Supervised Learning Methods for Vision Based Road Detection

Vivek Nair VNAIR611@STANFORD.EDU
Nikhil Parthasarathy NIKHILP@STANFORD.EDU

Abstract

One of the most important problems in the development of autonomous driving systems is the detection of navigable road. This paper explores a formulation of this issue as a supervised learning problem. Given highway video taken by a frontal camera, a naive method for generating positive and negative test images is proposed in order to implement binary classification. Two promising classification systems are implemented and compared. The first uses a linear SVM to classify image sections featurized by Segment-Based Fractal Texture Analysis (SFTA). This approach is compared to supervised learning via a multilayer convolutional neural network (CNN). Both methods achieve very high accuracy but the CNN is shown to perform slightly better due to higher specificity.

1. Introduction

With the advent of environment-aware automobiles (e.g. adaptive cruise control and Google's self-driving vehicle), developing cheap and efficient algorithms for detecting road will be crucial in making the latest AI technology accessible to the average car owner. In particular, road detection is complicated by different obstacles such as road markings (e.g. lane dividers and car-pool signs) and light differences (e.g. trees casting shadow onto the road). For this project, we want to answer the following question: can we accurately detect road on a highway, even given the presence of various obstructions?

1.1. Related Work

In recent years, there has been increasing work done in the area of drivable road detection. Given the large number of negative examples that span an extremely large set of categories (vehicles, pedestrians, barriers etc.) and the difficulty of obtaining labeled training examples, early vision-based road detection systems avoided supervised learning models for road detection. Instead much of this early work implemented unsupervised image segmentation using basic color, edge features, and other rules Hu et al. (2004) These unsupervised learning systems often created complex appearance models for road detection by segmenting and clustering individual pixels from many positive road examples Alvarez et al. (2013). Methods have also been explored to improve the robustness of such models by specific feature engineering to capture lighting-invariance Alvarez & Lopez (2011).

Outside the realm of unsupervised classification, there has been research into self-supervised online learning to continuously update a classification model (such as an SVM) for road detection Zhou et al. (2010). Purely supervised approaches have also been implemented, but these scene segmentation algorithms are usually trained on general datasets such as LabelMe and then applied to the specific domain of road detection. Alvarez et al., 2012 specifically attempts to learn from noisy machine generated labels with a convolutional neural network.

1.2. Objective

As mentioned above, the methods for road detection mostly focused on unsupervised and self-supervised models. Even the self-supervised systems were trained either on small sets of manually labeled examples or on general datasets that are not domain specific. The goal of this study was to turn the problem of road detection into a completely supervised learning task. The rest of this paper is structured as follows. In Section 2, the dataset used is defined, along with an explanation of a process used to generate positive and negative labeled training examples. Section 3 details the different feature encodings and classification methods that were implemented. Section 4 discusses the performance of all implemented systems and specifically compares the results of the two most promising approaches (Segment-based Fractal Texture Analysis and Convolutional Neural Networks). Section 5 discusses the limitations of this work and suggests areas for improvement and future work.

2. The Dataset

Tao Wang, PhD student in Stanford AI Labs (SAIL), provided us with 20 hours of highway driving recordings to train, validate, and test our models. All of the recordings were taken on highways 280 and 880, which span large areas of the Bay Area.

2.1. Generating Labeled Training Examples

Generating labeled examples for road detection is a difficult problem for many reasons. In particular, hand-labeling thousands of negative examples can take a considerable amount of time investment. As a result, most vision based road detection systems are not completely supervised. We wanted to explore the performance of a supervised system, given a naive method for generating significant labeled training examples.

We generated our positive examples by always assuming that the lower cross-section of the image is road. This is a reasonable assumption made by many studies that attempt to perform similar road detection on images. In Figure 1, for example, this cross-section area is defined by the red rectangular box. Unlike many other studies, however, we utilized a similar idea for negative examples. We generated our negative examples by taking cross-sections of the image that do not overlap with the positive cross-section area (as defined by the rectangular box above). A negative cross-section example is shown in Figure 2. While this negative example generation is not perfect, we found that this approach provided a reasonable dataset to use for training at a fraction of the time.

