Road Detection using Convolutional Neural Networks

Aparajit Narayan[1], Elio Tuci[2], Frédéric Labrosse[1] and Muhanad H. Mohammed Alkilabi[1, 3]
[1]Aberystwyth University, [2]Middlesex University London, [3]University of Karbala
[1]apn3,ffl,mhm1@aber.ac.uk, [2]E.Tuci@mdx.ac.uk

Abstract

The work presented in this paper aims to address the problem of autonomous driving (especially along ill-defined roads) by using convolutional neural networks to predict the position and width of roads from camera input images. The networks are trained with supervised learning (i.e., back-propagation) using a dataset of annotated road images. We train two different network architectures for images corresponding to six colour models. They are tested "off-line" on a road detection task using image sequences not used in training. To benchmark our approach, we compare the performance of our networks with that of a different image processing method that relies on differences in colour distribution between the road and non-road areas of the camera input. Finally, we use a trained convolutional network to successfully navigate a Pioneer 3-AT robot on 5 distinct test paths. Results show that the network can safely guide the robot in this navigation task and that it is robust enough to deal with circumstances much different from those encountered during training.

Introduction

To achieve automatic driving for cars and robots, it is essential to differentiate between road and non-road based on camera inputs. While standard computer vision techniques can achieve this for roads that have clear lines demarcating the edges, the problem becomes more complex where the road surface is poorly delineated. The image processing algorithm used for this task is expected to be robust enough to detect roads across a variety of complex dynamic operational conditions. If road parameters such as position of the road area with respect to the vehicle headings, width and shape of the road can be successfully extracted by processing camera images, the decision of the vehicle on where to move during navigation becomes relatively simple to make.

A number works such as (Álvarez and López, 2011) and (S. Thrun et al., 2006) attempt to solve the road-detection problem by segmenting the color distribution of the input image into road and non-road areas. This approach is particularly effective since it does not depend on features such as road markings and is flexible enough to be applied on a much wider range of road environments. However the main issue with such explicit colour-based segmentation techniques is that the colour distributions along a road is not always static. Local and dynamic changes such as shadows, puddles and changing textures can reduce the accuracy of these methods. Another factor leading to incorrect segmentation is that some channels can have almost equivalent distributions in road and non-road parts of the image. Indeed, the number and nature of the colour channels considered can have a huge impact on the methods' performance. Based on these factors the authors in (Ososinski and Labrosse, 2015) describe a non-segmentation based method, which achieves low detection errors across a number of road-environments. The detection algorithm is also used to autonomously drive a rover along long stretches of poorly defined roads. This method, referred to as ASC (Adaptive Statistical Colour-Based) in subsequent sections, works by projecting a trapezoidal shape onto the camera image. The trapezoid delimits the road area onto the image plane. The main criteria for road detection is the Mahalanobis distance between pixels within and outside this trapezoidal shape. The authors have evaluated this method across a variety of colour models and on a large dataset of unwrapped panoramic images generated by a camera mounted on vehicles navigating multiple road environments. Despite low error rates achieved by this colour-based method for the vast majority of the frames, there were a number of failure cases and sequences of systematic detection offsets. Moreover, as the road model maintained by the method relies on colour characteristics of the previous frame, any sudden/drastic changes could potentially result in detection errors. We thus recognize a need to implement an alternative approach to compare the results obtained by the ASC method and explore the possibility for further improvements.

Related Work

Machine learning techniques and artificial neural networks have been proposed as a potentially effective solution to the problem of road detection in noisy and highly variable real world conditions. In the work presented in Zhou et al.

This work has been partly sponsored by Fujitsu and HPC-Wales.

Carole Knibbe et al, eds., Proceedings of the ECAL 2017, Lyon, France, 4-8 September 2017,
(Cambridge, MA: The MIT Press, ©2017 Massachusetts Institute of Technology).

(2010), a support vector machine segments the input image by classifying pixels into either road or non-road classes. Before starting navigation, pixels form part of the road are selected to form the SVM's training set. To make the system adapt to changing properties of the road and recover from initial misclassification, the training set is constantly updated by adding pixels from new frames and discarding pixels from previous ones (based on assumptions about the structure of the road). However this still doesn't prevent against classification errors in complex environments and dynamic weather conditions when the majority of pixels in the SVM training set cease to be representative of the road-surface.

