Hybrid Rotation Invariant Networks for small sample size Deep Learning

Alexander Katzmann[1,3] ALEXANDER.KATZMANN@SIEMENS.COM
Marc-Steffen Seibel[1,3] MARC-STEFFEN.SEIBEL@SIEMENS.COM
Alexander Mühlberg[1] ALEXANDER.MUEHLBERG.EXT@SIEMENS.COM
Michael Sühling[1] MICHAEL.SUEHLING@SIEMENS.COM

[1] *Siemens Healthcare GmbH, Computed Tomography, 91301 Forchheim, Germany*

Dominik Nörenberg[2] DOMINIK.NOERENBERG@MED.UNI-MUENCHEN.DE
Stefan Maurus[2] STEFAN.MAURUS@MED.UNI-MUENCHEN.DE
Thomas Huber[2] THOMAS.HUBER@MED.UNI-MUENCHEN.DE

[2] *University Hospital Grohadern, Ludwig-Maximilians-University Munich, Department of Radiology, 81377 Munich, Germany*

Horst-Michael Groß[3] HORST-MICHAEL.GROSS@TU-ILMENAU.DE

[3] *Ilmenau, University of Technology, Neuroinformatics and Cognitive Robotics Lab, 98693 Ilmenau, Germany*

Editors: Under Review for MIDL 2019

Abstract

Medical image analysis using deep learning has become a topic of steadily growing interest. While model capacity is continiously increasing, limited data is still a major issue for deep learning in medical imaging. Virtually all past approaches work with a high amount of regularization as well as systematic data augmentation. In explorative tasks realistic data augmentation with affine transformations may not always be possible, which prevents models from effective generalization. Within this paper, we propose inherently rotationally invariant convolutional layers enabling the model to develop invariant features from limited training data. Our approach outperforms classical convolutions on the CIFAR-10, CIFAR-100, and STL-10 datasets. We show the transferability to clinical scenarios by applying our approach on oncologic tasks for metastatic colorectal cancer treatment assessment and liver lesion segmentation in pancreatic cancer patients.

Keywords: rotational invariance, regularization, colorectal cancer, pancreatic cancer

1. Introduction

With the rise of deep learning and its success in a wide range of applications, within the last years (semi-)automated medical image analysis using deep learning has become a topic of steadily growing interest (Litjens et al., 2017; Shen et al., 2017). A major reason for the success of deep learning is its high model complexity. However, with an increasing amount of trainable parameters, deep learning models become highly prone to overfitting.

Most - or virtually all - medical applications of deep learning have to handle this problem, i.e. model regularization, in one or the other way. Standard techniques involve dropout (Srivastava et al., 2014), batch normalization (Ioffe and Szegedy, 2015), data augmentation and various forms of dimensionality or effective model complexity reduction, such as architectural bottlenecking through sparse representations or using denoising autoencoders (Vincent et al., 2008). Most of the mentioned techniques increase training set variance by (effectively) introducing semi-synthetically generated samples with no additional information, which eventually leads to a reduced model parameter variance, and thus, a reduced effective model complexity. This process can be seen as an a-priori knowledge-guided model regularization. However, the applied techniques differ: While dropout, batch-normalization, and architectural regularization tackle the problem from within the models training task, data augmentation is done from the outside. Non-augmenting model regularization induces invariance in a form which is part of the optimization process, thus making it possible for the model to learn more efficiently, while data augmentation assumes that semi-synthetically generated samples are semantically following some form of invariance visually perceived by human. This, however, might be a wrong assumption.

Neural networks apply a vast amount of non-linear projections. Meanwhile handy tools for visualization are existent, and while they can be considered as being generally helpful in neural network development, a) their introduction and distribution took considerably longer than the network advancements', and predominantly b) current visualization and inspection tools are still just starting to be at the edge of what can genuinely be considered as intuitive or obvious. As a consequence, there is still a lack of knowledge on the emergence of concrete interactions within the network.

In turn, this means that while samples and their respective labels might for human observers seem invariant to affine transformations, e.g. shearing, rotation, or scaling, these transformations could in fact induce semantic label errors. This might be mostly assumed irrelevant in scenarios where humans significantly outperform machines, so their sense of invariance should be reasonable to a specific point. However, especially within the medical field one prospective speculation is the improvement of diagnostic performance beyond the level of human vision. Therefore, it might be advisable to optimize the network's choice of transformations itself for maximizing the output accuracy.