Another issue in road detection is the fact that perspective changes the texture and look of the road across the image. In order to account for this, we took various subsections from the original cross-sections, and resized them to a fixed input size. These multiple resizings are a very simple approximation for the perspective differences within the image.

For this study, 9018 training examples were used, evenly distributed between positive and negative examples. The test set was composed of 5714 examples, evenly distributed between positive and negative examples. In order to make the training and test sets as different as possible, we used images taken on highway 280 for training and used images taken on highway 880 for testing.

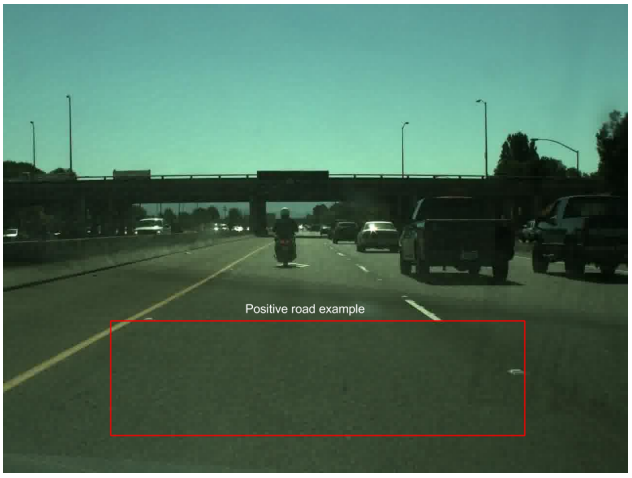

Figure 1. Example highway image taken with positive example bounding box

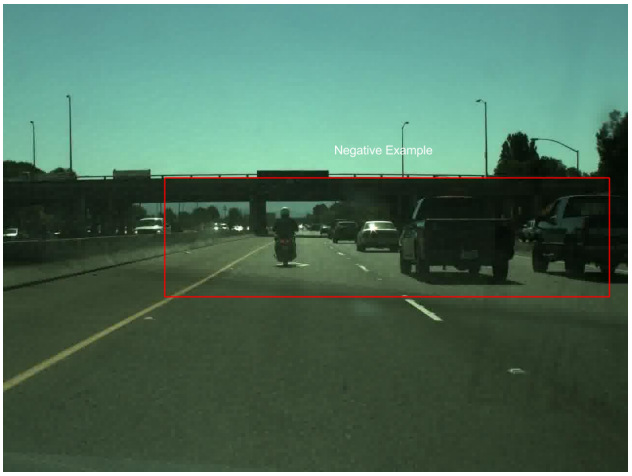

Figure 2. Example highway image taken with negative example bounding box

3. Methods

3.1. Baseline Soft-Margin SVM Classifiers

In order to obtain a simple end-to-end system, we implemented three SVM classifiers, encoding the images in three different naive ways.

1. We computed the average R, G, and B values for a given image and used those three values as a 3-dimensional feature vector.
2. We computed the average R, G, and B values for every 800 pixels in an image (3 feature values per 800 pix-

els). Given that each image has dimensions 1280 by 960, each image is represented as a 1536 * 3 = 4608 feature vector.

3. We first ran k-means on each image to normalize for RGB discrepancies and then applied the same batch averaging as in the second encoding.

We then ran a soft-margin SVM on these three different constructed feature vectors.

3.2. Segment-based Fractal Texture Analysis (SFTA)

Throughout our project, we researched numerous texture classifications processes to capture the intrinsic properties of road (e.g. gravel patterns, non-metallic surfaces). We found that analyzing the fractal dimension of a certain texture, essentially how the detail of a texture pattern changes with the scale of the pattern, could potentially yield high accuracy classifications.

