21
B	78.47 ± 2.43
C	84.30 ± 2.35
D	86.81 ± 3.01

Table 1: Table caption. foo bar...



(a) Some cool graphic

(b) Some cool related graphic

Figure 1: Caption that appears under the fig Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

6 Conclusion

Bibliography

[1] "The european pedestrian assistant," http://europa.informatik.uni-freiburg.de 2009.

[2] R. Kümmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard, "A navigation system for robots operating in crowded environments," in *2013 IEEE International Conference on Robotics and Automation*, IEEE, May 2013.

[3] R. Eckhorn, H. J. Reitboeck, M. Arndt, and P. Dicke, "Feature linking via stimulus - evoked oscillations: Experimental results from cat visual cortex and functional implications from a network model," in *International 1989 Joint Conference on Neural Networks*, IEEE, 1989.

[4] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, pp. 541–551, Dec. 1989.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[7] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014.

[8] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[9] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," 2018.

[10] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[11] J. Maye, R. Kaestner, and R. Siegwart, "Curb detection for a pedestrian robot in urban environments," in *Proceedings - IEEE International Conference on Robotics and Automation*, 05 2012.

[12] A. Abbott, A. Deshowitz, D. Murray, and E. C. Larson, "Walknet: A deep learning approach to improving sidewalk quality and accessibility," *SMU Data Science Review*, vol. 1, no. 1, 2018.

[13] G. Seif, "Semantic segmentation with deep learning," *Towards Data Science*, September 2018.

[14] K. Fukushima, "Neocognitron," *Scholarpedia*, vol. 2, no. 1, p. 1717, 2007. revision #91558.