ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Robust Visual Terrain Classification with Recurrent Neural Networks

Sebastian Otte[1*], Stefan Laible[1], Richard Hanten[1], Marcus Liwicki[2] and Andreas Zell[1]

1- University of Tuebingen - Cognitive Systems Group
Tuebingen - Germany

2- University of Kaiserslautern - Multimedia Analysis and Data Mining
Kaiserslautern - Germany

Abstract. A novel approach for robust visual terrain classification by generating feature sequences on repeatedly mutated image patches is presented. These sequences providing the feature vector progress under a certain image operation are learned with Recurrent Neural Networks (RNNs). The approach is studied for image patch based terrain classification for wheeled robots. Thereby, various RNN architectures, namely, standard RNNs, Long Short Term Memory networks (LSTMs), Dynamic Cortex Memory networks (DCMs) as well as bidirectional variants of the mentioned architecture are investigated and compared to recently used state-of-the-art methods for real-time terrain classification. The results show that the presented approach outperforms previous methods significantly.

1 Introduction

For autonomous navigation in outdoor environments terrain classification is an important ability of a mobile robot. However, it is a challenging task, because terrain may look quite different at different spots, there are illumination changes and motion blur. It is therefore crucial to have a classifier that can cope with this kind of data and reliably delivers good results. Since classification results are needed for further tasks like path planning, the classifier should also provide real-time efficiency.

In previous work, we investigated terrain classification for driving and flying robots [1, 2]. In addition to various feature descriptors, different classifiers have been tested. Here, Random Forests (RFs) [3], an ensemble learning method, always performed better than the other tested classifiers, like, e.g., SVMs.

The contribution of this paper is twofold. First, instead of using single feature vectors for the terrain classification, feature progression sequences are used. Hereby, such a sequence is gained by repeatedly mutating the original image patch, e.g, sub-sampling or blurring, and recomputing the feature vector after each mutation. Second, these sequences are classified with several recurrent neural network architectures, namely, standard Recurrent Neural Networks (RNNs), Long Short Term Memories (LSTM) [4] and Dynamic Cortex Memories (DCMs) [5] as well as bidirectional variants of them.

*Corresponding author, sebastian.otte@uni-tuebingen.de

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

We show that the combination of feature progression sequences and recurrent architectures provide superior classification results and outperform RFs on terrain classification for wheeled robots significantly.

2 Visual Feature Progression

In the following we outline a methodical concept that enables us to utilize the progression of a feature vector under a repeatedly mutated image patch. Consider therefore Figure 1.

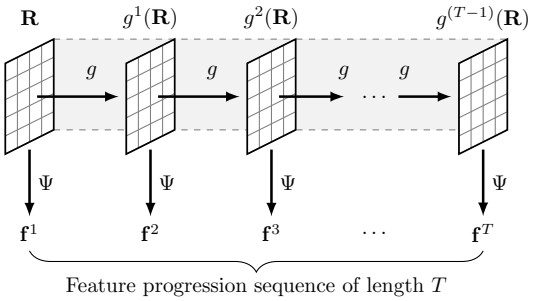

Fig. 1: Illustration of the feature progression sequence acquisition. The original image patch R is consecutively mutated $T-1$ times using a mutation operation g. For each of those T image patches a feature vector using a given feature generator Ψ is computed.

The procedure starts with a given image patch $\mathbf{R}$. A feature vector for the patch $\mathbf{R}$ is computed with a feature generator Ψ, e.g., texture features like LTP [3] or TSURF [1]. Then the patch is mutated with an image operator g. In this paper the experiments are based on a scale and blur operator as it is used to build Gaussian image pyramids. Hereafter, a feature vector of the mutated patch is computed. This process is performed iteratively until T feature vectors are generated. It is important to our approach that for the original patch and each of its mutated instances, the feature vector is computed in the same way providing the same dimension. Due to the overall process, a sequence is computed representing the successive feature vector progression under g.

It should be mentioned that in [6] experiments were presented, where digit and object recognition with bidirectional LSTMs was performed on sequences with repeatedly blurred images. However, learning pure image data instead of texture features was successful on our data.

3 Recurrent Neural Networks

With the introduction of the Long Short Term Memory (LSTM) [4] artificial neural networks (ANNs), especially recurrent neural networks (RNNs), enjoy

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

again emerging popularity. Due to their cyclic connections RNNs can learn temporal dependencies in data sequences. In contrast to standard RNNs, networks containing LSTM blocks [4], which are themselves a special kind of a recurrent network that can be seen as differentiable memory cells, are even able to deal with long time-lag problems. But LSTMs provide additional abilities. These are, for instance, that they are able to learn highly non-uniformly compressed sequences [7],[8], where the position of important input events is difficult to predict, or the adequate recognition of even noisy sequences [8].

