26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), Bruges, Belgium, April 2018.

Hierarchical Recurrent Filtering for Fully Convolutional DenseNets

Jörg Wagner[1,2], Volker Fischer[1], Michael Herman[1] and Sven Behnke[2]

1- Bosch Center for Artificial Intelligence - 71272 Renningen - Germany

2- University of Bonn - Computer Science VI, Autonomous Intelligent Systems - Friedrich-Ebert-Allee 144, 53113 Bonn - Germany

Abstract. Generating a robust representation of the environment is a crucial ability of learning agents. Deep learning based methods have greatly improved perception systems but still fail in challenging situations. These failures are often not solvable on the basis of a single image. In this work, we present a parameter-efficient temporal filtering concept which extends an existing single-frame segmentation model to work with multiple frames. The resulting recurrent architecture temporally filters representations on all abstraction levels in a hierarchical manner, while decoupling temporal dependencies from scene representation. Using a synthetic dataset, we show the ability of our model to cope with data perturbations and highlight the importance of recurrent and hierarchical filtering.

1 Introduction

A robust and reliable perception and interpretation of the environment is a crucial competence of autonomous systems. Deep learning based methods greatly advanced the generation of robust environment representations and dominate the majority of perception benchmarks. From a safety point of view, a major drawback of popular datasets is their recording at daytime under good or normal environment conditions. In order to deploy autonomous systems in an unconstrained world without any supervision, one has to make sure that they still work in challenging situations such as sensor outages or heavy weather. These situations induce failures of the perception algorithm, which are not solvable by just using a single image. We denote these failures in accordance to Kendall *et al.* [1] as aleatoric failures. To tackle such failures, one has to enhance the information provided to the perception algorithm. This can be achieved by using a better sensor, fusing information of multiple sensors, utilizing additional context knowledge, or by integrating information over time.

In this paper, we focus on using temporal coherence to reduce aleatoric failures of a single-frame segmentation model. We build upon the Fully Convolutional DenseNet (FC-DenseNet) [2] and propose a temporal filtering concept, which extends it to work with multiple frames. The temporal integration is achieved by recurrently filtering the representations on all abstraction levels in a hierarchical manner. Due to the hierarchical nature of the filter concept, our model—the Recurrent Fully Convolutional DenseNet (RFC-DenseNet)—can utilize temporal correlations on all abstraction levels. Additionally, the RFC-DenseNet decouples temporal dependencies from scene representation, which increases its transparency and enables a direct transformation of any FC-DenseNet.

Despite its recurrent nature, the proposed model is highly parameter efficient. Using simulated data, we show the ability of RFC-DenseNet to reduce aleatoric failures and highlight the importance of recurrent and hierarchical filtering.

2 Related Work

The majority of research, focused on using temporal information to reduce aleatoric failures of segmentation models, applies a non-hierarchical filter approach. Valipour *et al.* [3] generate a representation for each image in a sequence and use a Recurrent Network to temporally filter them. Jin *et al.* [4] utilize a sequence of previous images to predict a representation of the current image. The predicted representation is fused with the current one and propagated through a decoder model. Similar approaches exist, which apply post-processing steps on top of frame segmentations. Kundu *et al.* [5] use a Conditional Random Field operating on an Euclidean feature space, optimized to minimize the distance between features associated with corresponding points in the scene. Our approach differs from the above methods due to its hierarchical nature.

Tran *et al.* [6] build a semantic video segmentation network using spatio-temporal features computed with 3D convolutions. We differ from this approach due to the explicit utilization of recurrent filters. The Recurrent Convolutional Neural Network of Pavel *et al.* [7] is similar to our architecture. This method uses layer-wise recurrent self-connections as well as top-down connections to stabilize representations. The approach focuses on a fully recurrent topology, while our approach decouples temporal filtering and scene representation. Additionally, our approach uses a dense connection pattern to get an improved signal flow.

