Learning to Drive: End-to-End Off-Road Path Prediction

Christopher J. Holder and Toby P. Breckon
Department of Computer Science, Durham University, United Kingdom

xxxxxx

Abstract—Autonomous driving is a field currently gaining a lot of attention, and recently 'end to end' approaches, whereby a machine learning algorithm learns to drive by emulating a human driver, have demonstrated significant potential. However, recent work has focused on the on-road environment, rather than the more challenging off-road environment. In this work we propose a new approach to this problem, whereby instead of learning to predict immediate driver control inputs, we train a deep convolutional neural network (CNN) to predict the future path that a vehicle will take through an off-road environment visually, addressing several limitations inherent in existing methods. We combine a novel approach to automatic training data creation, making use of stereoscopic visual odometry, with a state-of-the-art CNN architecture to map a predicted route directly onto image pixels, and demonstrate the effectiveness of our approach using our own off-road data set.

1. Introduction

A huge body of research has been conducted in the field of autonomous driving, from both academia and the automotive industry, with much notable work in the areas of scene understanding [6] and road detection [10]. However, only a very limited body of work covers the more challenging problem of off-road autonomous driving [8], [17]. In the off-road environment, path detection can be much more difficult than on-road, due to uneven terrain, hidden obstacles and an overall lack of structure, however there are many real-world applications for such technology.

Convolutional Neural Networks (CNN) have demonstrated unprecedented results at a multitude of image classification tasks [11], revolutionizing the field of computer vision in recent years. Loosely based on the biological brain, CNNs offer a 'black box' approach to machine learning, where the designer is aware of input and output data, but not necessarily of how that data is processed intermediately. This means that CNNs are particularly suited to tasks that humans can perform intuitively without relying on a structured set of rules, such as planning a safe route through an off-road environment.

This idea underpins the concept of end-to-end autonomous driving, first proposed by Pomerleau in 1989 [14] with the Autonomous Land Vehicle in a Neural Network (ALVINN), which uses a neural network

Digital Object Identifier 10.1109/MITS.2019.2898970

comprising a single fully-connected layer, taking a grayscale image and laser rangefinder data as input, trained to predict the steering wheel inputs made by a human driver. In 2004, the DARPA Autonomous Vehicle (DAVE) project [12] trained a more complex, six-layered network to drive a radio-control car in off-road environments, using data collected over several hours of human driving. More recent advances in deep-learning have led to the approach proposed in [2], which uses a network of 5 convolutional layers and 3 fully-connected layers, trained with 72 hours of human driving data, to successfully follow lanes on public roads.

The work in [18] builds on these ideas, learnings to predict a probability distribution of possible vehicle actions from a sequence of images, exploiting temporal information through Long-Short-Term Memory (LSTM). The use of 'privileged' training, where a network simultaneously learns a secondary task, in this case semantic segmentation, is also shown to improve performance. In [4], the idea of conditional imitation learning is introduced, whereby high-level navigation commands are input along with imagery to facilitate an amount of control over the route an autonomous vehicle takes.

In most existing approaches, a neural network is fed an image from a vehicle mounted camera and trained to predict the steering input a human driver would make at the time the image was captured. However, we have identified three major limitations with this method: a) only immediate driving inputs are considered, with no thought as to how vehicle path might change over time; b) driving inputs are learned for a specific vehicle, and adapting a model to a new vehicle is a non-trivial task; c) the relationship between steering input and vehicle movement are not consistent in off-road environments where effective traction may be limited.

In this work we address these limitations by proposing a visual end-to-end path planning approach, whereby a CNN is trained to map future vehicle path directly onto pixels from a vehicle mounted camera. Training data ground truth is created automatically through a novel visual odometry-based pixel labelling approach. This addresses the identified limitations of existing end-to-end autonomous driving approaches [2], [12], [14] by predicting a path that takes account of future changes in direction and does not rely on a direct link to driver inputs. Furthermore, the output of this process could be combined with semantic scene understanding, such as the approach described in [8], to identify upcoming terrain and adjust vehicle driving parameters accordingly.