Besides RNNs and LSTMs, also networks consisting of recently introduced Dynamic Cortex Memorys (DCMs) [5] are applied in this paper. A DCM block is in principle an LSTM block with *forget gates* [9] and *peep-hole connections* [10] but has several novel weighted connections within each block, i.e., connections from each gate to each other gate, and a self-recurrent connection for each gate. Nonetheless, even if there are more connections per block, DCM networks tend to require less blocks and, thus, less weights than LSTM networks to achieve similar or even better results. In fact DCMs were shown to converge faster than LSTMs during training and generalize better. For both LSTMs and DCMs the gradient is computed with *back-propagation through time* (BPTT) as proposed in [11].

In classical *unidirectional* recurrent architectures, only a "past" context is provided. To provide also a "future" context *bidirectional recurrent neural networks* (BRNNs) were introduced [12, 11] by keeping separate hidden layers, one for forward computation and the other one for backward computation. The past context of the forward layer and the future context of the backward layer are present in the output layer at each time step.

The network architectures contain two hidden layer (or three in case of bidirectionality). The first hidden layer is just a simple non-recurrent fully-connected non-linear layer that can be seen as a kind of input reduction to significantly reduce the relatively high input dimension of the data passed into the recurrent hidden layer. The recurrent layer consists of non-linear units, LSTM blocks or DCM blocks, respectively. In the bidirectional architectures the recurrent layer is duplicated, such that there is one layer for the past and one for the future context. As general hidden activation function a scaled and shifted sigmoid covering the interval $[-1, 1]$ is used in all architectures with exception of the gates in LSTMs and DCMs, which are activated with a standard sigmoid. All architectures use softmax activation in the output layer.

4 Experiments

Our experiments are based on images taken by a mobile robot driving in an outdoor environment with four terrain types, namely asphalt, cobblestones, grass, and gravel (see Fig. 2). The camera was mounted at a height of approximately 45 cm and at an angle of about 15 degrees to the horizontal. From these VGA images, we hand-labeled 135 images to get ground truth. Each image is then divided into equally–sized grid cells with a cell length of 30 pixels. There is

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Fig. 2: The mobile robot used for ground image acquisition (left). The four terrain classes used in the ground dataset (right). From left to right and top to bottom: Asphalt, cobblestones, grass, gravel

great variation even within the grid cells of the same terrain type, enhanced by perspective distortion caused by the camera angle.

For feature extraction we use Local Ternary Patterns (LTP) [13], since previous work has shown that they provide good results for this kind of data [14]. LTPs are an extension of Local Binary Patterns (LBP) [15] and LBPs are essentially histograms of binary-encoded differences in pixel intensity. While LBP is parameter-free, LTP has a parameter to threshold intensity differences into three categories and yields a 512-dimensional feature vector. The resulting dataset consists of 4,000 samples with 1,000 samples per class. Furthermore 10 fold cross-validation was applied. To train the neural networks, we used gradient descent with momentum term. The learning rate was set to 0.001, the momentum rate to 0.9 and the maximum number of epochs was limited to 100 (but rarely reached due to early stopping). All results in this paper concerning neural networks are achieved using the JANNLab framework [16].

5 Results

Table 1 shows the classification accuracy results yielded by RFs and Table 2 shows results yielded by various various RNNs. All recurrent architectures used a reduction layer size of 16 neurons, which we figured out to be good choice. It can be clearly identified that all tested RNN architectures are significantly better than the RFs tested on this dataset. The best accuracy achieved by an RF with 200 trees is about 76.88 %, while the best neural network, a BRNN with a hidden layer size of 16, achieved an accuracy of 83.49 %. It should be mentioned that we even tested RFs with 1,000 and more trees as well, but this did not further improve the results. Nonetheless, it is conspicuous that all networks providing a capacity of at least eight cells or four blocks respectively, yield nearly the same performance regardless whether they are bidirectional or use memory blocks like LSTMs or DCMs. However, during the experiments the latter kept much more stable when varying learning and momentum rate. We also observed that on the given training data DCMs converged slightly faster than LSTMs.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

Table 1: Classification results for Random Forrests

Number of trees	Sequence Length	Test Accuracy [%]
100	1	71.03 ± 2.77
200	1	72.05 ± 2.41
500	1	71.70 ± 2.64
100	8	75.85 ± 1.65
200	8	76.88 ± 1.53
500	8	76.83 ± 1.35

