Structured Receptive Fields in CNNs

Jacobsen, J.-H.; van Gemert, J.; Lou, Z.; Smeulders, A.W.M.

DOI
10.1109/CVPR.2016.286

Publication date
2016

Document Version
Author accepted manuscript

Published in
Proceedings 29th IEEE Conference on Computer Vision and Pattern Recognition : CVPR 2016

Structured Receptive Fields in CNNs

Jörn-Henrik Jacobsen[1], Jan van Gemert[1,2], Zhongyou Lou[1], Arnold W. M. Smeulders[1]
[1]University of Amsterdam, The Netherlands
[2]TU Delft, The Netherlands
{j.jacobsen,z.lou,a.w.m.smeulders}@uva.nl, j.c.vangemert@tudelft.nl

Abstract

Learning powerful feature representations with CNNs is hard when training data are limited. Pre-training is one way to overcome this, but it requires large datasets sufficiently similar to the target domain. Another option is to design priors into the model, which can range from tuned hyperparameters to fully engineered representations like Scattering Networks. We combine these ideas into structured receptive field networks, a model which has a fixed filter basis and yet retains the flexibility of CNNs. This flexibility is achieved by expressing receptive fields in CNNs as a weighted sum over a fixed basis which is similar in spirit to Scattering Networks. The key difference is that we learn arbitrary effective filter sets from the basis rather than modeling the filters. This approach explicitly connects classical multiscale image analysis with general CNNs. With structured receptive field networks, we improve considerably over unstructured CNNs for small and medium dataset scenarios as well as over Scattering for large datasets. We validate our findings on ILSVRC2012, Cifar-10, Cifar-100 and MNIST. As a realistic small dataset example, we show state-of-the-art classification results on popular 3D MRI brain-disease datasets where pre-training is difficult due to a lack of large public datasets in a similar domain.

1. Introduction

Where convolutional networks have appeared enormously powerful in the classification of images when ample data are available [14], we focus on smaller image datasets. We propose structuring receptive fields in CNNs as linear combinations of basis functions to train them with fewer image data.

The common approach to smaller datasets is to perform pre-training on a large dataset, usually ImageNet [29]. Where CNNs generalize well to domains similar to the domain where the pre-training came from [27, 40], the performance decreases significantly when moving away from the pre-training domain [40, 37]. We aim to make learning more effective for smaller sets by restricting CNNs parameter spaces. Since *all images* are spatially coherent and human observers are considered to only cast local variations up to a certain order as meaningful [11, 18] our key assumption is that it is unnecessary to learn these properties in the network. When visualizing the intermediate layers of a trained network, see e.g. [39] and Figure 2, it becomes evident that the filters as learned in a CNN are locally coherent and as a consequence can be decomposed into a smooth compact filter basis [12].

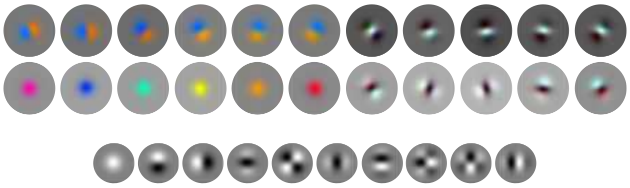

Figure 1: A subset of filters of the first structured receptive field CNN layer as trained on 100-class ILSVRC2012 and the Gaussian derivative basis they are learned from. The network learns scaled and rotated versions of zero, first, second and third order filters. Furthermore, the filters learn to recombine the different input color channels which is a crucial property of CNNs.

We aim to maintain the CNN's capacity to learn general variances and invariances in arbitrary images. Following from our assumptions, the demand is posed on the filter set that i) a linear combination of a finite basis set is capable of forming any arbitrary filter necessary for the task at hand, as illustrated in Figure 1 and ii) that we preserve the full learning capacity of the network. For i) we choose the family of Gaussian filters and its smooth derivatives for which it has been proven [12] that 3-rd or 4-th order is sufficient to capture all local image variation perceivable by humans. According to scale-space theory [11, 35], the Gaussian family constitutes the Taylor expansion of the image function which guarantees completeness. For ii) we maintain backpropagation parameter optimization in the network, now applied to learning the weights by which the filters are summed into the effective filter set.