Another factor is that data augmentation can be seen as inefficient with respect to training time. When invariance is encoded directly into the model, data augmentation can be kept as low as possible. For example, there have been many efforts to introduce rotational invariance into neural networks (Marcos et al., 2016; Winkels and Cohen, 2018; Cheng et al., 2016) with some of them reaching back to non-convolutional approaches (Fasel and Gatica-Perez, 2006). Also there have been approaches to systematically learn steerable filters for convolutional neural networks being inherently rotationally invariant (Weiler et al., 2017) based on the work from Jacob and Unser (2003, 2004).

While differing in their choice of concrete methods, all these approaches have in common that they either create invariance a) by some form of rotated duplication of filters, or b) by constraining the network to only allow special filter configurations which are rotationally invariant themselves, e.g. via custom loss function or network constraints. While a) unintentionally reduces the effective network capacity, b) prevents the network from detecting actually non-invariant features.

Within this paper, we demonstrate that simple transformational layers within the network for invariance might significantly improve the network's performance. We show this by introducing an inherently rotation-invariant convolutional layer, while preserving the input's local pose. We show that our network outperforms classic convolutional neural networks which share the same model complexity, i.e. number of optimizable parameters, with respect to the final test set performance. In our experiments, we show that this especially holds true when data is limited, as it is typical within clinical scenarios. As a proof of concept, we provide an exhaustive evaluation with a variety of metrics on the well-known benchmark datasets CIFAR-10, CIFAR-100 and STL-10. We furthermore demonstrate the transferability to the clinical context using two self-acquired medical datasets with the tasks of oncological treatment response classification in metastatic colorectal cancer patients (**mCRC**) and liver lesion segmentation in pancreatic cancer patients.

2. Material and methods

Invariance to visual distortions is often important in terms of generalization performance. Common distortion types within visual perception comprise translation, scale, rotation, illumination, and many more. Deep learning research often relies on the belief that data amount might compensate for variance with respect to these distortions at least to some point. Still, most top-performing neural networks apply data augmentation even when trained on large datasets. Data augmentation induces additional variance into the training data which in turn increases the probability of the classifier being *invariant* to changes represented within the augmented data with respect to an output label. The applied transformations generally rely on the a-priori knowledge of label invariance regarding specific types of transformations based on visual-perception. More formally, the transformations are input-variant but label-invariant, or in simple terms: transformations are generally reasonable with respect to the actual label, i.e. with the example of the well-known CIFAR 10 dataset within a reasonable range of transformations the perceived class of an image of the class "bird" will not change into one of the class "frog" or vice-versa.
While this is immediately comprehensible for visually perceivable classes, the situation changes when deep learning is applied to explorative tasks where the ground-truth labels can not directly be perceived by human visual inspection. Assessment beyond human perception, however, is a major goal of many approaches within the clinical domain, e.g. by Radiomics (Gillies et al., 2015). Instead, transformations might induce unnatural variance in these cases and, thus, even worsen classification performance as labels might actually be rotationally and transformationally variant. With our approach, we show that it is feasible to make the search for invariant vs. variant features a matter of the network optimization process.

2.1. The ORB approach

Our approach is built upon the idea of **ORB** - Oriented FAST and Rotational Brief from Rublee et al. (2011). ORB was one of the most recognized approaches introducing rotational invariance into actually non-invariant visual patch recognition utilizing the intensity centroids described by Rosin (1999). For this purpose Rublee et al. determine image patch

moments as:

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y) \tag{1}$$

based on normalized image intensities I at positions (x, y). Image patch moments can be used for finding the image centroid as:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \tag{2}$$

which allows rotating the patch according to the inverse of its orientation Θ:

$$\Theta = \text{atan2}(m_{01}, m_{10}) \tag{3}$$

After rotation, originally non-invariant transformations can be applied to the image while preserving rotational invariance.

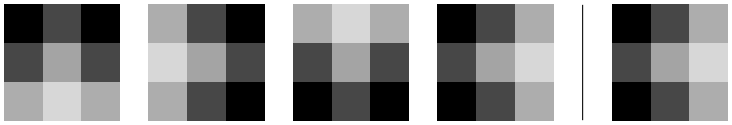

Figure 1: Left: image patches with various orientations. Within convolutional neural networks, these patches would have to be detected with respectively equally oriented filters. Right: Images after centroid rotation. With rotational invariance, all four patches would be mapped to the same orientation.