Specifically, we researched and implemented the Segment-based Fractal Texture Analysis algorithm, which decomposes images into various thresholded images using several sets of lower and upper threshold values. Our implementation was based on the implementation in Costa et al., 2012. The two-threshold segmentation was applied using the following representation:

$$I_b(x,y) = \begin{cases} 1 & : t_l < I(x,y) < t_u \\ 0 & : otherwise \end{cases} \quad (1)$$

These thresholded images are used to extract the fractal dimension. Since images with more jagged edges and prominent color differences tend to have higher fractal dimension, images with more uniform texture properties (in this case road) will have closer fractal dimension.

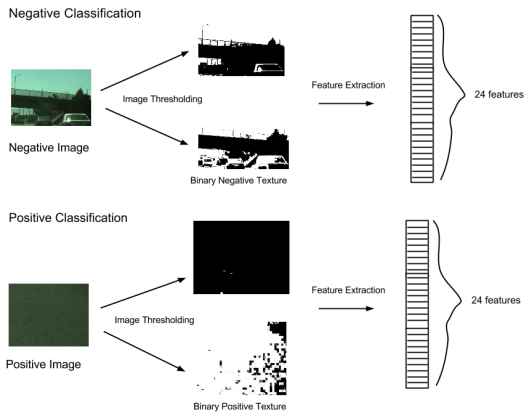

Figure 3. Diagram of the SFTA algorithm

3.3. Convolutional Neural Networks (CNNs)

The above methods all center around the idea of obtaining the right feature representation for an image and then applying a standard classifier such as an SVM to classify the data based on these input features. In this section, we describe an approach that does not require specific feature engineering but rather uses a multi-layer network to automatically obtain higher-order features that are used in classification. This method is based on Convolutional Neural Networks (LeCun & Bengio, 1995). The basic idea behind CNNs can be seen in the figure below: Our CNN implementation utilized the Theano python library for deep learning and was based on the LeNet CNN that was first used for detection of handwritten digits. The input to the CNN was a 128x128 fixed input image. This image was featurized by converting each image's 128x128 array into a 5052x1 vector. The implementation that we used consisted of two alternating convolution subsampling layers (with max-pooling) followed by a logistic regression output layer. The learning rate was decayed from an initial 0.01 with a 0.999 decay rate.

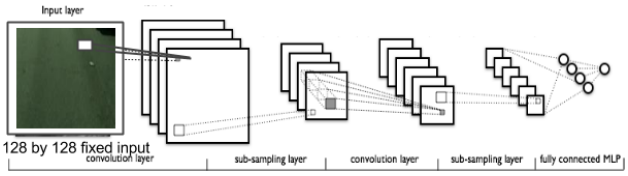

Figure 4. Diagram of the CNN Model

4. Results

4.1. Overview

The results of the three baseline models, SFTA, and CNN are summarized in the table below:

Table 1. Accuracies of All Implemented Systems

Model	Prediction Accuracy
SVM RGB Average over Full Image	0.587
SVM RGB Batched RGB Average	0.602
SVM with K-Means Normalized Images	0.537
SFTA Texture Classification	0.984
CNN Classification	0.987

As expected, the three baseline classifiers had high testing errors because of the naive methods that were used to featurize the images. In contrast, both the texture analysis and the CNN implementations achieved high performance. The specific results for each of these methods and further anal-

ysis is provided in the following sections.

4.2. SFTA Results and Discussion

Using these extracted features from the SFTA algorithm, we ran regular soft-max SVM with no kernel, the RBF kernel (Radial Basis Function), and a quadratic polynomial kernel on the constructed feature vectors with different box constraint parameter values. Our SFTA algorithm achieved optimal performance at C = 4 with the regular linear order features. This shows that the texture feature dataset is best modeled by basic linear decision boundaries rather than complex polynomial models.