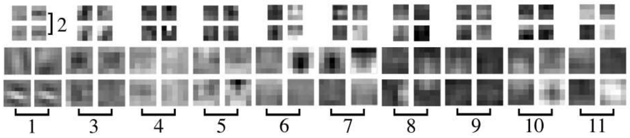

Figure 2: Filters randomly sampled from all layers of the GoogLenet model [33], from left to right layer number increases. Without being forced to do so, the model exhibits spatial coherence (seen as smooth functions almost everywhere) after being trained on ILSVRC2012. This behaviour reflects the spatial coherence of the input feature maps even in the highest layers.

Similarly motivated, the Scattering Transform [2, 20, 30], a special type of CNN, uses a complete set of wavelet filters ordered in a cascade. However, different from a classical CNN, the filters parameters are not learned by backpropagation but rather they are fixed from the start and the whole network structure is motivated by signal processing principles. In the Scattering Network the choice of local and global invariances are tailored to the type of images specifically. In the Scattering Transform invariance to group actions beyond local translation and deformation requires explicit design [20] with the regards to the variability encountered in the target domain such as translation [2], rotation [30] or scale. As a consequence, when the desired invariance groups are known a priori, Scattering delivers very effective networks.

Our paper takes the best of two worlds. On the one hand, we adopt the Scattering principle of using fixed filter bases as a function prior in the network. But on the other hand, we maintain from plain CNNs the capacity to learn arbitrary effective filter combinations to form complex invariances and equivariances.

Our main contributions are:

- Deriving the structured receptive field network (RFNN) from first principles by formulating filter learning as a linear decomposition onto a filter basis, unifying CNNs and multiscale image analysis in a learnable model.
- Combining the strengths of Scattering and CNNs. We do well on both domains: i) small datasets where Scattering is best but CNNs are weak; ii) complex datasets where CNNs excel but Scattering is weak.
- State-of-the-art classification results on a small dataset where pre-training is infeasible. The task is Alzheimer's disease classification on two widely used brain MRI datasets. We outperform all published results on the ADNI dataset.

2. Related Work

2.1. Scale-space: the deep structure of images

Scale-space theory [35] provides a model for the structure of images by steadily convolving the image with filters of increasing scale, effectively reducing the resolution in each scale step. While details of the image will slowly disappear, the order by which they do so will uniquely encode the deep structure of the image [11]. Gaussian filters have the advantage in that they do not introduce any artifacts [18] in the image while Gaussian derivative filters form a complete and stable basis to decompose locally any realistic image. The set of responses to the derivative filters describing one patch is called the N-jet [5].

In the same vein, CNNs can be perceived to also model the deep structure of images, this time in a non-linear fashion. The pooling layers in a CNN effectively reduce resolution of input feature maps. Viewed from the top of the network down, the spatial extent of a convolution kernel is increased in each layer by a factor 2, where a 5x5 kernel at the higher layer measures 10x10 pixels on the layer below. The deep structure in a CNN models the image on several discrete levels of resolution simultaneously, precisely in line with Scale-space theory.

Where CNNs typically reduce resolution by max pooling in a non-linear fashion, Scale-space offers a linear theory for continuous reduction of resolution. Scale-space theory treats an image as a function of the mathematical apparatus to reveal the local image structure. In this paper, we exploit the descriptive power of Scale-space theory to decompose the image locally on a fixed filter basis of multiple scales.

2.2. CNNs and their parameters

CNNs [15] have large numbers of parameters to learn [13]. This is their strength as they can solve extremely complicated problems [13, 34]. At the same time, their number of unrestricted parameters is a limiting factor in terms of the large amounts of data needed to train. To prevent overfitting, which is an issue even when training on large datasets like the million images of the ILSVRC2012 challenge [29], usually regularization is imposed with methods like dropout [32] and weight decay [22]. Regularization is essential to achieving good performance. In cases where limited training data are available, CNN training quickly overfits regardless and the learned representations do not generalize well. Transfer learning from models pre-trained in similar domains to the new domain is necessary to achieve competitive results [23]. One thing pre-training on large datasets provides is knowledge about properties inherent to all natural images, such as spatial coherence and robustness to uninformative variability. In this paper, we aim to design these properties into CNNs to improve generalization when limited training data are available.