The ALVINN project (Pomerleau, 1992) was one of the earliest examples of using neural networks to control an autonomous vehicle driving on real outdoor roads. A 3 layered feed-forward neural network taking in a grey-scale image as its input was used to output the vehicle's steering commands. The network's training data was initially generated by a human navigating a road-simulator (based on real images) developed in-house by the authors. Later versions used "on-the-fly" images and control commands captured from a human controller driving on outdoor roads to train the network. While the network could control the vehicle in stretches of roads similar to those it was trained with, it was severely limited in generalizing to new road environments. To overcome this, the same authors proposed a new modular architecture called MANIAC (Jochem et al., 1993). MANIAC consisted of several individual networks trained under different types of roads integrated into a modular superstructure, wherein activation from all the modules are combined to form the final output vector. Whilst being capable of driving in multiple road-environments without needing to switch controllers, the system was still not adequate in the sense that accurately representing the entire range of possible road-types would require a progressively larger number of individually trained modules, and each time a new module was added to the system, the entire structure would require retraining.

In (Shinzato and Wolf, 2011) another solution using multiple neural networks was proposed, to make the system applicable to a wider variety of roads. The networks were trained on images which were divided into blocks of 10x10 pixels. Each block was annotated as being either navigable (part of the road) or not-navigable (part of the non-road), and features extracted from a block formed the input to the network. The responses from multiple networks (different networks being fed a different set of features) were combined to predict the final classification results of blocks in new frames. The limitations of these and other methods such as (Tudoran and Neagoe, 2010) that use neural networks trained for specific road types is that while they can adapt to further sections of already trained roads, they are often poor at adapting to previously unseen environments.

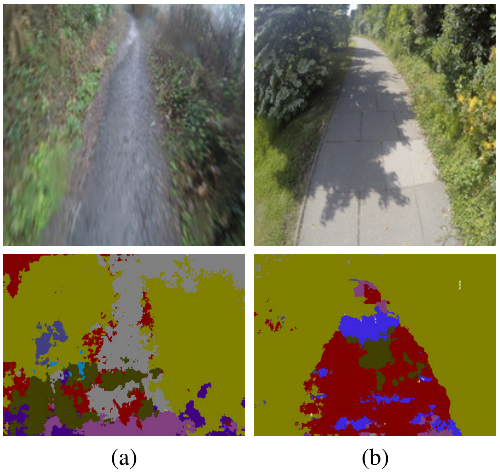

Figure 1: The top row shows raw images of two road environments. The bottom row shows the corresponding pixel wise classification of these road images using the deep convolutional neural network described in (Badrinarayanan et al., 2015). Purple pixels (almost completely absent in these images) refer to parts of the image which the network identifies as the road.

With the growing usage of deeper neural architectures (LeCun et al., 2015), researchers have been trying to apply deep convolutional neural networks to various instances of the problem of detection and navigation on a road by autonomous vehicles. Compared with more conventional design based computer vision techniques, deep learning models such as convolutional neural networks offer significantly higher accuracy for image recognition tasks such as MNIST (Lecun et al., 1998) and ImageNet (Krizhevsky et al., 2012). Convolutional networks are powerful due to their ability to transform small local image patches into higher level feature representations. Thus having such a model that carries information of universal features that are common to any road-environment could result in more robust and general-purpose detection. Most of the work using convolutional networks for automatic driving has been focused and evaluated on highways and urban roads. An exception to this is the work described in (Hadsell et al., 2009), wherein a controller which was able to autonomously drive on highly delineated off-road terrains was developed. A convolutional network (trained offline) was used as a feature extractor providing a robust representation of complex environments. Terrain in front of the robot was segmented into multiple categories by a classifier, which was trained online using self-supervised data labels. In (Chen et al., 2015) and (Liu and Wang, 2016), a deep convolutional neural network was trained using hours of footage taken from a human driving a car in a racing simulator. Both implementa-

Carole Knibbe et al, eds., Proceedings of the ECAL 2017, Lyon, France, 4-8 September 2017,
(Cambridge, MA: The MIT Press, ©2017 Massachusetts Institute of Technology).

tions outperformed a baseline method where road detection was carried out using pre-defined features based on Gabor filters (see (Oliva and Torralba, 2001)). While the neuro-controllers were successfully tested on real world video sequences, these were limited to a subset of roads found in urban environments. Similarly a convolutional network for lane detection on highway roads was trained in (Huval et al., 2015) using a large dataset of video sequences collected over the course of multiple days. Using this approach over more traditional and simplistic 'line detector' methods meant that lanes could be accurately estimated even when there were occlusions and environmental disturbances. In (Badrinarayanan et al., 2015), a de-convolutional (convolutional encoder-decoder) network was used for semantic pixel-wise image labeling. The results show that the network is capable of successfully parsing and segmenting urban road scenes into multiple categories such as roads, pavements, trees, etc. The results also show that the network performs significantly better than various other machine learning methods. The method described in (Alvarez et al., 2012) also relied on scene segmentation by a convolutional network detecting the road. Classification from the network is aggregated with that from a statistical color-information based method. With this approach the generalization capabilities of the convolutional network which had learned high-level features from other road scenes could be combined with the color-based method which could adapt to dynamic changes in the current road. One of the key aspects of this work was that the convolutional network was trained in a semi-supervised manner on road images using noisy labels generated by classifiers trained on a larger general image data set. To the best of our knowledge, convolutional neural networks such as those described in (Badrinarayanan et al., 2015), (Chen et al., 2015) and (Alvarez et al., 2012) have not been evaluated on non-urban roads, and their performance in such environments is yet to be ascertained. We have tested the system described in (Badrinarayanan et al., 2015) on various images of poorly delineated roads, some of which taken from the environments where we conducted the experiments described in this study. The performance of the system turned out to be relatively poor (see Figure 1).