Subsequently, we use this automatically labelled data to train three state-of-the-art CNN architectures, each originally designed to perform a different segmentation or classification task. We also create a test dataset in the same manner, which we use to carry out a quantitative analysis of the performance of our approach using the three architectures.

2. Approach

The problem we are solving is the prediction of the path that a human driver would take through an off-road environment, made from a single image of that environment taken by a forward-facing vehicle-mounted camera. Our approach involves the automated labelling of training-data via the tracking of a human-driven vehicle using visual odometry, then using this data to train a CNN to map future vehicle path to image pixels.

2.1. Automated Dataset Creation

Our training data comprises individual color images captured by vehicle-mounted stereoscopic camera and corresponding labelled binary ground truth images.

Data was initially captured by stereoscopic video camera mounted on a human-driven off-road vehicle. To select the frames that will form our dataset, we begin at the start of a video sequence and look for the first frame containing movement, f_0, for which we create a label image L of matching dimensions with every pixel labelled as 'not path'. 3D transformation matrices $[T_1] \dots [T_n]$ are computed between f_0 and subsequent frames $f_1 \dots f_n$ using the stereo visual odometry approach of Geiger et al. [6]. These matrices give us the relative camera location and orientation in each of these frames, from which we can compute vehicle footprint. By projecting this footprint into image space at f_0, we can label all pixels the vehicle drives over accordingly in L.

This process, illustrated in Fig. 1, continues until the magnitude of the global transformation vector between the camera positions in f_0 and f_n is greater than distance threshold $D = 20$ m, at which point the process is started again using the frame midway between f_0 and f_n as the new starting point. This visual odometry step is only required for the creation of ground-truth training data, and so output is manually checked to ensure errors are not introduced that may propagate through the process and affect network convergence.

In total, our dataset comprises ~1000 RGB images of dimensions 512×288 along with corresponding binary ground truth images of the same dimensions. We use a 90/10 split to divide our data into training and test sets.

2.2. Network Architectures

We train three CNN models: Segnet [1], Fully Convolutional Network (FCN) [13], and u-net [15]. SegNet was motivated by semantic segmentation of road-scenes and uses an encoder based on VGG16 [16], comprising thirteen 3×3 convolutions, and a symmetrical decoder which uses max-pooling indices retained from the encoder to inform upsampling operations. FCN [13], also based on VGG16, uses 1×1 convolutions to predict class likelihoods at each