2.2. Hybrid Rotation Invariant Networks

As all aforementioned operations are differentiable, it is possible to transfer these operations to convolutional neural networks, which is reasonable as ORB was created to be applied on image patches itself. We limit this publication to only take into account the two-dimensional case, as the datasets we consider mostly consist of two-dimensional data only. However, the generalization to the n-dimensional case is straightforward. Analogously to classical 2D convolutional layers, our approach first extracts image patches parameterizable by kernel size, strides, extraction rate and border padding. In a second step, image centroids are derived according to Equation (2), and patches are rotated by their respective centroid rotation Θ from Equation (3). The data is rotated by multiplying the homogeneous extraction coordinates $(x, y, 1)$ with the matrix M:

$$(x', y', z) = (x, y, 1) \cdot M \quad \text{with} \quad M = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

When extraction coordinates (x', y') lie between pixels, a bilinear interpolation is applied. While homogeneous and non-homogeneous coordinates result in the same image positions

using the transformation matrix M, i.e. $(\frac{x'}{z}, \frac{y'}{z}) = (x', y')$, this matrix format shows that arbitrary transformations are easy to implement, making the approach feasible for invariance to other affine transformations, e.g. scale or sheering. Further approaches could include a learned transformation (see Section 4). Finally, the filter output is calculated analogously to classical convolutions with:

$$y = W * p + b \tag{5}$$

for each rotated $k \times k$ image patch p, kernel size k, weight matrix W and filter bias b.

2.2.1. Rotation invariance vs. rotational equivariance

Rotation invariance might be suboptimal, as the patches' local rotation might constrain global relations. This is for example the case with face images, where the rotation of eyes or mouth generally have implications for valid rotations of nose and eyebrows. Thus each on its own is not invariant, but equivariant (invariant with knowledge on its pose), while the patches are covariant with respect to other patches (i.e. the rotation of patches is defining for the rotation of other patches). Rotational invariance would imply an arbitrary rotation for each of the subfeatures, meaning that the actual rotation of eyes or nose would have no implications for the rotation of the mouth features, etc.. To preserve rotational equivariance (resp. the knowledge of local rotation), for every rotationally invariant layer two additional filter maps are appended, containing the values $\sin\Theta(x, y)$ and $\cos\Theta(x, y)$ for the local rotation Θ (Equation (3)) at each filter position (x, y). Still, we can not preserve actual covariance between patches as it is possible with classical convolutions where different rotations are represented by separate filters. To address this issue, we propose a combination of rotationally invariant and classic convolutions called *hybrid layers*. We expect this mixture to yield significantly better performance: while rotationally invariant input image properties should fastly be covered by the rotationally invariant filters, the non-invariant properties can be treated using standard convolutions, without the need of learning rotated duplicates of the same filters. As we use hybrid layers, i.e. half of the filters are rotationally invariant, while the other half is not, rotation variance and invariance can be preserved at every abstraction stage within the network.

2.3. Working Hypotheses and Evaluation

Based on the assumptions mentioned above, we expect our approach to especially perform well when sample size is large enough for successful formation of simple filters but not for sufficiently developing rotated versions of these filters, or for discovering spatial covariances within deeper layers. We hypothesize that a) our approach should reach similar results as with classical convolutions with very small training sets, b) significantly outperform them with medium sample sizes, and c) eventually converge to a value comparable to non-invariant convolutions. Furthermore, we expect our approach to be especially beneficial for difficult learning tasks, as its rotational invariance allows it to develop comparably simple features more efficiently. To test these assumptions, we systematically modified training set sizes S to be:

$$S_X = \{2^n \mid n_0 \leq n \leq \log_2(|X|)\} \cup \{|X|\} \tag{6}$$

for dataset X. We chose a start value of $n_0 = 7$ resulting in the smallest set having a cardinality of $\min(S_X) = 2^7 = 128$ samples since the set with the highest amount of classes

has exactly 100 classes, meaning that $\approx 78\%$ of classes are represented by only one sample within the training set.