Methods

For the work presented in this paper we train and evaluate convolutional neural networks (CNN) with two different architectures on 10 sets of road image sequences representative of a variety of environments (urban and rural). We train the CNNs for 6 different colour models to explore whether differences in colour representation influence the networks' performance. A detailed description of the colour models we use (*RGB*, *HSV*, *YUV*, *YCbCr*, *lab*, *CbCra*) can be found in (Ososinski and Labrosse, 2015). We also evaluate the performance of the method described in (Ososinski and Labrosse, 2015) (ASC) for these datasets, providing a bench-mark for how well the networks perform. The results of our study show that the best performing networks match the ASC method's detection accuracy for majority of the datasets and even outperform it for two environments.

Road Shape

Drawing inspiration from the method described in [2], we use a trapezoidal model with two changeable parameters to provide a best-fit to the road in the input image. The convolutional neural network generates two parameters, the position (x) and the width (w) of the trapezoid delimiting the road area. We acknowledge that varying other parameters such as θ and h (see Fig. 2) could have provided a closer fit between the trapezoid and the road. However, the addition of further parameters inevitably increases the complexity of task for the network, and it clearly generates overheads to the process of annotating the images for the network training. Indeed the position x on its own provides enough information for a robot to stay on the road. However, the width (w) parameter is required if the speed of the robot needs to be varied and is useful for more complex control procedures resulting in smoother, less oscillatory motion.

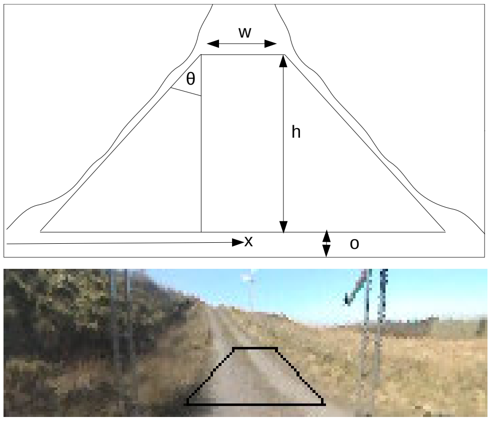

Figure 2: Figure showing the parameters of the trapezoidal road model and the projection of the model on an image of a road.

Convolutional Neural Network (CNN)

Besides exploring colour representation, we also investigate the effect of neural model complexity on the road-detection accuracy. Do deeper and more complex neural network architectures give us better detection results? For this purpose we train and evaluate two different sized convolutional neural network architectures, as well as a simple 4-Layered feed-forward network. The feed-forward network receives a 50×50 3 channel image which is flattened to form an input vector of 7500 input neurons. It has 2 hidden layers with 1000 and 600 neurons respectively and two nodes coding for the parameters x and w, which define the position and width

Carole Knibbe et al, eds., Proceedings of the ECAL 2017, Lyon, France, 4-8 September 2017,
(Cambridge, MA: The MIT Press, ©2017 Massachusetts Institute of Technology).

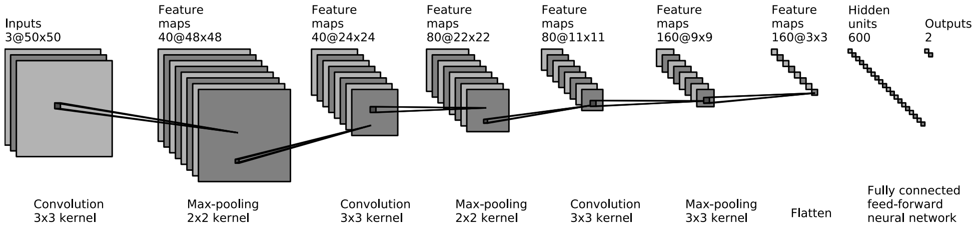

Figure 3: Figure showing the architecture of the LCNN. Sizes of the associated convolution and max-pooling kernels are annotated at the bottom of each relevant layer.