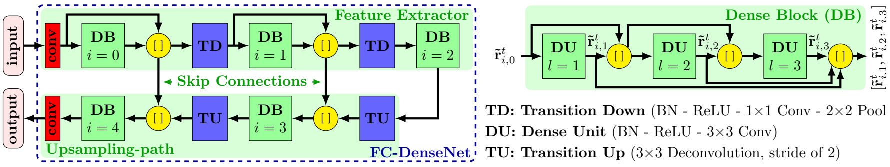
3 Recurrent Fully Convolutional DenseNets

3.1 Revisiting the Fully Convolutional DenseNet (FC-DenseNet)

The FC-DenseNet [2] is constructed by using a fully convolutional version of DenseNet as feature extractor, utilizing a Dense Block (DB) enhanced upsampling path, and interlinking both paths using skip connections (Fig. 1). The DenseNet, used in the feature extractor, is a convolutional network, which iteratively adds features to a stack, the global feature state. Newly added features $\tilde{\mathbf{r}}^t_{i,l}$ are computed using all previous ones $[\tilde{\mathbf{r}}^t_{i,l-1}, \dots, \tilde{\mathbf{r}}^t_{i,0}]$ of matching spatial size:

$$\tilde{\mathbf{r}}^t_{i,l} = f^{DU}_{i,l}([\tilde{\mathbf{r}}^t_{i,l-1}, \dots, \tilde{\mathbf{r}}^t_{i,0}]; \theta^{DU}_{i,l}); \tag{1}$$

where $f^{DU}_{i,l}$ is the function of the Dense Unit (DU) with parameters $\theta^{DU}_{i,l}$.


input
output
conv
conv
DB $i=0$
DB $i=1$
DB $i=2$
DB $i=3$
DB $i=4$
TD
TD
TU
TU
[ ]
Feature Extractor
Skip Connections
Upsampling-path
FC-DenseNet
Dense Block (DB)
$\tilde{\mathbf{r}}^t_{i,0}$
DU $l=1$
DU $l=2$
DU $l=3$
$\tilde{\mathbf{r}}^t_{i,1}$
$\tilde{\mathbf{r}}^t_{i,2}$
$\tilde{\mathbf{r}}^t_{i,3}$
$[\tilde{\mathbf{r}}^t_{i,1}, \tilde{\mathbf{r}}^t_{i,2}, \tilde{\mathbf{r}}^t_{i,3}]$

TD: Transition Down (BN - ReLU - 1×1 Conv - 2×2 Pool
DU: Dense Unit (BN - ReLU - 3×3 Conv)
TU: Transition Up (3×3 Deconvolution, stride of 2)

Fig. 1: FC-DenseNet of depth two with three layers per Dense Block.

3.2 Temporal Representation Filtering

Due to perturbations inherent in the data, the features $\tilde{\mathbf{r}}^t_{i,l}$ computed in each Dense Unit are only a crude approximation of the true representation $\mathbf{r}^t_{i,l}$. To get an improved estimate, we propose to filter them using a Filter Module (FM):

$$\hat{\mathbf{r}}^t_{i,l} = f^{FM}_{i,l}(f^{DU}_{i,l}([\hat{\mathbf{r}}^t_{i,l-1}, \dots, \hat{\mathbf{r}}^t_{i,0}]; \theta^{DU}_{i,l}), \mathbf{m}^{t-1}_{i,l}; \theta^{FM}_{i,l}); \tag{2}$$

where $f^{FM}_{i,l}$ is the filter function with parameters $\theta^{FM}_{i,l}$ and hidden state $\mathbf{m}^{t-1}_{i,l}$.

FC-DenseNet can be transformed into our proposed RFC-DenseNet by using a recurrent version of the Dense Block (see Fig. 2), which employs a Filter Module after each Dense Unit. To compute a robust segmentation, RFC-DenseNet utilizes the information of multiple images. These images are propagated through the feature extractor and the subsequent upsampling path, while taking temporal correlations via the Filter Modules into account.