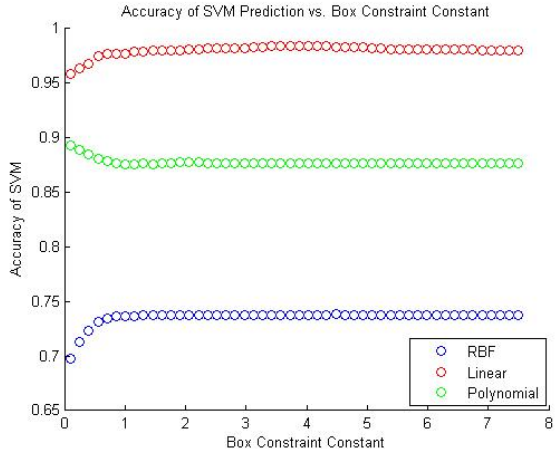

Figure 5. SFTA Accuracies for Different SVM Kernels

4.3. Failure Cases and Potential Issues

There were two main issues with the SFTA texture classification algorithm. Primarily, the system classified many false positives for areas of the image that had relatively low fractal dimension. Specifically, if the image section had very few edges and was relatively uniform in texture (e.g. sky), SFTA misclassified the section as road. To see an illustration of this issue, consider the resulting thresholded images for a certain section of sky and road. The resulting thresholded image yields fractal dimensions that are relatively close. This can be seen in Fig. 6 . To see a specific example of this issue we took a single image and created sliding windows across the image. Each window was classified using the SFTA algorithm and a heatmap was generated and overlayed on top of the image (see Fig. 7), indicating our predictions. While the algorithm successfully classified road sections and obstacles, it also classified the sky as road.

To combat this issue, we hope to encode more RGB features into the SFTA vector to bias the classification to color

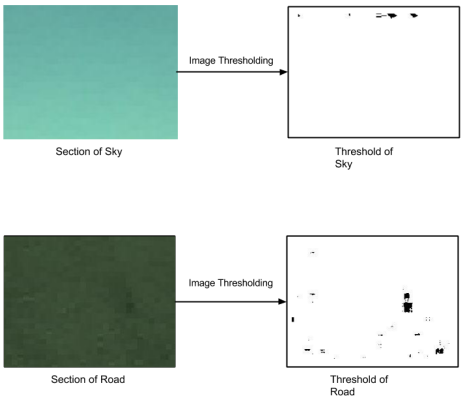

Figure 6. thresholded images for sky and road

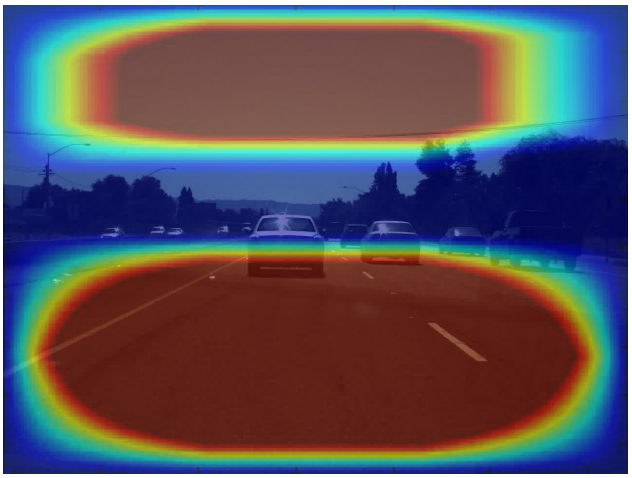

Figure 7. heatmap of a single image classified using SFTA

properties and not just texture. We also hope to incorporate more training examples with just sky because many of our negative examples were not solely sky.

Conversely, the system classified many false negatives for road image sections that had significant amounts of visual noise (e.g. large carpool signs, lanes, and cast shadows from buildings). For an example, see 5.In most cases, however, we found that the density of road that was classified correctly around our false negatives mitigated the misclassification. Nonetheless, in order to improve our accuracy, we hope to use contrast normalization techniques on our examples to reduce the sharpness of road markers and cast shadows. For positive examples with significantly high color and texture variance (bridge shadows), we will need to use more sophisticated models that gain context from

previous images.

