Road estimation

Bharath R. Vellaboyana
1001268200
bharath.vellaboyana@mail.utoronto.ca

Abstract

Road estimation can be a useful part of an autonomous driving system. In this project I estimated the road area in an image using an artificial neural network which was trained using KITTI database. Features were selected to obtain good generalizing ability. Performance statistics are provided for training, validation and test data sets for three kinds of roads.

1 Introduction

Advanced driver assistance systems are being increasingly deployed in the vehicles around the world. These systems use both active and passive sensors to understand the environment. One such passive input can be images from a camera. Estimating the road area and then the ego-lane of the vehicle from such an image form an important part of that understanding, as it reduces the area where other vehicles [1] and pedestrians [2] are to be detected in.

In this project an artificial neural network was used to estimate the road area in an image. Features were selected to provide better ability to discriminate between examples. Different models of the neural network were studied to find the one that provided better learning and generalizable abilities.

The selected features and the algorithm provide an acceptable performance when compared to the submissions in the KITTI website[1]. The results can be seen from the table in section 4.

In section 2 we will see how the features were selected. Then in section 3 different models will be compared. In section 4 performance statistics and other results will be provided for the chosen model from the section 3 and observations will be provided. Finally, in the section 5 we will conclude the paper by providing a summary.

2 Feature selection

An image in the KITTI dataset [3] has $375 * 1242$ pixels. Considering that these are a lot of examples (which can slow down the training process) we use SLICO super pixels [4] to reduce the number of examples and also to obtain additional information such as colour histogram.

The SLICO algorithm divides the image into contiguous areas which have similar characteristics not unlike K-Means algorithm. From these we can obtain histograms of colour distribution and texture distribution. Images have been divided into 500 super pixels.

As roads everywhere have nearly same colour characteristics they can be used as features. There are other authors [5] who considered the standard RGB values of images as features while focussing on the classification part of the algorithm. The colour characteristics were obtained for each super pixel with each bin of the histogram (one histogram for each colour channel R,G and B was constructed) as a feature.And the whole range of 0 to 255 (for an image) was divided into ten bins for this purpose. But it has been observed that most of the pixels in any given super pixel accumulated in the first one

[1]http://www.cvlibs.net/datasets/kitti/eval_road.php

or two of the bins, thus reducing our ability to differentiate. In order to avoid this problem, the range from 0 to 100 was divided into seven bins while the other three bins got equal parts of 101 to 255.

Now we should see that roads have similar texture characteristics too. So these can be used as a feature. This was achieved by using the MATLAB function *rangefilt*, which provides the range (maximum-minimum) in the local neighbourhood of 3 by 3 pixels. The distributions of these values too were used in the form of histograms. The histogram bins had unequal ranges as problems were encountered which were similar to those faced with colour histograms. It was observed that introduction of texture features improved the ability of learning and generalization of the neural network. This can be seen from the improvement in F1 measure for the road area of training set from 0.894857 to 0.922458 and that of validation set from 0.892574 to 0.918360. Both sets were from the Urban Marked Multiple lanes dataset.

Besides these features location of the super pixel was also used as a feature. Shadows in images can reduce our ability to learn and generalize. So, in order to reduce it a method proposed in [6] was studied.

Before seeing the method let us assume R, G and B are the standard colour channels and let $r = log(R/G)$ and $b = log(B/G)$ be the corresponding log chromaticity values using the G channel as the normalizing channel. Under some reasonable assumptions about the light source, if we were to plot the values of r and b (of an image containing shadows) against each other then the resultant scatter plot should form parallel lines (like line h) as shown in Figure 1. The line l_θ is to be perpendicular to them where θ is the *illuminant invariant angle* of the camera. But when such plot was done on one of the images from the dataset, the lines were found to be not in parallel as can be seen from the close up of the plot in Figure 2. We could have constructed illuminant invariant image which did not have the effects of shadows if we could find the θ. But as we can see, this is not to be.

Figure 1: Intended plot

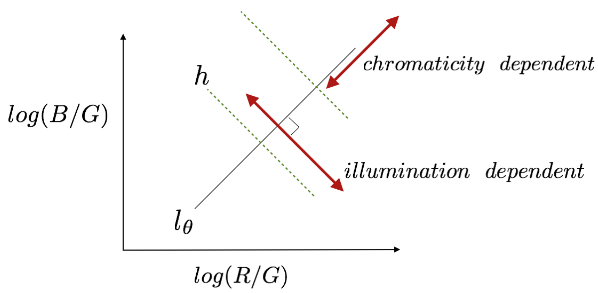
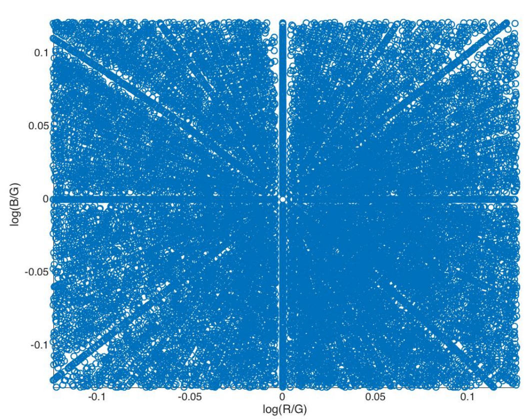