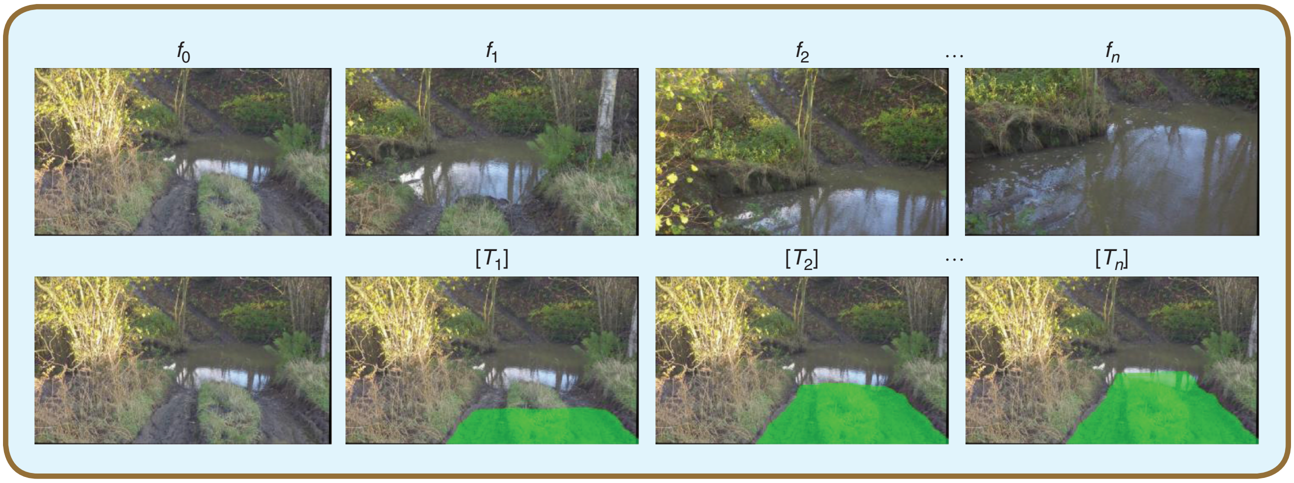

FIG 1 Example sequence from our data set: starting from frame f_0, transformation matrices $[T_1]$ to $[T_n]$ are computed for camera position in n subsequent frames, from which vehicle footprint can be calculated and translated back into image space so that path pixels can be labelled. Top row contains original frames f_0 to f_n, while bottom row shows aggregate computed footprint at each frame overlaid onto f_0.

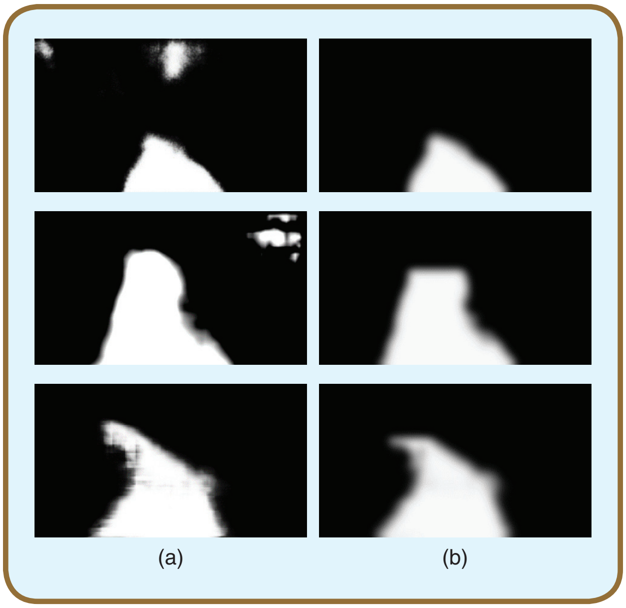

FIG 2 Example output path confidence maps, (a) before and (b) after post processing.

scale of the decoder network. U-net [15], originally motivated by segmentation of medical imagery, uses a more compact architecture, with the encoder and decoder each comprising nine 3×3 convolutions, however residual connections between corresponding pooling and upsampling layers retain high-frequency information that would otherwise be lost to pooling.

In all cases, input is a three-channel color image and output is a one-channel map of the same dimensions, where each pixel value encodes the likelihood that it is part of the future vehicle path. We use batch normalization and rectified linear activation functions after each convolution, and dropout is used to reduce overfitting.

Training images are input in batches of 6, cross entropy loss is computed per batch, and stochastic gradient descent is used to subsequently adjust network weights. Training continues until no further improvement in results is observed.

2.3. Post Processing

CNN output is confidence map $C\{0 \rightarrow 1\}$ that expresses the likelihood that each pixel belongs to the class ‘path’. We apply a post-processing step to C to give the final path confidence map for evaluation against ground truth. Firstly, we use stereo disparity data to compute the distance from the camera to each pixel location in the image, and any pixel further than the distance threshold $D = 20$ m is set to 0, as these pixels will have been ignored during the ground truth creation process. We then convolve the image with a Gaussian kernel of $\sigma = 6$ to smooth out any high-frequency noise.

Next, we set the confidence values of all pixels that are disconnected from the main path segment to 0. For the purposes of this step, we use a very low path confidence threshold δ and set all pixels where $C < \delta$ to 0. Empirically, we found a value of $\delta = 0.025$ to give the best results. If the image contains multiple disconnected path segments, we determine which to consider the actual path by finding the pixel where $C > \delta$ closest to the centre of the bottom of the image, and performing a flood fill that treats pixels with a value of 0 as component boundaries. Any pixel that is outside of the component filled by this operation is set to 0. Some examples of output confidence maps before and after post-processing are shown in Fig. 2.

2.4. Evaluation Methodology

We evaluate the performance of the three trained networks, both with and without the post processing steps detailed above, using our test dataset. In all cases we threshold the output path confidence map such that any pixel that satisfies the condition $C > 0.5$ is labelled ‘path’. We compare the output to the ground truth and compute accuracy, precision, recall and intersection over union (IoU).