4.4. CNN Results and Discussion

Using the resized 128x128 fixed input images, the CNN was run on the dataset described in 2. The best model converged after 90 epochs with the learning rate defined earlier. The model achieved the best performance of any of the systems we tested, with 0.987 accuracy. The CNN and SFTA systems had similar performance so we decided to look at their respective results more closely. The corresponding sensitivities and specificities of each model are summarized in the following table:

Table 2. Sensitivity and Specificity of SFTA and CNN

Model	Sensitivity (TP/(TP+FN))	Specificity (TN/TN+FP)
SFTA	0.999	0.964
CNN	0.980	0.991

In general, the CNN had similar failure cases and issues as those mentioned above for the SFTA algorithm. However, there were some important differences. The CNN actually had a lower sensitivity or more false negatives. This difference could be due to many reasons but no pattern was observed when looking at the additional failures. The higher accuracy of the CNN though, can be attributed to the much higher specificity. This indicates that the CNN had less false positives. This result seems to make sense because the CNN takes into account color information (as opposed to grayscaled images) and is not susceptible to the same fractal dimension issues that SFTA had.

5. Conclusions and Future Work

Currently, we achieve high performance with both SFTA (98.4%) and CNN (98.7%). Both models suffer from similar limitations, however, namely with images that have shadows cast from bridges and roadside buildings. Such an example is given below:

Figure 8. Bridge Failure Sequence

In this sequence of images, the bottom black section of the third image is classified incorrectly, so the question is, can we use the previous images in the sequence to inform our classification in the third image? Much of our future work will involve using sequential images to increase our confidence in road detection at later segments in the video. From an implementation perspective, we can quantify this confidence, derived from a fixed number of prior images, and use this value as an additional feature in both SFTA and CNN. Additionally, as mentioned earlier, we hope to use color information to help improve the SFTA model. We also hope to optimize the CNN and perhaps use it for not only supervised classification, but also unsupervised learning of new features.

6. Acknowledgements

We would like to thank our mentor Tao Wang for providing us with the dataset and guidance throughout this project.

References

Alvarez, J. and Lopez, A. Road detection based on illuminant invariance. *IEEE Transactions on Intelligent Transportation Systems*, 2011.

Alvarez, J., Salzmann, M., and Barnes, N. Learning appearance models for road detection. In *IEEE Intelligent Vehicles Symposium*, pp. 423–429, 2013.

Alvarez, Jose M., Gevers, Theo, LeCun, Yann, and Lopez, Antonio M. Road scene segmentation from a single image. In Fitzgibbon, Andrew, Lazebnik, Svetlana, Perona, Pietro, Sato, Yoichi, and Schmid, Cordelia (eds.), *European Conference on Computer Vision (ECCV 2012)*, volume 7578 of *Lecture Notes in Computer Science*, pp. 376–389. Springer, 2012. ISBN 978-3-642-33785-7.

Costa, Alceu Ferraz, Humpire-Mamani, Gabriel, and Traina, Agma Juci Machado. An efficient algorithm for fractal analysis of textures. *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images*, 0:39–46, 2012. ISSN 1530-1834. doi: http://doi.ieeecomputersociety.org/10.1109/SIBGRAPI.2012.15.

Hu, M., Yang, W., Ren, M., and Yang, J. A vision based road detection algorithm. In *Proceedings of the 2004 IEEE Conference on Robotics, Automation, and Mechatronics*, pp. 846–50, 2004.

LeCun, Y. and Bengio, Y. Convolutional networks for images, speech, and time-series. In Arbib, M. A. (ed.), *The Handbook of Brain Theory and Neural Networks*. MIT Press, 1995.

Zhou, S., Gong, J., Xiong, G., Chen, H., and Iagnemma, K. Road detection using support vector machine based on online learning and evaluation. In *IEEE Intelligent Vehicles Symposium*, pp. 256–61, 2010.