The RFC-DenseNet adds filtered features $\hat{\mathbf{r}}^t_{i,l}$ to the global feature state of the model. Features $\bar{\mathbf{r}}^t_{i,l}$ computed in each Dense Unit are thus derived from already filtered ones, generating a hierarchy of filtered representations. Due to the hierarchical filter nature, RFC-DenseNet can utilize temporal correlations on all abstraction levels. In comparison, a non-hierarchical filter only has access to a sequence of high-level representations. The availability of all features, required to solve aleatoric failures within the filter, is not guaranteed in such a setting.

The RFC-DenseNet decouples temporal dependencies from scene representation by using dedicated Filter Modules. This property makes it easy to transform any single-image FC-DenseNet into a corresponding multi-image RFC-DenseNet. One could also use the weights of a FC-DenseNet to initialize the non-recurrent part of the corresponding RFC-DenseNet. The decoupling additionally increases the transparency of the model, enabling a dedicated allocation of resources for temporal filtering and hierarchical feature generation (scene representation).

The proposed filter approach can also be employed in other models, but is especially suitable for the FC-DenseNet architecture. The explicit differentiation of newly computed features $\bar{\mathbf{r}}^t_{i,l}$ and features stored in the global feature state makes the temporal filtering very parameter efficient. Each filter only has to process a small number of feature maps—resulting in a moderate increase in the total number of model parameters. The distinct focus on a small feature set in each Filter Module also reduces the computational complexity of the filter task.

3.3 Instances of the Filter Module

We investigated three instances of the Filter Module with increasing complexity. All modules are based on Convolutional Long Short Term Memory cells (Conv-

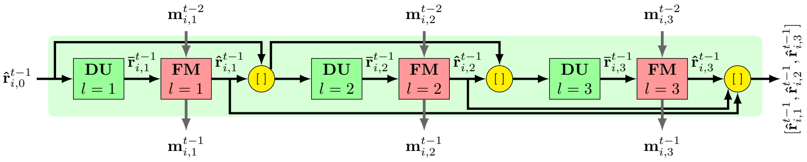

$\mathbf{m}^{t-2}_{i,1}$
$\mathbf{m}^{t-2}_{i,2}$
$\mathbf{m}^{t-2}_{i,3}$
$\hat{\mathbf{r}}^{t-1}_{i,0}$
DU $l=1$
$\bar{\mathbf{r}}^{t-1}_{i,1}$
FM $l=1$
$\hat{\mathbf{r}}^{t-1}_{i,1}$
[ ]
DU $l=2$
$\bar{\mathbf{r}}^{t-1}_{i,2}$
FM $l=2$
$\hat{\mathbf{r}}^{t-1}_{i,2}$
[ ]
DU $l=3$
$\bar{\mathbf{r}}^{t-1}_{i,3}$
FM $l=3$
$\hat{\mathbf{r}}^{t-1}_{i,3}$
[ ]
$[\hat{\mathbf{r}}^{t-1}_{i,1}, \hat{\mathbf{r}}^{t-1}_{i,2}, \hat{\mathbf{r}}^{t-1}_{i,3}]$
$\mathbf{m}^{t-1}_{i,1}$
$\mathbf{m}^{t-1}_{i,2}$
$\mathbf{m}^{t-1}_{i,3}$

Fig. 2: Recurrent Dense Block using a Filter Module after each Dense Unit.