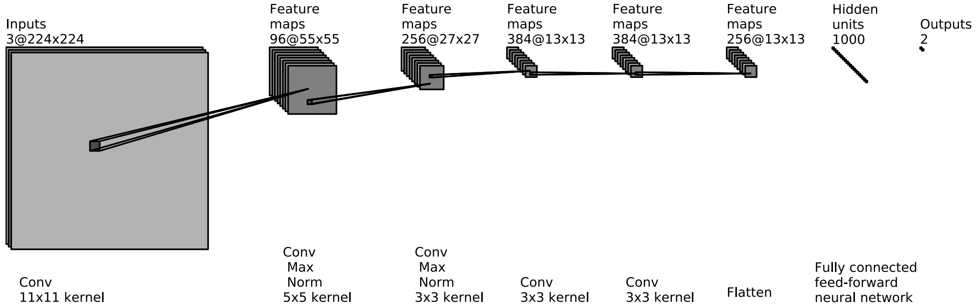

Figure 4: Figure showing the architecture of the modified AlexNet. Conv, Max, Norm refer to convolution, max-pooling and local contrast normalization operations respectively (see (Krizhevsky et al., 2012)). The sizes of the associated convolution kernels for each layer are annotated at the bottom of the image.

respectively of the road model (see Section). The values of *x* and *w* are then mapped to the range of 0-180 pixels. The results of this network were very poor with average position errors greater than 20 pixels across all datasets. These results are not discussed in this paper, but established the fact that a more complex model than a regular feed-forward neural network is required for this task.

$$y_j = ReLU(\sum k_{ij} * x_i) \tag{1}$$

$$ReLU(x) = max(0, x) \tag{2}$$

The architecture of the smaller convolutional network (which we shall refer to as the Light Convolutional Neural Network or LCNN) can be seen in figure 3. Equations 1 and 2 describe the multi-channel convolution operation which takes place at each convolution layer (Jarrett et al., 2009). For the j^{th} filter in a layer y_j is the output corresponding to a particular input patch. x_i is the i^{th} channel of the input and k_{ij} is the corresponding convolution kernel. The network follows the conventional expanding shape approach used for most architectures by having 40, 80 and 160 filters in its first, second and third convolution layers. It receives a 50×50 image as its input. Depending on which colour model is being used the image is split up into its constituent channels and these are individually fed into the filters for the first layer. The network activations are then propagated to the output layer, made of two output nodes predicting the values of *x* and *w*. Weights are initialized randomly and updated using a variant of standard mini-batch gradient descent called rmsprop (Dauphin et al., 2015). We use a learning rate (η) of 0.001 and to prevent over-fitting of these complex models, dropout noise (see (Srivastava et al., 2014)) of 0.2 and 0.5 is used at the convolution and fully connected layers respectively. Theano (see (Theano Development Team, 2016)) running on the HPC-Wales GPU cluster is used to train and evaluate multiple networks (corresponding to different colour models) in parallel for a maximum of 100 iterations. Besides this we also train a slightly modified version of the AlexNet architecture described in (Krizhevsky et al., 2012). As shown in figure 4 this is a much larger and deeper network with as many as 5 convolution layers, compared to the LCNN architecture described above. Similar to the LCNN it has 2 output nodes predicting the values of *x* and *w* and receives a 3 channel image as its input. Refer to (Krizhevsky et al., 2012) for a more detailed overview of this architecture. The original unmodified architecture has 2 fully connected layers (with 4096 neurons each) after the fifth convolution layer. We take a network with this original architecture trained on the ImageNet dataset and remove the fully connected and output layers. A randomly initialized hidden and 2 node output layer are then introduced. The entire model is then re-trained end to end for 30 iterations

Carole Knibbe et al, eds., Proceedings of the ECAL 2017, Lyon, France, 4-8 September 2017,
(Cambridge, MA: The MIT Press, ©2017 Massachusetts Institute of Technology).

Table 1: Median and Standard Deviation of position error (in pixels) of best LCNN for each colour model across all datasets. Negative values indicate the predicted position of the road-shape being to the left of that in the ground truth.

Dataset	Track		Llan		Farm		Rugged		Rain		Footpath		Lakeside		Steep		K59		K56	
	med	std	med	std	med	std	med	std	med	std	med	std	med	std	med	std	med	std	med	std
RGB	-2.1	4.4	20.4	11.6	-1.8	10.6	-13.8	10.5	-7.2	11.6	-7.8	10.7	-2.3	9.0	-13.0	6.3	-6.3	6.0	5.9	6.5
HSV	4.4	7.6	11.0	8.9	5.1	9.0	4.2	7.3	-4.0	7.4	3.3	9.4	13.2	5.0	**1.9**	**6.0**	4.3	3.4	**0.5**	**3.9**
YUV	-3.9	1.6	**-1.9**	**5.6**	**-1.1**	**7.8**	-5.2	9.1	**-2.1**	**12.3**	-9.1	7.6	-5.4	3.5	-4.5	5.9	-0.9	3.1	6.7	42
YCbCr	-5.9	1.7	-3.2	5.9	-3.6	7.7	-5.9	9.0	-3.9	12.4	12.2	7.7	-8.8	3.3	-5.1	6.0	-1.9	2.3	3.5	4.4
lab	-1.3	1.6	2.2	5.2	-1.9	7.2	**-2.2**	**9.1**	-3.7	10.4	-2.1	8.2	**-1.6**	**2.3**	-6.4	5.4	-1.9	1.1	-3.2	1.0
CbCra	**0.03**	**1.2**	-2.8	5.4	-5.3	6.9	-3.1	8.9	-5.7	10.9	**-2.1**	**7.4**	-5.8	1.9	-5.8	5.7	**-0.6**	**1.4**	-2.8	1.8