3. Results

Our results are shown in Table 1 with illustrative examples shown in Fig. 3, based on an evaluation over our test dataset.

In terms of accuracy, the performance was similar across all three network types—SegNet and u-net both demonstrated an accuracy of 0.95, while FCN did slightly worse with 0.94 - and the effect of post-processing was negligible. Looking at recall, we again see very similar performance from SegNet and u-net while FCN performs slightly worse, however in this case post-processing degraded performance: from 0.86 to 0.85 in the case of SegNet and u-net and from 0.84 to 0.82 in the case of FCN. The opposite is true of precision, which increased slightly with post processing—from 0.84 to 0.85 in the case of FCN, from 0.86 to 0.88 in the case of SegNet and to from 0.86 to 0.89 in the case of U-Net. This is because the post-processing step will have caused more pixels to change from ‘path’ to ‘not path’ than vice versa.

Regarding IoU, SegNet performed best without post-processing (0.76), however u-net output would appear to benefit the most from post-processing, improving from 0.75 to 0.77. Again, FCN performed worst (0.72), and neither the results from it nor SegNet showed any improvement with post-processing. We believe IoU to be the most useful metric for measuring performance at this task as it takes account of both false positives and false negatives while ignoring true negatives, which make up a significant proportion of the data and are part of the reason accuracy is so high.

4. Conclusions

In this work we have proposed an approach to off-road path prediction that combines a novel method to automatically label training data with state-of-the-art CNN architectures designed for semantic segmentation tasks [1], [13], [15].

We created our own off-road dataset which we used to train networks based on SegNet, FCN, and u-net approaches, which we then evaluated over our test dataset. Overall, the best results were obtained from u-net, which considering its advantages in terms of speed and memory usage would make it ideally suited for deployment on an autonomous vehicle.

Our approach addresses several limitations of existing end-to-end driving methods [2], [12], [14], in which a neural network only learns to predict immediate driver control inputs.

Table 1. Results from each configuration.

	Accuracy	Recall	Precision	IoU
SegNet	**0.95**	**0.86**	0.87	0.76
SegNet post processed	**0.95**	0.85	0.88	0.76
FCN	0.94	0.84	0.84	0.72
FCN post processed	0.94	0.82	0.85	0.72
U-Net	**0.95**	**0.86**	0.86	0.75
U-Net post processed	**0.95**	0.85	**0.89**	**0.77**