2.3.1. Training details and sets

For training, we employed Keras (Chollet et al., 2015) using the Tensorflow backend (Abadi et al., 2015). As a proof of concept, we tested our approach on the well-known Cifar-10 and Cifar-100 datasets from Krizhevsky and Hinton (2009), as well as the STL-10 dataset from Coates et al. (2011). To show the impact for classification in typical clinical datasets, we trained our model with an extended version of the metastatic colorectal cancer dataset (**mCRC**) from Katzmann et al. (2018). To analyze whether our approach is transferable to other tasks, we also employed the aproach for liver lesion segmentation in pancreas carcinoma patients, using a newly acquired dataset. For each set size, we trained both models (i.e. rotationally invariant and classic) for three iterations with varying random seeds in each iteration. Within each iteration, seeds were fixed for both models for ensuring equal initialization and training sample order. For Cifar-10, Cifar-100 and STL-10, all results were evaluated on the respective full test set. For tumor growth prediction and liver lesion segmentation, we chose a 4-fold grouped stratified cross-validation as a tradeoff between training time and explanatory power, with 1/4 of the training set to be used for validation. In every case we optimized using categorical cross entropy loss. The best model was chosen as the one maximizing the lower bound of the 95 % bootstrapping confidence interval of ROC area-under-curve on the validation dataset, as this also takes into account metric variance and empirically provided clear advantages over using the validation loss with respect to the generalizability of the results. All results were obtained using bootstrapping until metrics convergence (Efron, 1982).

3. Results

3.1. Cifar-10, Cifar-100 and STL-10

The Cifar-10 and Cifar-100 datasets consist of 50,000 images each with a size of 32x32 from 10, respectively 100, different classes of everyday objects, e.g. planes, cars, or dogs. The test sets comprise 10,000 samples each. STL-10 is built up similarly to Cifar-10, but contains 5,000 images sized 96x96 pixels for training and 8,000 images for testing. As described above, we trained our model as well as an equivalently built one with classical convolutions while systematically varying the amount of training samples used as described in Section 2.3. The results are shown in Figure 2.

3.2. mCRC Dataset

Using an extended version of the dataset from Katzmann et al. (2018), we trained our model to predict disease progressions of patients with metastatic colorectal cancer based on baseline-followup-pairs of liver lesion slices from computed tomography images, that is, predicting growth at followup two relative to lesion size at followup one. This assesment was already shown to be partially predictive for biological information while providing benefits over the clinical standard Response Evaluation Criteria in Solid Tumors (RECIST v1.1) lesion assessment from Eisenhauer et al. (2009). Our target variable definition is defined