2.3. The Scattering representation

To reduce model complexity we draw inspiration from the elegant convolutional Scattering Network [2, 20, 30]. Scattering uses a multi-layer cascade of a pre-defined wavelet filter bank with nonlinearity and pooling operators. It computes a locally translation-invariant image representation, stable to deformations while avoiding information loss by recovering wavelet coefficients in successive layers. No learning is used in the image representation: all relevant combinations of the filters are fed into an SVM-classifier yielding state-of-the-art results on small dataset classification. Scattering is particularly well-suited to small datasets because it refrains from feature learning. Since all filter combinations are pre-defined, their effectiveness is independent of dataset size. In this paper, we also benefit from a fixed filter bank. In contrast to Scattering, we *learn* linear combinations of a filter basis into effective filters and non-linear combinations thereof.

The wavelet filterbank of Scattering is carefully designed to sample a range of rotations and scales. These filters and their properties are grounded in wavelet theory [19] and exhibit precisely formulated properties. By using interpretable filters, Scattering can design invariance to finite groups such as translation [2], scale and rotation [30]. Hard coding the invariance into the network is effective when the problem and its invariants are known precisely, but for many applications this is rarely the case. When the variability is unknown, additional Scattering paths have to be computed, stored and processed exhaustively before classification. This leads to a well-structured but very high dimensional parameter space. In this paper, we use a Gaussian derivatives basis as the filter bank, firmly grounded in scale-space theory [11, 18, 35]. Our approach incorporates learning effective filter combinations from the very beginning, which allows for a compact representation of the problem at hand.

2.4. Recent CNNs

Restriction of parameter spaces has led to some major advances in recent CNNs performance. Network in Network [17] and GoogleNet [33] illustrate that fully connected layers, which constitute most of Alexnet's parameters, can be replaced by a global average pooling layer reducing the number of parameters in the fully connected layers to virtually zero. The number of parameters in the convolution layers is increased to enhance the expressiveness of each layers features. Overall the total number of parameters is not necessarily decreased, but the function space is restricted, allowing for bigger models while classification accuracy improves [17, 33].

The VGG Network [31] improves over Alexnet in a different way. The convolution layers parameter spaces are restricted by splitting each 5x5 convolution layer into two 3x3 convolution layers. 5x5 convolutions and 2 subsequent 3x3 convolutions have the same effective receptive field size while each receptive field has 18 instead of 25 trainable parameters. This regularization enables learning larger models that are less prone to overfitting. In this paper, we follow a different approach in restricting the free parameter space without reducing filter size.

3. Deep Receptive Field Networks

3.1. Structured receptive fields

In our structured receptive field networks we make the relationship between Scale-space and CNNs explicit. Whereas normal CNNs treat images and their filters as pixel values, we aim for a CNN that treats images as functions in Scale-space. Thus, the learned convolution kernels become functions as well. We therefore approximate an arbitrary CNN filter $F(x)$ with a Taylor expansion around a up to order M

$$F(x) = \sum_{m=0}^{M} \frac{F^m(a)}{m!}(x-a)^m. \tag{1}$$

Scale-space allows us to use differential operators on images, due to linearity of convolution we are able to compute the exact derivatives of the scaled underlying function by convolution with derivatives of the Gaussian kernel

$$G(.;\sigma) * F(x) = \sum_{m=0}^{N} \frac{(G^m(.;\sigma) * F)(a)}{m!}(x-a)^m, \tag{2}$$

where $*$ denotes convolution, $G(.;\sigma)$ is a Gaussian kernel with scale σ and $G^m(.;\sigma)$ is the m^{th} order Gaussian derivative with respect to it's spatial variable. Thus, a convolution with a basis of weighted Gaussian derivatives receptive fields is the functional equivalent to pixel values in a standard CNN operating on a scaled infinitely differentiable version of the image.

To construct the full basis set in practice, one can show that the Hermite polynomials emerge from a sequence of Gaussian derivatives up to order M [28]. A Gaussian derivative of arbitrary order can be obtained from the orthogonal Hermite polynomials H_m through pointwise multiplication with a Gaussian envelope

$$G^m(.;\sigma) = (-1)^m \frac{1}{\sqrt{\sigma}^m} H_m(\frac{x}{\sigma\sqrt{2}}) \circ G(x;\sigma). \tag{3}$$

The resulting operators allow computation of an image's local geometry at scale σ and location x up to any order of precision M. This basis is thus a complete set. Each derivative corresponds to an independent degree of freedom, making it also a minimal set.