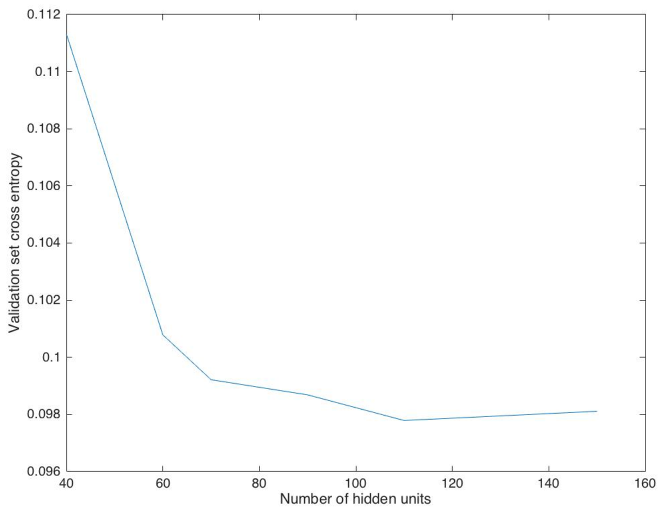
Figure 2: Actual plot



With these selected features training (60%), validation (10%) and test (30%) sets were constructed for the three KITTI data sets. The KITTI datasets are as follow: 'Urban Marked'(UM), 'Urban Marked Multiple lanes'(UMM) and 'Urban Unmarked'(UU).

3 Model selection

The cross entropy of neural networks with different number of hidden units was calculated and it was found that the model with 110 hidden units generalized well. The plot is shown below.



Deep neural networks were also considered as alternative models. A network with three layers of hidden units with each layer having 35 units was considered. This was chosen as the best one in terms of learnability when compared with three layered networks of one with 30 units and another with 45 units in each layer. It was found that the deep network was not better than the single layer network with 110 units. While the accuracy for the validation set of UMM dataset with the single layer network was found to be 0.041816, the same value with the deep network came out to be 0.053548.

Convolutional neural networks (CNN) were not considered because the super pixels (from SLICO) did not have constant number of pixels in them. While the CNN filters will need a constant number of pixels.

So, with all these results a single layer neural network with 110 hidden units was fixed to be the model.

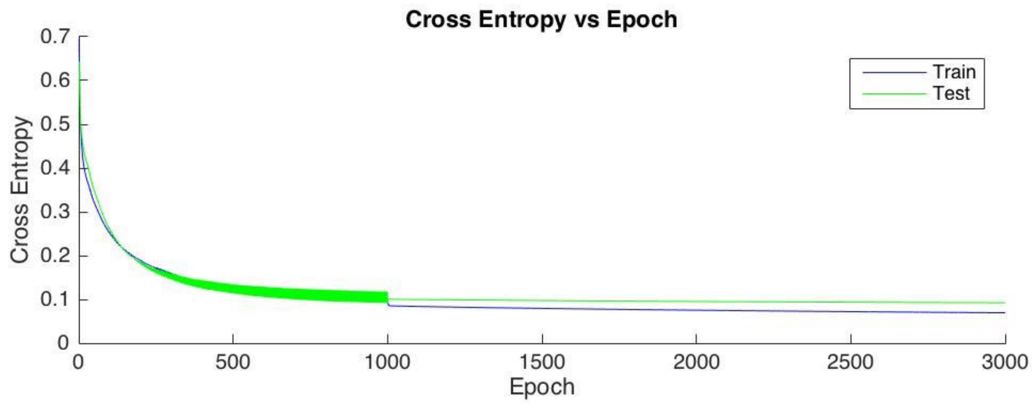
4 Results and observations

The plot of the evolution of cross entropy error for UM dataset with the assumed neural network is given in Figure 3. The plots for UMM, UU are similar.

And the achieved F1 measures for the road area of the training, validation and test sets are given in the following table:

Set	UM	UMM	UU
Training	0.928062	0.922458	0.905786
Validation	0.896658	0.91836	0.85452
Test	0.8822	0.9230	0.8484

Figure 3: Evolution of Cross Entropy error for UM dataset



As we can see from the table that 'UMM' roads can be estimated more accurately than either 'UU' or 'UM'. This is understandable considering that the 'UMM' roads tend to be out in the open with well defined edges separating textures and have fewer shadows because of being wider. Lack of these same characteristics can work against 'UU' roads which can be seen to fare the worst.

To show how the algorithm fares with a single example, an image 'umm_000047' was selected from the dataset. We will see the original image in Figure 4, the super pixels with ground truth (generated using the *gt_image_2* images) in Figure 5 and the predictions (from the artificial neural network) for the super pixels in Figure 6.