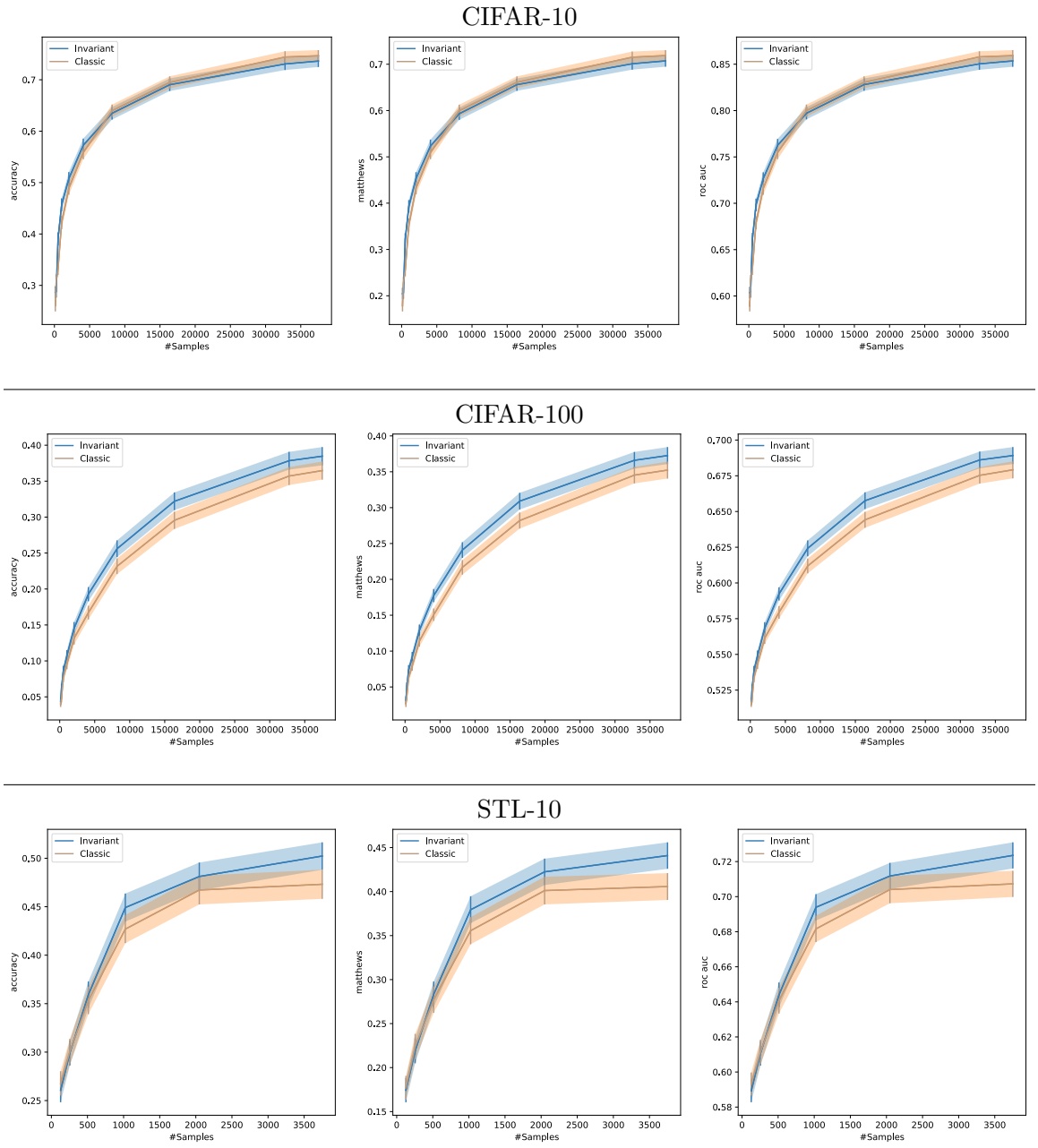

Figure 2: Results of rotationally invariant vs. variant convolutional neural networks on the CIFAR-10, CIFAR-100 and STL-10 datasets. Results are shown with respect to accuracy, matthews correlation coefficient, and area under the receiver operating curve (ROC-AUC) using multiclass-macro-averaging. The metric's 99% confidence intervals are given as shadowed areas. Expectedly, rotationally invariant layers outperform classical convolutions with medium sample sizes but become less significant with larger sets or lower-difficulty problems. With the more difficult problems from CIFAR-100 and STL-10, the differences between rotationally invariant and non-invariant layers become more significant.

as the RECIST lesion progressive disease status, thus a lesion growth of $\geq 20\%$ is assumed progression. The dataset contains 592 baseline followup pairs of 320 lesions in 138 scans from 75 patients. The test results are shown in Table 1.

	Our approach	CI 95	Classic	CI 95
Accuracy	.636	[.598, .674]	**.724**	[.689, .760]
F1	**.335**	[.273, .402]	.313	[.232, .390]
Sensitivity	**.671**	[.569, .774]	.463	[.360, .577]
Specificity	.631	[.587, .671]	**.765**	[.727, .801]
Matthews	**.211**	[.132, .291]	.177	[.089, .264]
AUC	**.717**	[.659, .776]	.639	[.561, .713]

Table 1: Results on the mCRC dataset for tumor growth prediction. Label distribution was 81 positive vs. 511 negative samples. While classifiers focused on different classes, the balanced Matthews correlation coefficient indicates superiority of rotation-invariant convolutions with significantly higher ROC AUC with $p < .05$ (two-tailed z-test).

3.3. Liver lesion segmentation in pancreatic cancer patients

We applied our approach for liver lesion segmentation using a modified 2D version of the U-Net architecture from Ronneberger et al. (2015). The model is trained on a newly acquired dataset of pancreatic cancer patients. Segmenting liver lesions in pancreatic cancer patients can be seen as particulary difficult as one patient may suffer from more than a houndred liver lesions. The dataset consisted of 3481 samples from 135 volumes of 87 pancreas carcinoma patients with fully-volumetrically segmented livers and liver lesions. Metrics were tested with and without outlier detection. When applying outlier detection, surface distances $\geq 5\,cm$ were expected to be misclassifications. The obtained results can be found in Table 2. An examplary segmentation result is shown in Figure 3.