Table 2: Classification results for various RNN architectures

Architecture	Size	Test accuracy [%] (unidirectional)	(bidirectional)
RNN	2	72.71 ± 1.66	80.59 ± 1.08
RNN	4	80.50 ± 1.08	82.38 ± 1.00
RNN	8	82.23 ± 1.23	83.37 ± 1.22
RNN	16	83.10 ± 1.01	83.49 ± 1.38
RNN	32	82.89 ± 1.03	83.44 ± 1.38
LSTM	2	78.75 ± 1.41	82.02 ± 1.12
LSTM	4	82.27 ± 1.15	82.59 ± 1.23
LSTM	8	82.78 ± 1.17	82.93 ± 1.02
LSTM	16	83.20 ± 0.84	83.04 ± 1.00
LSTM	32	83.06 ± 0.99	83.03 ± 1.00
DCM	2	78.82 ± 1.28	82.03 ± 1.32
DCM	4	82.47 ± 1.01	82.42 ± 1.19
DCM	8	82.75 ± 0.94	82.78 ± 1.11
DCM	16	83.05 ± 0.99	83.22 ± 1.01
DCM	32	83.21 ± 0.95	83.40 ± 0.88

Moreover, we experimented with Multi-Layer Perceptrons (MLPs). While they yielded (after some fine tuning) even better results than RFs on the non-sequential data, they failed on the sequences, due to the large amount of required weights, which results in strongly over-fitted networks. This observation paired with the results from above lead us to the assumption that probably the most important aspect why RNNs perform so well on this task is the effect of shared weights, forcing the networks to generalize more from the given data. However, their performance drops when the vector order within the sequences is changed, which hints at the importance of the recurrence.

6 Conclusion

In this paper we presented an approach for robust visual terrain classification. In contrast to previous approaches not only single feature vectors were com-

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

puted, but sequences representing the progression of feature vector over repeatedly applied image mutations. This acquisition method in combination with the recurrent neural networks for the classification improved the recognition accuracy compared to previously used RFs from 76.88 % to 83.49 % on terrain image patches recorded by a wheeled robot.

References

[1] Y. N. Khan, A. Masselli, and A. Zell. Visual terrain classification by flying robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 498 –503, St. Paul, Minnesota, USA, may 2012.

[2] S. Laible, Y. N. Khan, K. Bohlmann, and A. Zell. 3d lidar- and camera-based terrain classification under different lighting conditions. In *Autonomous Mobile Systems 2012*, Informatik aktuell, pages 21–29. Springer Berlin Heidelberg, 2012.

[3] L. Breiman. Random forests. In *Machine Learning*, pages 5–32, 2001.

[4] S. Hochreiter and J. Schmidhuber. Long Short-Term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[5] Sebastian Otte, Marcus Liwicki, and Andreas Zell. Dynamic Cortex Memory: Enhancing Recurrent Neural Networks for Gradient-Based Sequence Learning. In Stefan Wermter et al., editor, *Artificial Neural Networks and Machine Learning – ICANN 2014*, number 8681 in Lecture Notes in Computer Science, pages 1–8. Springer International Publishing, September 2014.

[6] S. Biswas, M.Z. Afzal, and T. Breuel. Using recurrent networks for non-temporal classification tasks. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 135–140, July 2014.

[7] S. Otte, M. Liwicki, D. Krechel, and A. Dengel. Local feature based online mode detection with recurrent neural networks. In *Proceedings of the 13th International Conference on Frontiers in Handwriting Recognition*, pages 533–537. IEEE Computer Society, 2012.

[8] S. Otte, C. Otte, A. Schlaefer, L. Wittig, G. Hüttmann, D. Drömann, and A. Zell. OCT A-Scan based lung tumor tissue classification with bidirectional long short term memory networks. In *Machine Learning for Signal Processing (MLSP), 2013 IEEE International Workshop on*, 2013.

[9] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12:2451–2471, 1999.

[10] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber. Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research*, 3:115–143, 2002.

[11] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610, July 2005.

[12] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, November 1997.

[13] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *Trans. Img. Proc.*, 19(6):1635–1650, June 2010.

[14] S. Laible, Y. N. Khan, and A. Zell. Terrain classification with conditional random fields on fused 3d lidar and camera data. In *European Conference on Mobile Robots (ECMR 2013)*, Barcelona, Catalonia, Spain, September 2013.

[15] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, January 1996.

[16] S. Otte, D. Krechel, and M. Liwicki. JANNLab neural network framework for java. In *Poster Proceedings Conference MLDM 2013*, pages 39–46, New York, USA, 2013. ibai-publishing.