Table 2: Median and Standard Deviation of position error (in pixels) of best modified AlexNet for each colour model across all datasets. Negative values indicate the predicted position of the road-shape being to the left of to that in the ground truth.

Dataset	Track		Llan		Farm		Rugged		Rain		Footpath		Lakeside		Steep		K59		K56	
	med	std	med	std	med	std	med	std	med	std	med	std	med	std	med	std	med	std	med	std
RGB	-0.7	2.3	2.6	5.3	-3.1	6.9	-3.7	8.9	-4.1	10.6	-3.1	8.3	-2.5	1.8	-5.8	5.5	11.6	2.6	5.4	2.8
HSV	4.0	2.4	9.7	5.4	-2.1	6.7	4.0	8.4	**-1.0**	**9.5**	3.5	8.3	**0.2**	**1.9**	**1.7**	**6.1**	1.9	3.2	0.5	3.8
YUV	2.3	1.5	1.1	5.3	**-1.8**	**6.6**	-2.4	8.8	-4.6	10.4	0.09	7.4	-2.6	1.7	-5.0	5.4	1.3	1.5	1.2	1.5
YCbCr	1.1	1.4	1.8	5.3	-2.0	6.6	-2.1	9.0	-3.9	10.5	-0.6	7.5	-2.0	1.7	-4.4	5.6	8.2	18.3	-2.1	27.0
lab	**1.2**	**1.2**	**-0.02**	**5.2**	-5.6	6.6	-2.0	8.9	-4.7	10.5	-0.02	7.5	-4.3	1.7	-4.6	5.5	1.0	1.0	**-0.6**	**1.7**
CbCra	2.1	1.2	1.5	5.3	-3.7	6.7	**-0.5**	**9.0**	-3.1	10.7	**-0.1**	**7.6**	-3.2	1.7	-3.7	5.6	**0.5**	**1.1**	-0.8	1.3

(lesser training time because of pre-initialized weights) with GPU acceleration in Caffe ((Jia et al., 2014)), using the step learning policy (base_lr = 0.0001).

Datasets

As mentioned before we want to train our networks to be able to detect roads in any environment irrespective of colour, lighting, geometry etc. We also want our network to be flexible enough to work for images captured on any platform and camera configuration. For this reason 10 datasets (20880 frames) corresponding to road-sequences in varied en- vironments and captured from 4 different sources (camera, platform configurations) are used to train and evaluate our network. Each image in these datasets is manually annotated with the position and width parameters of the road- model (see figure 2). Select images along with descriptions of each dataset are available in the supplementary document (at `https://www.aber.ac.uk/en/cs/research/ir/dss/#road-driving`).

Offline Detection Results

After training we select the best network from each colour model of each architecture for off-line evaluation. For this paper we select networks solely based on their accuracy in predicting the position (x) parameter, as position (x) infor mation is more important for navigation than width (w). We look for networks that can perform to a reliable degree of accuracy across all datasets rather than networks which minimize error for a few datasets but fail in the rest. For each colour model we set this average error threshold at 9 pixels per dataset. If no networks are found matching this criteria for one for more datasets we increase this threshold by 1 pixel each for the corresponding datasets and search again. If multiple networks are found within the threshold, we select the one with lowest overall position error. The median position errors of the best networks for each colour model with the LCNN and AlexNet architectures is presented in tables 1 and 2 respectively. Cells with bold text highlight the lowest error across all colour spaces for each dataset. Results for the width (w) parameter prediction of these selected networks as well as videos which provide a better understanding of the detection behaviour across colour models and architectures are presented in the supplementary document[1].