Figure 4: Original image

As can be seen from the figures 5 and 6, the algorithm does a good job in estimating the road area in 'UMM' figures.

Figure 5: Ground Truth labelled super pixels

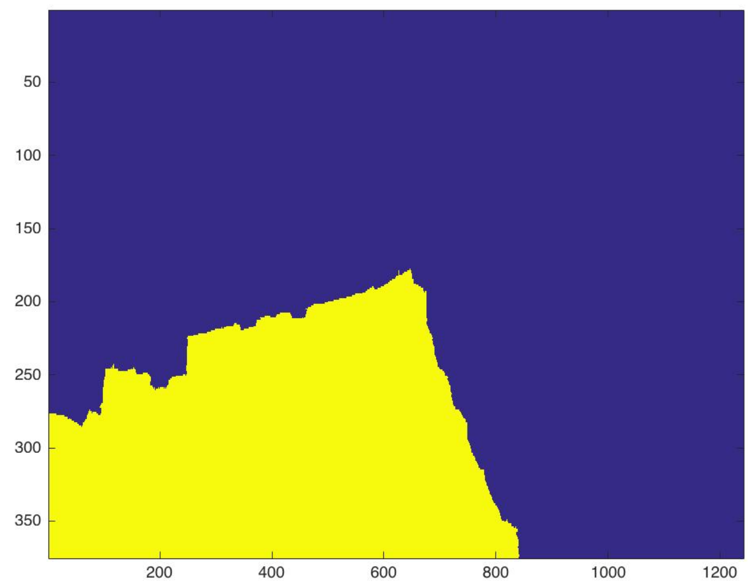
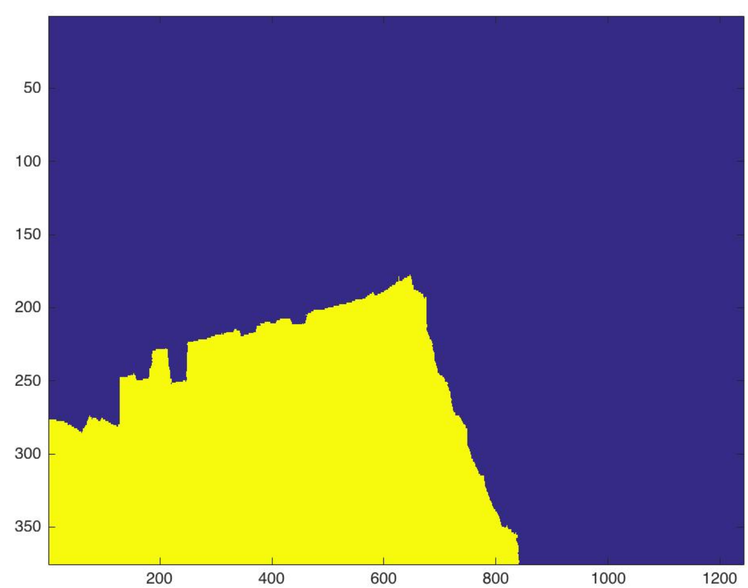


Figure 6: Super pixels with predictions



5 Conclusion

In summary I have seen that feature selection and construction are important tools in improving the learning process. I have also seen that these features can be constructed based on the cues that humans generally use to differentiate between things. I have also learnt the importance of reducing the noise in the data, and that shadows in an image can be considered as noise. Moving to the process of training, I have observed that deep networks require higher learning rate and momentum in the beginning when compared to single layer networks. Looking at the data set, I can say that UMM roads were easiest to generalize and I have understood the reasons for it. That lead to understanding the reasons for why the UU roads are the hardest ones to generalize.

If this process were to be applied on pixels rather than on super pixels, I would like to use stochastic gradient descent in order to speed up the process. And with pixels I would also be facing fewer features too (no histograms). That would also help me a little in scaling.

References

[1] Sun, Zehang, George Bebis, and Ronald Miller. "On-road vehicle detection using evolutionary Gabor filter optimization." Intelligent Transportation Systems, IEEE Transactions on 6.2 (2005): 125-137.

[2] Broggi, Alberto, et al. "A new approach to urban pedestrian detection for automatic braking." Intelligent Transportation Systems, IEEE Transactions on 10.4 (2009): 594-605.

[3] Geiger, Andreas, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite." Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.

[4] Achanta, Radhakrishna, et al. "SLIC superpixels compared to state-of-the-art superpixel methods." Pattern Analysis and Machine Intelligence, IEEE Transactions on 34.11 (2012): 2274-2282.

[5] Tan, Ceryen, et al. "Color model-based real-time learning for road following." Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE. IEEE, 2006.

[6] lvarez, Jos M., and Antonio M. Lopez. "Road detection based on illuminant invariance." Intelligent Transportation Systems, IEEE Transactions on 12.1 (2011): 184-193.