Thus, an RFNN is a general CNN when a complete polynomial up to infinite order is considered. We restrict the basis based on the requirement that one can construct quadrature pair filters as suggested by Scattering and by evidence from Scale-space theory [12] that considers all orders up to a maximum of 4, as it has been suggested that orders beyond that does not carry any information meaningful to visual perception.

3.2. Transformation properties of the basis

The isotropic Gaussian derivatives exhibit multiple desirable properties. It is possible to create complex multi-orientation pyramids that constitute wavelet representations similar to the Morlet Wavelet pyramids used in Scattering Networks [2]. A complex multiresolution filterbank can be constructed from a dilated and rotated Gaussian derivative quadrature. The exact dilated versions of an arbitrary Gaussian derivative G^m can be obtained through convolution with a Gaussian kernel of the desired scale increase $\sigma = n$

$$G^m(.;j+n) = G^m(.;j) * G(.;n). \tag{4}$$

Arbitrary rotations of Gaussian derivative kernels can be obtained from a minimal set of basis filters without the need to rotate the basis itself. This property is referred to as steerability [6]. Steerability is a property of all functions that can be expressed in a polynomial in x and y times an isotropic Gaussian. This certainly holds for the Gaussian derivatives according to equation 3. For example a quadrature pair of 2nd and 3rd order Gaussian derivatives G^{xx} and G^{xxx} rotated by an angle θ can be obtained from a minimal 3 and 4 x-y separable basis set given by

$$\begin{aligned} G^{xx}_{\theta} &= \cos^2(\theta)G^{xx} - 2\cos(\theta)\sin(\theta)G^{xy} + \sin^2(\theta)G^{yy} \\ G^{xxx}_{\theta} &= \cos^3(\theta)G^{xxx} - 3\cos^2(\theta)\sin(\theta)G^{xxy} \\ &\quad +3\cos(\theta)\sin^2(\theta)G^{xyy} - \sin^3(\theta)G^{yyy} \end{aligned} \tag{5}$$

A general derivation of the minimal basis set necessary for steering arbitrary orders can be found in [6]. Note that the anisotropic case can be constructed in analogous manner according to [25]. This renders Scattering as a special case of the RFNN for fixed angles and scales, given a proper choice of pooling operations and possibly skip connections to closely resemble the architecture described in [2]. In practice this allows for seamless integration of the Scattering concept into CNNs to achieve a variety of hybrid architectures.

Algorithm 1 RFNN Learning - updating the parameters α^l_{ij} between input map indexed by i and output map indexed by j of layer l in the Mini-batch Gradient Decent framework.

```
Input: input feature maps $o^{l-1}_i$ for each training sample (computed for the previous layer, $o^{l-1}$ is the input image when $l = 1$), corresponding ground-truth labels $\{y_1, y_2, \ldots, y_K\}$, the basic kernels $\{\phi_1, \phi_2, \ldots, \phi_M\}$, previous parameter $\alpha^l_{ij}$.
compute the convolution $\{\zeta_1, \zeta_2, \ldots, \zeta_m\}$ of $\{{o^{l-1}}_i\}$ respect to the basic kernels $\{\phi_1, \phi_2, \ldots, \phi_M\}$
obtain the output map $o^l_j = \alpha^l_{ij1} \cdot \zeta_1 + \alpha^l_{ij2} \cdot \zeta_2 + \ldots + \alpha^l_{ijM} \cdot \zeta_M$
compute the $\delta^l_{jn}$ for each output neuron $n$ of the output map $o^l_j$
compute the derivative $\psi'(t^l_{jn})$ of the activation function
compute the gradient $\frac{\partial E}{\partial \alpha^l_{ij}}$ respect to the weights $\alpha^l_{ij}$
update parameter $\alpha^l_{ij} = \alpha^l_{ij} - r \cdot \frac{1}{K} \cdot \sum_{k=1}^{K} [\frac{\partial E}{\partial \alpha^l_{ij}}]_k$, $r$ is the learning rate
Output: $\alpha^l_{ij}$, the output feature maps $o^l_j$
```

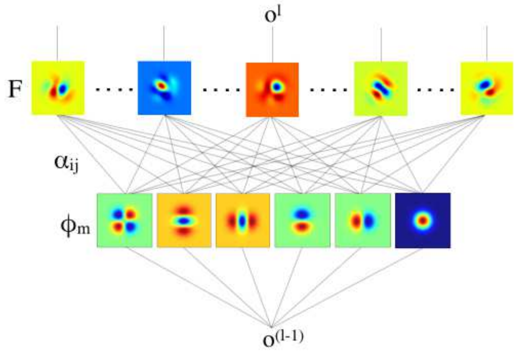

Figure 3: An illustration of the basic building block in an RFNN network. A linear comibination of a limited basis filter set ϕ_m yields an arbitrary number of effective filters. The weights α_{ij} are learned by the network.