	Our approach		**Classic**	
Metric	**w**	**w/o**	**w**	**w/o**
Mean SD	**8.36** [8.16, 8.60]	**26.6** [25.0, 28.6]	8.42 [8.16, 8.68]	27.6 [25.8, 29.6]
Median SD	2.98 [2.82, 4.00]	**4.70** [4.48, 5.66]	**2.82** [2.82, 4.00]	4.76 [4.48, 5.66]
Dice	**.552** [.536, .566]		.542 [.526, .559]	

Table 2: Results for the pancreatic cancer dataset with hybrid (left) vs. classical convolutions (right). Surface metrics (in mm) were calculated with (w) and without (w/o) outlier removal. When applying outlier removal, both models performed similarly, while without outlier removal our model outperformed classical convolutions with respect to mean and median surface distance (SD).

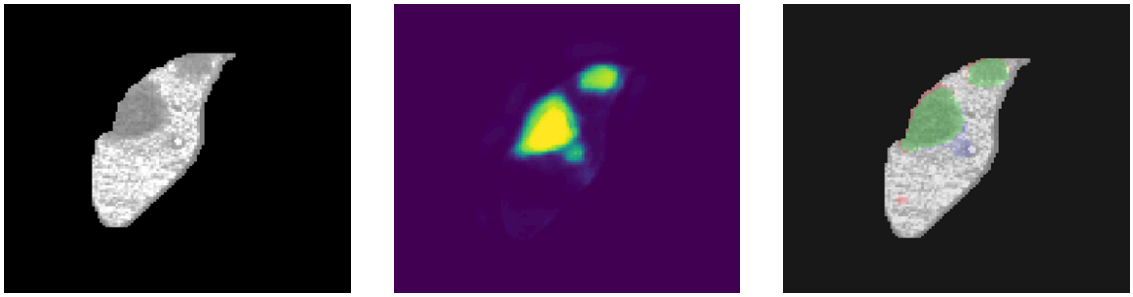

Figure 3: Examplary segmentation result using rotationally invariant filters. Left: original image; middle: segmentation result; right: overlay. True positives, false positives and false negatives are respectively marked as green, blue, and red.

4. Discussion

As shown in Section 3, our approach mostly outperformed classical convolutions for all tested datasets. As expected, this especially holds true when working with medium training set sizes and/or difficult problems, as shown on the example of Cifar-100 and STL-10 with the resulting differences being highly significant for all tested metrics ($p \leq .01$). However, more simple problems like Cifar-10 do not neccessarely benefit from the approach when large training sets are available. On medical data, our approach performed superior to classical convolutions, both, for classification ($p_{AUC} \leq .05$) as well as for segmentation tasks. Though there generally is some understanding on how rotational invariance can be realized by rotationally non-invariant networks (Sauder, 2006), the introduction of inherently rotationally invariant networks might provide major benefits for various training problems especially within the medical domain, where training data is rare and models can easily suffer from overfitting. While deep neural networks are thought to develop even rotationally invariant features on their own, their capabilities are limited due to training data amount, training time and model complexity. Especially these issues major arguments for directly encoding invariance into the model itself.

In future work, the proposed approach should be expanded to explicitly learn invariance when needed, as the choice of using invariant vs. non-invariant features within this publication was realized by statically mixing both layer types only. Extending the given approach, the general form of the transformational matrices like shown in Section 2.2 could be used for making models inherently invariant to other affine transformations, e.g. scale or sheering.

Acknowledgments

This work has received funding from the German Federal Ministry of Education and Research as part of the PANTHER project under grant agreement no. 13GW0163A. The concepts and information presented in this article are based on research and are not commercially available.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Gong Cheng, Junwei Han, Peicheng Zhou, and Dong Xu. Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection. *IEEE Transactions on Image Processing*, 28(1):265–278, 2016.

François Chollet et al. Keras. https://keras.io, 2015.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

Bradley Efron. *The jackknife, the bootstrap, and other resampling plans*, volume 38. Siam, 1982.

Elizabeth A Eisenhauer, Patrick Therasse, Jan Bogaerts, Lawrence H Schwartz, D Sargent, Robert Ford, Janet Dancey, S Arbuck, Steve Gwyther, Margaret Mooney, et al. New response evaluation criteria in solid tumours: revised recist guideline (version 1.1). *European journal of cancer*, 45(2):228–247, 2009.