From these tables, we can observe that detection accuracy varies according to the colour model being used. This can be attributed to manner in which colour information is represented in these models. For example, *RGB* doesnt distinguish between luminance and chrominance, while other models have a dedicated channel for brightness (except for *CbCra* which is a hybrid model with only chrominance information). Indeed *YUV* and *YCbCr* which are the most similar among the colour models, show a much lower degree of disparity in performance. This variability across colour models is however somewhat less pronounced for AlexNets position prediction (table 2) where results are more uniform (including *RGB*). For position errors across all colour models and datasets it can be said that the AlexNet architecture with its deeper structure provides a higher level of accuracy on the whole, compared to LCNN. However choosing an appropriate colour model with the LCNN (such as lab orCbCra) can give position errors comparable to those achieved with AlexNet (see figure 6).

We compare the two best performing colour models for the ASC method (*HSV* and *lab*) with the corresponding LCNN and AlexNet networks (see figure 6 and 5). For both

[1] At `https://www.aber.ac.uk/en/cs/research/ir/dss/#road-driving`

Carole Knibbe et al, eds., Proceedings of the ECAL 2017, Lyon, France, 4-8 September 2017,
(Cambridge, MA: The MIT Press, ©2017 Massachusetts Institute of Technology).

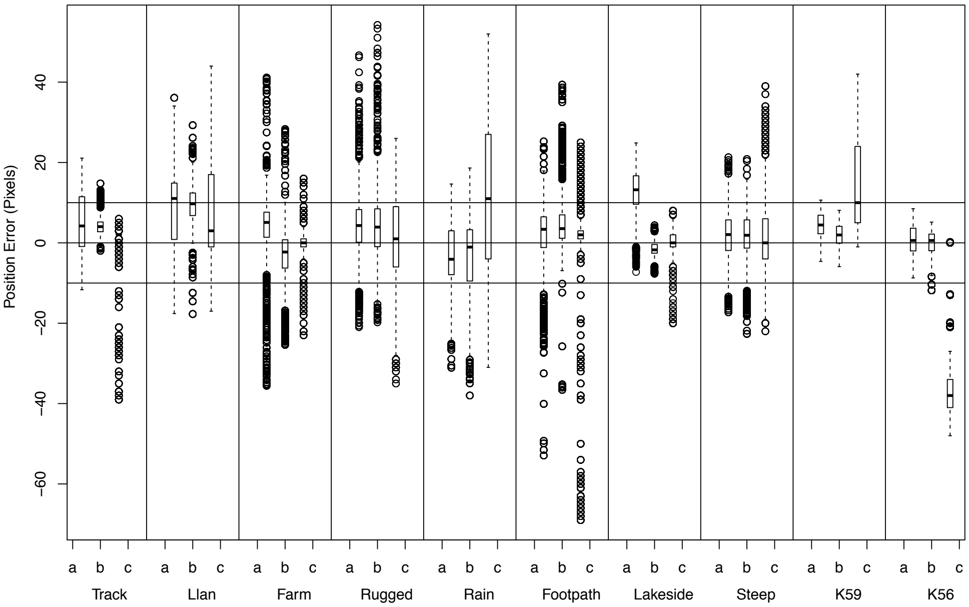

Figure 5: Boxplots comparing the position (*x*) accuracy of the two convolutional network architectures (LCNN and AlexNet) and the adaptive statistical colour-based (ASC) method for the HSV colour model across all datasets. Plots for LCNN, AlexNet and ASC for each dataset correspond to (a), (b) and (c) respectively. Horizontal lines are drawn at the -10, 0 and 10 pixel error marks as visual aids.