3.3. Learning basis filter parameters

Learning a feature representation boils down to convolution kernel learning. Where a classical CNN learns pixel values of the convolutional kernel, a RFNN learns Gaussian derivative basis function weights that combine to a convolution kernel function. A 2D filter kernel function $F(x, y)$ in all layers, is a linear combination of i unique (non-symmetric) Gaussian derivative basis functions ϕ

$$F(x, y) = \alpha_1\phi_1 + \cdots + \alpha_n\phi_i, \tag{6}$$

where $\alpha_1, \ldots, \alpha_i$ are the parameters being learned.

We learn the filter's weights α by mini-batch stochastic gradient descent and compute the derivatives of the loss function E with respect to the parameters α through backpropagation. It is straightforward to show the independence between the basis weights α and the actual basis (see Appendix for derivation). Thus, we formulate the basis learning as a combination of a fixed basis layer with a 1x1 convolution layer that has a kernel depth equal to the basis order. Propagation through the 1x1 layer is done as in any CNN while propagation through the basis layer is achieved by a convolution with flipped versions of the Gaussian filters. This makes it straightforward to include into any existing deep learning framework. The basic structured receptive field building block is illustrated in figure 3, showing how each effective filter is composed out of multiple basis filters. Note that the linearity of convolution allows us to never actually compute the effective filters. Convolving with effective filters is the same as convolving with the basis and then recombining the feature maps, allowing for efficient implementation. Algorithm 1 shows how the parameters are updated.

3.4. The network

In this work, we choose the Network in Network (NiN) architecture [17] as the basis into which we integrate the structured receptive fields. It is particularly suited for an analysis of the RFNN approach, as the absence of a fully connected layer ensures all parameters to be fully concerned with re-combining basis filter outcomes of the current layer. At the same time, it is powerful, similar in spirit to the state of the art Googlenet [33], while being comparably small and fast to train.

NiN alternates one spatial convolution layer with 1x1 convolutions and pooling. The 1x1 layers form non-linear combinations of the spatial convolution layers outputs. This procedure is repeated four times in 16 layers, with different number of filters and kernel sizes for the spatial convolution layer. The final pooling layer is a global average pooling layer. Each convolution layer is followed by a rectifier nonlinearity. Details on the different NiNs for Cifar and Imagenet can be found in the Caffe model zoo [9].

In the RFNN version of the Network in Network model, the basis layer including the Gaussian derivatives set is replacing the spatial convolution layer and corresponds to ϕ_m in equation 6. Thus, each basis convolution layer has a number of filters depending on order and scale of the chosen basis set. The basis set is fixed: no parameters are learned in this layer. The linear re-combination of the filter basis is done by the subsequent 1x1 convolution layer, corresponding to α_{ij} in equation 6. Note that there is no non-linearity between ϕ_m and α_{ij} layer in the RFNN case, as the combinations of the filters are linear. Thus the RFNN model is almost identical to the standard Network in Network.
We evaluate the model with and without multiple scales σ_s. When including scale, we extract 4 scales, as the original model includes 3 pooling steps and thus operates on 4 scales at least. In the first layer we directly compute 4 scales, sampled continuously with $\sigma_s = 2^s$ where $s = scale$ as done in [2]. In each subsequent layer we discard the lowest scale. The dimensionality reduction by max pooling renders it meaningless to insert the lowest scale of the previous layer into the filter basis set as it is already covered by the pyramidal structure of the network. This enables us to save on basis filters in the higher layers of the network. In conclusion we reduce the total number of 2D filters in the network from 520,000 in the standard Network in Network to between 12 and 144 in the RF Network in Network (RFNiN), while retaining the models expressiveness as shown in the experimental section.

4. Experiments

The experiments are partitioned into four parts. i) We show insight in the proposed model to investigate design choices; ii) we show that our model combines the strengths of Scattering and CNNs; iii) we show structured receptive fields improve classification performance when limiting training data; iv) we show a 3D version of our model that outperforms the state-of-the-art, including a 3D-CNN, on two brain MRI classification datasets where large pretraining datasets are not available. We use the Caffe library [9] and Theano [1] where we added RFNN as a separate module. Code is available on github[1].