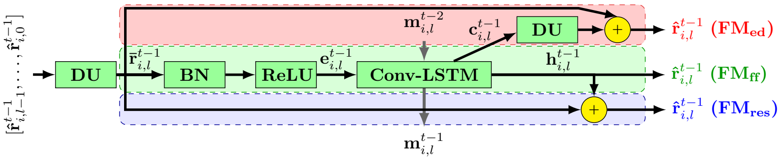

$[\hat{\mathbf{r}}_{i,l-1}^{t-1}, \dots, \hat{\mathbf{r}}_{i,0}^{t-1}]$
DU
$\bar{\mathbf{r}}_{i,l}^{t-1}$
BN
ReLU
$\mathbf{e}_{i,l}^{t-1}$
$\mathbf{m}_{i,l}^{t-2}$
Conv-LSTM
$\mathbf{c}_{i,l}^{t-1}$
DU
$\mathbf{h}_{i,l}^{t-1}$
+
$\mathbf{m}_{i,l}^{t-1}$
$\hat{\mathbf{r}}_{i,l}^{t-1}$ ($\mathbf{FM_{ed}}$)
$\hat{\mathbf{r}}_{i,l}^{t-1}$ ($\mathbf{FM_{ff}}$)
$\hat{\mathbf{r}}_{i,l}^{t-1}$ ($\mathbf{FM_{res}}$)

Fig. 3: Three separate Filter Module instances: $\mathrm{FM_{ff}}$, $\mathrm{FM_{res}}$ and $\mathrm{FM_{ed}}$.

LSTMs), which have proven to produce state-of-the-art results on a multitude of spatio-temporal sequence modeling tasks. The Conv-LSTM is defined by:

$$\begin{aligned}
\mathbf{k}_{i,l}^{t} &= \sigma(\mathbf{W}_{i,l}^{ek} * \mathbf{e}_{i,l}^{t} + \mathbf{W}_{i,l}^{hk} * \mathbf{h}_{i,l}^{t-1} + \mathbf{b}_{i,l}^{k}),\ \forall \mathbf{k} \in \{\mathbf{i}, \mathbf{f}, \mathbf{o}\}\,; && (3)\\
\mathbf{c}_{i,l}^{t} &= \mathbf{f}_{i,l}^{t} \circ \mathbf{c}_{i,l}^{t-1} + \mathbf{i}_{i,l}^{t} \circ \tanh(\mathbf{W}_{i,l}^{ec} * \mathbf{e}_{i,l}^{t} + \mathbf{W}_{i,l}^{hc} * \mathbf{h}_{i,l}^{t-1} + \mathbf{b}_{i,l}^{c}); && (4)\\
\mathbf{h}_{i,l}^{t} &= \mathbf{o}_{i,l}^{t} \circ \tanh(\mathbf{c}_{i,l}^{t}); && (5)
\end{aligned}$$

where $*$ is the convolutional operator, $\circ$ the Hadamard product, and $\mathbf{e}_{i,l}^{t}$ the input. The hidden state $\mathbf{m}_{i,l}^{t-1}$ of Equation 2 is a summary of $\mathbf{c}_{i,l}^{t-1}$ and $\mathbf{h}_{i,l}^{t-1}$. A property of all Filter Modules are matching dimensions between the unfiltered $\bar{\mathbf{r}}_{i,l}^{t}$ and filtered $\hat{\mathbf{r}}_{i,l}^{t}$ representation and a filter size of 3×3 for all kernels $\mathbf{W}_{i,l}^{e*}$.

The Filter Module $\mathbf{FM_{ff}}$ (Fig. 3, green) uses a single Conv-LSTM following the two *pre-activation* layers (Batch Normalization (BN) and Rectified Linear Unit (ReLU)). The number of feature maps stored in the cell state $\mathbf{c}_{i,l}^{t-1}$ matches the number of feature maps of the unfiltered representation. This property restricts the filter capabilities but also limits the number of required parameters.

The Filter Module $\mathbf{FM_{res}}$ (Fig. 3, green and blue) uses the concept of residual learning [8] and applies it to $\mathrm{FM_{ff}}$. The introduced skip connection ensures a direct information and gradient flow, preventing signal degradation.

$\mathbf{FM_{ed}}$ (Fig. 3, green and red) alleviates the limitation on the complexity of the filter, introduced by matching feature dimensions of the unfiltered representation and the cell state. This instance employs an encoder-decoder structure, consisting of the Conv-LSTM and a Dense Unit. The number of feature maps stored in the Conv-LSTM can be chosen to be a multitude α_{ed} of the number of unfiltered maps. A drawback of $\mathrm{FM_{ed}}$ is the increased parameter count.