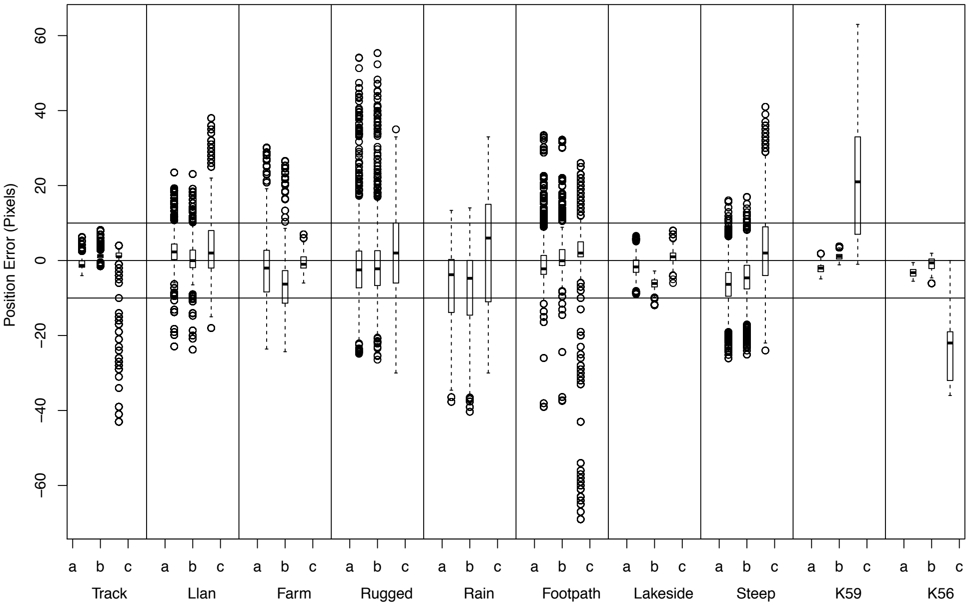

Figure 6: Boxplots comparing the position (*x*) accuracy of the two convolutional network architectures (LCNN and AlexNet) and the adaptive statistical colour-based (ASC) method for the lab colour model across all datasets. Plots for LCNN, AlexNet and ASC for each dataset correspond to (a), (b) and (c) respectively. Horizontal lines are drawn at the -10, 0 and 10 pixel error marks as visual aids.

colour models the AlexNet's performance can be said to be comparable or better than the ASC across all datasets. The same could be said for LCNN too baring the Lakeside dataset in which the network's detection exhibit an offset to the right for a sequence of frames. The ASC method performed poorly in the two KITTI datasets which contained urban roads for both colour-models (see figure 7). The errors exhibited for both datasets were detrimental for navigation. In K59, the detected shape drifted off towards the right as the image-sequence progressed to include the pavement and not the road. In K56 the trapezoidal model was required to stay within the white lane demarcations (see figure 7). Instead for both colour-models (more so for *lab*) the ASC's detected shape gradually shifted towards the adjoining lane on the left.

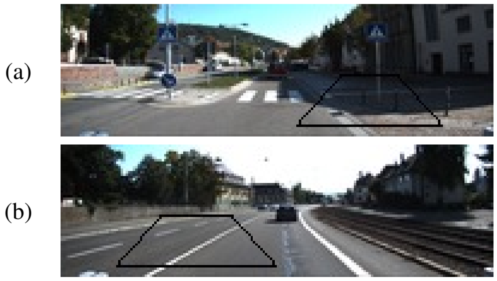

Figure 7: Images showing detection failure of the ASC method. (a) corresponds to a frame in the K59 dataset for the *HSV* colour model, (b) corresponds to a frame in the K56 dataset for the *lab* colour model.

Robot Trials

To ascertain the fact that road-detection outputs from convolutional networks can be used to control an autonomous vehicle, we ported one of our networks with the LCNN architecture (operating in the *HSV* colour model) onto a Pioneer 3-AT robot. The robot's motion is controlled by a simple design rule which changes its bearing based on the changes in position values generated by the network. It should be noted due to hardware limitations (absence of a GPU) on the Pioneer 3-AT each update cycle took ≈ 3 seconds. It was for this reason we did not test the larger AlexNet which takes even longer to complete an update-cycle without GPU acceleration. We conducted a total of 25 trials across 5 different road environments that we created in an indoor laboratory. While not being representative of roads in the real-world these environments contain a lot of dynamic variations with respect to colour and geometry and they are significantly distinct from each other. More importantly, they were completely different to the type of roads that network had encountered as part of its training set. In addition the camera configuration used in these trials is also different to those used in training. All paths required the robot to make at least one turn to stay on-course. Since the purpose of these experiments is to test the network's ability to maintain track of unseen roads while on a highly noisy platform, we do not terminate a trial when the robot goes partly outside the boundaries. The trial is allowed to continue if the robot can correct its course and come back inside the road within 2 update cycles. If however the robot completely loses track of the road and travels off-course the trial is terminated.

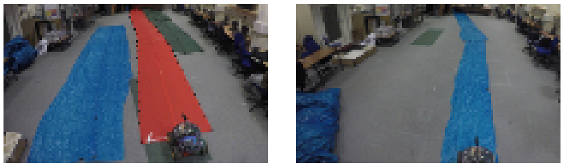

Figure 8: Figure showing environments Red Blue (a) and Blue Floor (b) for the robot trials. The path in image (a) has been manually annotated by black markers to show that the tarpaulin and netting on either side are not considered to be part of the road.

Table 3: Summary of robot trial results across 5 environments. Divergence measures the robot's deviation from the center of the road.

Environment	Width	Divergence (cm)		Success
	(cm)	mean	sd	num
Red-Floor	101 ± 15	18.79	17.12	4/1
Red-Green	101 ± 15	23.10	18.34	5/0
Blue-Floor	78 ± 18	12.29	10.89	5/0
Green-Blue	83 ± 7	17	14.74	5/0
Red-Blue	101 ± 15	37.74	37.80	4/1

The robot was successful in reaching the end of the road in 23 out the 25 trials (see table 3). As indicated by the divergence values in table 3, apart from environment Blue-Floor the robot's motion was at times quite oscillatory. It drifted towards either edge of the road at certain points and then had to make sharp turns to stay on course. It especially struggled to stay in the middle for environment Red-Blue where after navigating the first half of the road it kept turning towards the green netting (see Fig. 8) on the right. However the fact that the robot despite its rudimentary control system was able to execute turns and make constant adjustments is testament to the robustness of the network. Only a small minority of training images accounted for the sharp changes in detection the network had to repeatedly make to correct its course during these trials. From our observations of these trials we feel it is important to devise a more complex control strategy that is compatible with the detection values returned by the network. It would also be worthwhile integrating the width information for future trials to ensure better motion control.