4.1. Experiment 1: Model insight

The RFNN used in this section is the structured receptive field version of the Network in Network (RFNiN) introduced in section 3.3. We gain insight into the model by evaluating the scale and order of the basis filters. In addition, we analyze the performance compared to the standard Network in Network (NiN) [17] and Alexnet [13] and show that our proposed model is not merely a change in architecture. To allow overnight experiments we use the 100 largest classes of the ILSVRC2012 ImageNet classification challenge [29]. Selection is done by folder size, as more than 100 classes have 1,300 images in them, yielding a dataset size of 130,000 images. This is a real-world medium sized dataset in a domain where CNNs excel.

Experimental setup. The Network in Network (NiN) model and our Structured Receptive Field Network in Network (RFNiN) model are based on the training definitions provided by the Caffe model zoo [9]. Training is done with the standard procedure on Imagenet. We use stochastic gradient descent, a momentum of 0.9, a weight decay of 0.0005. The images are re-sized to 256x256, mirrored during training and the dataset mean is subtracted. The base learning rate was decreased by a factor of 10, according to the reduction from 1,000 to 100 classes, to ensure proper scaling of the weight updates, NiN didn't converge with the original learning rate. We decreased it by a factor of 10 after 50,000 iterations and again by the

[1] https://github.com/jhjacobsen/RFNN

ILSVRC2012-100 Subset			
Method	Top-1	2DFilters	#Params
RFNiN 1^{st}-order	44.83%	12	1.8M
RFNiN 2^{nd}-order	61.24%	24	3.4M
RFNiN 3^{rd}-order	63.64%	40	5.5M
RFNiN 4^{th}-order	62.92%	60	8.1M
RFNiN-Scale 1^{st}-order	57.21%	24	2.2M
RFNiN-Scale 2^{nd}-order	67.56%	54	4.2M
RFNiN-Scale 3^{rd}-order	**69.65%**	94	6.8M
RFNiN-Scale 4^{th}-order	68.95%	144	10.1M
Network in Network	67.30%	$520k$	8.2M
Alexnet	54.86%	$370k$	60.0M

Table 1: Results on 100 Biggest ILSVRC2012 classes: The table shows RFNiN with 1st, 2nd, 3rd and 4th order filters in the whole network. Row 1-4 are applying basis filters in all layers on a scale of σ=1. RFNiN-scale in row 5-8 applies basis filters on 4 scales, where σ=1,2,4,8. The results show that a 3^{rd} order basis is sufficient while incorporating scale into the network gives a big gain in performance. The RFNiN is able to outperform the same Network in Network architecture.

same factor after 75,000 iterations. The networks were trained for 100,000 iterations. Results are computed as the mean Top-1 classification accuracy on the validation set.

Filter basis order. In table 1, the first four rows show the result of RFNiN architectures with 1st to 4th order Gaussian derivative basis filter set comprised of 12 to 60 individual Gaussian derivative filters in all layers of the network. In these experiments the value of σ=1, fixed for all filters and all layers. Comparing first to fourth order filter basis in table 1, we conclude that third order is sufficient, outperforming first and second order as predicted by Scale-space theory [12]. The fourth order does not add any more gain.

Filter scale. The RFNiN-Scale entries of table 1 show the classification result up to fourth order now with 4 different scales, σ=1, 2, 4, 8 for the lowest layer, σ=1, 2, 4 for the second layer, σ=1, 2 for the third, and σ=1 for the fourth. This implies that the basis filter set expands from 24 up to 144 filters in total in the network. Comparing the use of single scale filters in the network to dilated copies of the filters with varying scale indicates that a considerable gain can be achieved by including filters with different scales. This observation is supported by Scattering [2], showing that the multiple scales can directly be extracted from the first layer on. In fact, normal CNNs are also capable of similar behavior, as positive valued low-pass filter feature maps are not affected by rectifier nonlinearities [30]. Thus, scale can directly be computed from the first layer onwards, which yields a much smaller set of basis filters and fewer convolutions needed in the higher layers. Note that number of parameters is not directly correlated with performance.