Beat Fasel and Daniel Gatica-Perez. Rotation-invariant neoperceptron. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 336–339. IEEE, 2006.

Robert J Gillies, Paul E Kinahan, and Hedvig Hricak. Radiomics: images are more than pictures, they are data. *Radiology*, 278(2):563–577, 2015.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Mathews Jacob and Michael Unser. Optimal steerable filters for feature detection. In *ICIP (3)*, pages 433–436, 2003.

Mathews Jacob and Michael Unser. Design of steerable filters for feature detection using canny-like criteria. *IEEE transactions on pattern analysis and machine intelligence*, 26 (8):1007–1019, 2004.

Alexander Katzmann, Alexander Muehlberg, Michael Suehling, Dominik Noerenberg, Julian Walter Holch, Volker Heinemann, and Horst-Michael Gross. Predicting lesion growth and patient survival in colorectal cancer patients using deep neural networks. 2018.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

Diego Marcos, Michele Volpi, and Devis Tuia. Learning rotation invariant convolutional filters for texture classification. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 2012–2017. IEEE, 2016.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

Paul L Rosin. Measuring corner properties. *Computer Vision and Image Understanding*, 73(2):291–307, 1999.

Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.

Nathaniel Sauder. Encoded invariance in convolutional neural networks. *University of Chicago*, pages 2–6, 2006.

Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221–248, 2017.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

Maurice Weiler, Fred A Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant cnns. *arXiv preprint arXiv:1711.07289*, 2017.

Marysia Winkels and Taco S Cohen. 3d g-cnns for pulmonary nodule detection. *arXiv preprint arXiv:1804.04656*, 2018.

Appendix A. Detailed dataset description

A.1. Tumor growth dataset

The classification goal was defined analogously to the RECIST guideline (Eisenhauer et al., 2009) as a lesion having $\geq 20\%$ growth at the next followup after the input timepoints. Each lesion was masked using fully-volumetric segmentations created by radiologists. All lesions were isotropically rescaled with a target voxel size of $1mm \times 1mm \times 1mm$ using bicubic interpolation. For all lesions a lesion-centered window of $64mm \times 64mm \times 64mm$ due to lesion diameter quantiles of $(\varnothing_{Pr(\varnothing)\leq.1}, \varnothing_{Pr(\varnothing)\geq.9}) = (10.5\,\text{mm}, 55.2\,\text{mm})$. For each lesion, only the middle slice was classified. In total the dataset contained 592 valid samples, with 81 being positive (growth) vs. 511 being negative (non-growth). All samples underwent histogram equalization. Examplary data is shown in Figure 4.

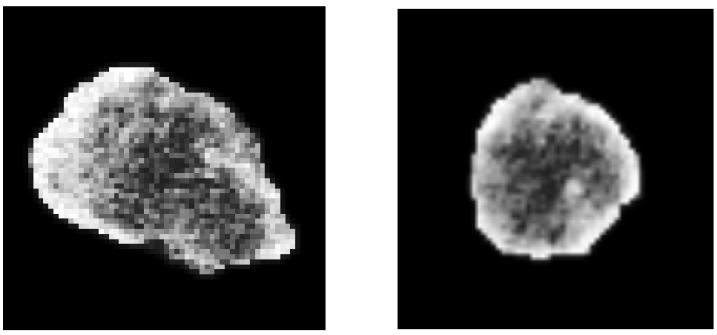

Figure 4: Example images of the CRC dataset after histogram equalization.

A.2. Pancreas dataset

The complete dataset consisted of 135 volumes from 87 patients. Classification was done in 2D using single slices of masked livers. Isotropic resampling to $1mm \times 1mm \times 1mm$ voxel size was applied using bicubic interpolation. All slices had a size of 512×512 voxels. Although only venous phases were used, data heterogeneity was high due to various treatments, kernels, scanners and comorbidities, such as fatty liver and cholestasis. All samples were resized to 256×256 pixels. All samples having at least one voxel of lesion tissue were involved in the dataset, resulting in a total quantity of 3536 valid liver slices. Again, all samples underwent histogram equalization. An example image is shown in Figure 3.