Carole Knibbe et al, eds., Proceedings of the ECAL 2017, Lyon, France, 4-8 September 2017,
(Cambridge, MA: The MIT Press, ©2017 Massachusetts Institute of Technology).

CONCLUSION

For view-invariant problems, such as detecting roads from a static camera configuration, deep convolutional networks with their stacked internal representation of the input space share many similarities with biological vision systems (Kriegeskorte, 2015). Convolutional networks may not be easy to analyse and decompose into clear operational principles. However, contrary to other design based methods, they are free from those human injected biases that sometimes limit the robustness and the capability of autonomous systems to operate in a-priori unknown conditions. We have shown that when compared with the colour based method (ASC) used as a benchmark, the deep-convolutional neural network (AlexNet) was able to perform equally well if not better for all the datasets in the off-line detection tests. Moreover with the right colour representation, a shallower architecture (LCNN) can also achieve similar levels of detection accuracy. Our experiments in section show that such a network's road shape predictions can be used to successfully navigate an autonomous vehicle/robot in environments very different from what it was trained for. Future work will include further evaluation across a wider range of datasets and actual outdoor roads to conclusively determine if convolutional neural networks consistently offer a significant performance advantage. We are in the process of integrating an embedded GPU module onto our mobile robot platform to enable us to run deep convolutional network models in real-time with a more complex control strategy.

References

Álvarez, J. and López, A. (2011). Road detection based on illuminant invariance. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):184–193.

Alvarez, J. M., Gevers, T., LeCun, Y., and Lopez, A. M. (2012). *Road Scene Segmentation from a Single Image*, pages 376–389. Springer Berlin Heidelberg, Berlin, Heidelberg.

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2015). SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*.

Chen, C., Seff, A., Kornhauser, A., and Xiao, J. (2015). DeepDriving: Learning affordance for direct perception in autonomous driving. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2722–2730.

Dauphin, Y., de Vries, H., and Bengio, Y. (2015). Equilibrated adaptive learning rates for non-convex optimization. In *Advances in Neural Information Processing Systems*, pages 1504–1512.

Hadsell, R., Sermanet, P., Ben, J., Erkan, A., Scoffier, M., Kavukcuoglu, K., Muller, U., and LeCun, Y. (2009). Learning long-range vision for autonomous off-road driving. *Journal of Field Robotics*, 26(2):120–144.

Huval, B., Wang, T., Tandon, S., Kiske, J., Song, W., Pazhayampallil, J., Andriluka, M., Rajpurkar, P., Migimatsu, T., Cheng-Yue, R., et al. (2015). An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*.

Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y. (2009). What is the best multi-stage architecture for object recognition? In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2146–2153.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

Jochem, T. M., Pomerleau, D. A., and Thorpe, C. E. (1993). MANIAC: A next generation neurally based autonomous road follower. In *Proceedings of the International Conference on Intelligent Autonomous Systems*.

Kriegeskorte, N. (2015). Deep neural networks: A new framework for modeling biological vision and brain information processing. *Annual Review of Vision Science*, 1(1):417–446.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Liu, Z. and Wang, Y. (2016). Deeper direct perception in autonomous driving. *Technical report*.

Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175.

Ososinski, M. and Labrosse, F. (2015). Automatic driving on ill-defined roads: An adaptive, shape-constrained, color-based method. *Journal of Field Robotics*, 32(4):504–533.

Pomerleau, D. (1992). Progress in neural network-based vision for autonomous robot driving. In *Proceedings of the Intelligent Vehicles '92 Symposium*, pages 391–396.

S. Thrun et al. (2006). Stanley: The robot that won the DARPA grand challenge. *Journal of Field Robotics*, 23(9):661–692.

Shinzato, P. Y. and Wolf, D. F. (2011). A road following approach using artificial neural networks combinations. *Journal of Intelligent & Robotic Systems*, 62(3):527–546.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Theano Development Team (2016). Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688.

Tudoran, C.-T. and Neagoe, V.-E. (2010). A new neural network approach for visual autonomous road following. *Latest Trends on Computers*, 1:266–271.

Zhou, S., Gong, J., Xiong, G., Chen, H., and Iagnemma, K. (2010). Road detection using support vector machine based on online learning and evaluation. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 256–261. IEEE.