4 Experimental Results

4.1 Dataset

To evaluate the proposed models, we use a simulated dataset (see Fig. 4). The sequences emulate a 2D environment of 64×64 pixels, in which squares represent dynamic and static objects, rectangles represent borders and walls, and circles represent moving foreground objects. Each square is marked with a random MNIST digit. The dynamic squares elastically collide with all other squares, borders and walls. The moving circles occlude each other, as well as all other objects. The number of objects, their size and color, the color of MNIST digits as well as the velocity of all dynamic objects is randomly sampled for each sequence.

To simulate aleatoric failures, we perturb the data with noise, ambiguities, missing information, and occlusions. Noise is simulated by adding zero mean

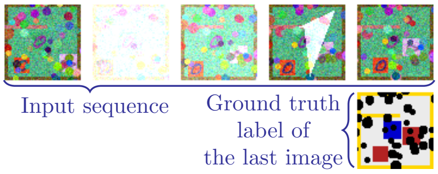


Fig. 4: Test sequence of length 5 with label.

	$\mathrm{FCD_b}$	$\mathrm{FCD_s}$
Model depth	2	2
Layers per DB	9	7
Features per DU	12	8
Features of the first convolution	48	48

Table 1: FC-DenseNets.

Gaussian noise to each pixel. Occlusions are introduced by the foreground circles. To simulate missing information, we increase or decrease the intensity of pixels by a random value and let this offset decay. This effect is added to whole images and to subregions. Ambiguities are simulated using different classes for static and dynamic squares. Our dataset contains 25,000 independently sampled sequences of length 5, which are split into 20,000 training, 4,000 validation and 1,000 test sequences. We additionally generate a clean test set with no aleatoric failures. For the segmentation task, we define 14 classes: background, borders and walls, static squares, circles, and dynamic squares with one class per MNIST digit. A label is only available for the last image in each sequence.

4.2 Models

We perform a grid search to find the best FC-DenseNet ($\mathrm{FCD_b}$). The grid parameters are listed in Table 1. Our temporal models are built based on a smaller version of the FC-DenseNet ($\mathrm{FCD_s}$) to reduce training time.

In total, we train seven temporal models: Four RFC-DenseNets using our filter concept (Table 2), a recurrent, non-hierarchical model $\mathrm{RM_{gf}}$ (cf. [3]), and two non-recurrent models: $\mathrm{TM_{3D}}$ and $\mathrm{TM_{st}}$. $\mathrm{RM_{gf}}$ globally filters the representation generated in the last Dense Block of $\mathrm{FCD_s}$ using $\mathrm{FM_{ed}}$ with $\alpha_{ed} = 0.625$ and a hidden-to-hidden filter size of 9. $\mathrm{TM_{3D}}$ and $\mathrm{TM_{st}}$ are FC-DenseNets: one using 3D convolutions of size 3×3×3 and one operating on stacked input sequences. The filter size of kernels $\mathbf{W}^{h*}_{i,l}$ in our RFC-DenseNets are set to $[9, 5, 3, 5, 9]$. All temporal models, except for $\mathrm{RFCD_{ed2}}$, have roughly the same number of parameters.

	Filter Module
$\mathrm{RFCD_{ff}}$	$\mathrm{FM_{ff}}$
$\mathrm{RFCD_{res}}$	$\mathrm{FM_{res}}$
$\mathrm{RFCD_{ed1}}$	$\mathrm{FM_{ed}}$, $\alpha_{ed} = 1$
$\mathrm{RFCD_{ed2}}$	$\mathrm{FM_{ed}}$, $\alpha_{ed} = 2$

Table 2: RFC-DenseNets.

4.3 Evaluation

We summarize the results in Table 3 by reporting the mean Intersection over Union (mean IoU) on the test dataset, as well as the clean test dataset.