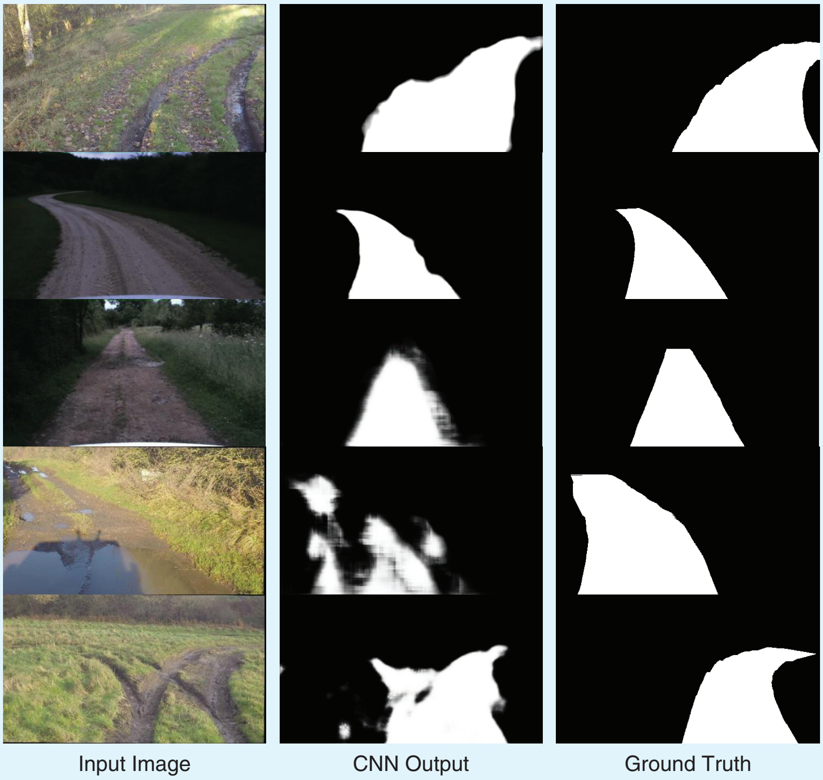

FIG 3 Samples from our test data set. Rows 1–3: good results obtained respectively from FCN, Segnet and u-net; row 4: a poor result, possibly caused by shadows and water on the ground; row 5: an example of a fork in front the vehicle creating two valid paths, although our ground truth only includes the path that the vehicle originally took.

About the Authors

Christopher J. Holder received a PhD in Computer Science at Durham University, UK, where he specialised in the application of deep learning techniques to off-road autonomous driving. He has been a researcher at the Institute for Infocomm Research, Singapore, and is currently a post-doctoral researcher at Durham University. His research focuses on the application of deep learning to visual problems. c.j.holder@durham.ac.uk

Toby P. Breckon is a Professor within the Department of Computer Science, Durham University, UK. He leads a range of research activity in the domain of computer vision and image processing and holds a PhD in informatics (computer vision) from the University of Edinburgh (UK). He has been a visiting member of faculty at the Ecole Supérieure des Technologies Industrielles Avancées (France), Northwestern Polytechnical University (China), Shanghai Jiao Tong University (China) and Waseda University (Japan). toby.breckon@durham .ac.uk

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: a deep convolutional encoder–decoder architecture for scene segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017.

[2] M. Bojarski et al., “End to end learning for self-driving cars,” arXiv Preprint, arXiv:1604.07316, 2016.

[3] Carnegie Robotics LLC, MultiSense Stereo Compact and Accurate 3D Data Collection, 2014. [Online]. Available: http://files.carnegierobotics.com/products/MultiSense_S21/MultiSense_Stereo_brochure.pdf

[4] F. Codevilla, M. Miiller, A. López, V. Koltun, and A. Dosovitskiy, “End-to-end driving via conditional imitation learning,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 2018.

[5] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, “Imagenet: a large-scale hierarchical image database,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009.

[6] A. Ess, T. Müller, H. Grabner, and L. J. Van Gool, “Segmentation-based urban traffic scene understanding,” in *Proc. British Machine Vision Conf.*, 2009.

[7] A. Geiger, J. Ziegler, and C. Stiller, “StereoScan: dense 3D reconstruction in real-time,” in *Proc. Intelligent Vehicles Symp.*, 2011.

[8] C. J. Holder, T. P. Breckon, and X. Wei, “From on-road to off: transfer learning within a deep convolutional neural network for segmentation and classification of off-road scenes,” in *Proc. European Conf. Computer Vision*, 2016.

[9] Y. Jia et al., “Caffe: convolutional architecture for fast feature embedding,” in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014.

[10] H. Kong, J. Y. Audibert, and J. Ponce, “General road detection from a single image,” *IEEE Trans. Image Process.*, vol. 19, no. 8, pp. 2211–2220, 2010.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Adv. Neural Inf. Process. Syst.*, 2012.

[12] Y. LeCun, E. Cosatto, J. Ben, U. Muller, and B. Flepp, “End-to-end learning of vision-based obstacle avoidance for off-road robots,” in *Proc. Learning@Snowbird Workshop*, Apr. 2004.

[13] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2015.

[14] D. A. Pomerleau, “ALVINN: an autonomous land vehicle in a neural network,” *Adv. Neural Inf. Process. Syst.*, 1989.

[15] O. Ronneberger and P. Fischer, “Brox T. U-Net: convolutional networks for biomedical image segmentation,” in *Proc. Int. Conf. Medical Image Computing and Computer-Assisted Intervention*, 2015.

[16] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv Preprint, arXiv:1409.1556, 2014.

[17] S. Thrun et al., “Stanley: the robot that won the DARPA grand challenge,” *J. Field Robot.*, vol. 23, no. 9, pp. 661–692, 2006.

[18] H. Xu, Y. Gao, F. Yu, and T. Darrell, “End-to-end learning of driving models from large-scale video datasets,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017.