Appendix B. Architectures

Following we list the architectures used to calculate the results from Section 3. The networks are given as convolutional networks. The respective hybrid networks are created by simply replacing the convolutional layers with hybrid layers as described in Section 2.2.

B.1. Cifar-10, Cifar-100, STL-10

The architectures used for the classification results from Section 3.1 on CIFAR-10, CIFAR-100 and STL-10 are shown in Table 3.

Description	CIFAR-10/-100	STL-10	filter size
input	(32,32,1)	(96,96,1)	-
conv + BN + leaky ReLU	(32,32,32)	(96,96,32)	(3,3)
max pooling	(16,16,32)	(48,48,32)	
conv + BN + leaky ReLU	(16,16,48)	(48,48,48)	(3,3)
max pooling	(8,8,48)	(24,24,48)	-
conv + BN + leaky ReLU	(8,8,64)	(24,24,64)	(3,3)
conv + BN + leaky ReLU	(8,8,96)	(24,24,96)	(3,3)
flatten	(6144,)	(55296,)	-
dense + BN + leaky ReLU	(512,)	(512,)	-
softmax	(10,)/(100,)	(10,)	-

Table 3: For CIFAR-10, CIFAR-100 and STL-10, a very simple network architecture was chosen. It is based on a sequence of blocks of convolutional layers (conv), batch normalization (BN), leaky ReLU activation and max pooling, followed by one fully-connected as well as one softmax-output layer and is inspired by the keras example network for the CIFAR-10 dataset.

B.2. Tumor growth

Tumor growth (Section 3.2) was classified using a two-laned convolutional neural network built analogously to the model from Katzmann et al. (2018) with baseline and followup being processed in one lane each. The network weights were initialized using autoencoder pretraining. The complete architecture is shown in Table 4.

B.3. Liver lesion segmentation

The architecture for liver lesion segmentation (Section 3.3) is based on the U-Net approach from Ronneberger et al. (2015) with minor modifications. The full architecture is shown in Table 5.

	Description	**shape**	**filter size**
	input	(2,64,64,1)	-
2×	conv + BN + leaky ReLU	(64,64,8)	(5,5)
	max pooling	(32,32,8)	-
	conv + BN + leaky ReLU	(32,32,16)	(5,5)
	max pooling	(16,16,16)	-
	conv + BN + leaky ReLU	(16,16,24)	(5,5)
	max pooling	(8,8,24)	-
	conv + BN + leaky ReLU	(8,8,32)	(5,5)
	max pooling	(4,4,32)	-
	conv + BN + leaky ReLU	(4,4,40)	(5,5)
	flatten	(640,)	-
	dense + BN + leaky ReLU	(8,)	-
	dense + BN + leaky ReLU	(4,)	-
	dense + BN + leaky ReLU	(4,)	-
	softmax	(2,)	-

Table 4: Classifier architecture for tumor growth classification consisting of blocks of convolutional layers (conv), batch normalization (BN), leaky ReLU activation and max pooling layers, followed by two fully-connected as well as a softmax-output layer.

	Description	shape	filter size
	input	(256,256,1)	
1	conv + BN + leaky ReLU	(256,256,4)	(3,3)
	max pooling	(128,128,4)	
2	conv + BN + leaky ReLU	(128,128,16)	(3,3)
	max pooling	(64,64,16)	
3	conv + BN + leaky ReLU	(64,64,32)	(3,3)
	max pooling	(32,32,32)	
	conv + BN + leaky ReLU	(32,32,64)	(3,3)
4	upsampling	(64,64,64)	
	concat (3,4)	(64,64,96)	
	conv + BN + leaky ReLU	(64,64,32)	(3,3)
5	upsampling	(128,128,32)	
	concat (2,5)	(128,128,48)	
	conv + BN + leaky ReLU	(128,128,16)	(3,3)
6	upsampling	(256,256,16)	
	concat(1,6)	(256,256,20)	
	conv + BN + sig.	(256,256,1)	(3,3)

Table 5: Segmentation network architecture used for liver lesion segmentation based on U-Net (Ronneberger et al., 2015). As in U-Net, the network convolves and downsamples the image with blocks of convolutional layers (conv), batch normalization (BN), and leaky ReLU activation. Afterwards the image is upsampled again and convolved with the information of the horizontal shortcuts (concat). The final output is generated using a sigmoid function (sig.).