All models which utilize temporal information significantly outperform the single-image FC-DenseNets on the test data—showing the importance of temporal filtering. On the clean test data, the single-image and multi-image models are roughly on par, suggesting that temporal information especially benefits the reduction of aleatoric failures. The superior performance of $\mathrm{FCD_b}$ on the clean

	$\mathrm{FCD_b}$	$\mathrm{FCD_s}$	$\mathrm{RFCD_{ff}}$	$\mathrm{RFCD_{res}}$	$\mathrm{RFCD_{ed1}}$	$\mathrm{RFCD_{ed2}}$	$\mathrm{RM_{gf}}$	$\mathrm{TM_{3D}}$	$\mathrm{TM_{st}}$
Test dataset	45.10 %	43.37 %	67.11 %	67.93 %	**69.20** %	68.42 %	65.43 %	60.84 %	50.42 %
Clean test dataset	93.06 %	91.00 %	92.03 %	92.18 %	**94.37** %	93.00 %	92.13 %	89.78 %	89.37 %

Table 3: Mean IoU of the different models on the test and clean test dataset.

test data compared to most of the temporal models can most likely be attributed to its increased non-temporal depth and width.

The mean IoUs of the different RFC-DenseNets suggest a correlation between filter complexity and performance. However, the difference is relatively small. The performance of $\mathrm{RFCD}_{\mathrm{ed2}}$ is unexpectedly poor in comparison to $\mathrm{RFCD}_{\mathrm{ed1}}$. Looking at class-wise IoUs did not provide any additional insight regarding possible systematic failures. We plan to further investigate other recurrent regularization techniques to improve the performance of $\mathrm{RFCD}_{\mathrm{ed2}}$.

The hierarchical RFC-DenseNet models outperform the non-hierarchical, recurrent baseline $\mathrm{RM}_{\mathrm{gf}}$ by $2\,\%$ to $4\,\%$ on the test data, showing the superiority of our models in the reduction of aleatoric failures. We suspect that our hierarchical filter concept better utilizes temporal information, compared to a non-hierarchical approach. Taking only the diverse MNIST digits into account, the performance difference increases further. For these classes, it is important to model low-level temporal dependencies. The non-hierarchical approach possibly suffers, because of the loss in scene details from lower to upper layers.

The performance of the non-recurrent models, $\mathrm{TM}_{\mathrm{3D}}$ and $\mathrm{TM}_{\mathrm{st}}$, is inferior to the recurrent ones. This is most likely due to the explicit temporal structure of recurrent models, which benefits the detection of temporal correlations.

5 Conclusion

In this work, we proposed a parameter-efficient approach to temporally filter the representations of the FC-DenseNet in a hierarchical fashion, while decoupling temporal dependencies from scene representation. Using a synthetic dataset, we showed the benefits of using temporal information in regards to aleatoric failures, as well as the advantages introduced by our recurrent and hierarchical filtering concept. In the future, we plan to evaluate our approach on real-world datasets.

References

[1] A. Kendall and Y. Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In *arXiv:1703.04977*, 2017.

[2] S. Jégou, M. Drozdzal, D. Vázquez, A. Romero, and Y. Bengio. The one hundred layers tiramisu: Fully convolutional DenseNets for semantic segmentation. In *CVPR*, 2017.

[3] S. Valipour, M. Siam, M. Jagersand, and N. Ray. Recurrent fully convolutional networks for video segmentation. In *WACV*, pages 29–36, 2017.

[4] X. Jin, X. Li, H. Xiao, X. Shen, Z. Lin, J. Yang, Y. Chen, J. Dong, L. Liu, et al. Video scene parsing with predictive feature learning. *arXiv preprint arXiv:1612.00119*, 2016.

[5] A. Kundu, V. Vineet, and V. Koltun. Feature space optimization for semantic video segmentation. In *CVPR*, pages 3168–3175, 2016.

[6] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Deep end2end voxel2voxel prediction. In *CVPR Workshops*, pages 17–24, 2016.

[7] M. S. Pavel, H. Schulz, and S. Behnke. Object class segmentation of RGB-D video using recurrent convolutional neural networks. *Neural Networks*, 88:105–113, 2017.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, June 2016.