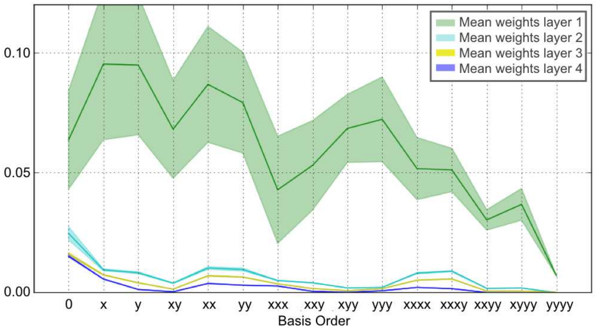

Figure 4: Mean of filter weights and variances per layer for 15 basis filters with no scale, as trained on ILSVRC2012-100 subset. Note that the lower order filters have the highest weights while zero-order filters are most effective in higher layers for combinations of lower responses.

Analysis of network layers. For the network RFNiN 4th-order Figure 4 provides an overview of the range of basis weights per effective filters in all layers, where the x-axis indexes the spatial derivative index and y-axis the mean value plus standard deviation of weights per layer over all effective filter kernels. The figure indicates that weights decrease towards higher orders as expected. Furthermore zero order filters have relatively high weights in higher layers, which hints to passing on scaled incoming features.

Comparison to Network in Network. The champion RFNiN in table 1 slightly outperforms the Network in Network with the same setting and training circumstances while only having 94 instead 520,000 spatial filters in the network in total. Note that the number of parameters is relatively similar though, as the scale component increases the number of basis functions per filter significantly. The result shows that our basis representation is sufficient for complex tasks like Imagenet.

Refactorize Network in Network. To illustrate that our proposed model is not merely a change in architecture we compare to a third architecture. We remove the Gaussian basis and we re-factorize the NiN such that it becomes identical to RFNiN. Both have almost the same number of parameters, but the NiN-factorize has a freely learnable basis. Re-factorizing only the first layer and leaving the rest of the network as in the original NiN, in table 2 we show that a Gaussian basis is superior to a learned basis. When re-factorizing all layers, RFNiN-Scale 3^{rd}-order results are superior by far to the identical NiN-factorize All Layers.

Model	Basis	#Params	Top-1
NiN-refactor Layer 1	Free	7.47M	64.10%
RFNiN-refactor Layer 1	Gauss	7.47M	68.63%
NiN-refactor All Layers	Free	6.87M	38.02%
RFNiN-Scale 3rd-order	Gauss	6.83M	69.65%

Table 2: Classification on ILSVRC2012-100 to illustrate influence of factorization on performance. The results show that the advantage of the Gaussian basis is substantial and our results are not merely due to a change in architecture.

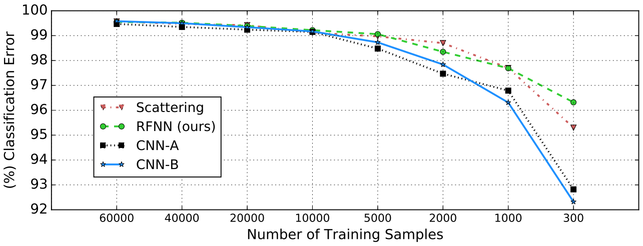

Figure 5: Classification performance of the Scattering Network on various subsets of the MNIST dataset. In comparison the state of the art CNN-A from [26]. RFNN denotes our receptive field network, with the same architecture as CNN-B. Both are shown, to illustrate that good performance of the RFNN is not due to the CNN architecture, but due to RFNN decomposition. Our RFNN performs on par with Scattering, substantially outperforming both CNNs.

4.2. Experiment 2: Scattering and RFNNs

Small simple domain. We compare an RFNN to Scattering in classification on reduced training sizes of the MNIST dataset. This is the domain where Scattering outperforms standard CNNs [2]. We reduce the number of training samples when training on MNIST as done in [2]. The network architecture and training parameters used in this section are the same as in [38]. The RFNN contains 3 layers with a third order basis on one scale as a multiscale basis didn't provide any gain. Scale and order are determined on a validation set. Each basis layer is followed by a layer of $\alpha_N = 64$ 1x1 units that linearly re-combine the basis filters outcomes. As comparison we re-implement the same model as a plain CNN. The CNN and Scattering results on the task are taken from [2, 26].

Results are shown in Figure 5, each number is averaged over 3 runs. For the experiment on MNIST the gap between the CNNs and networks with pre-defined filters increases when training data is reduced, while RFNN and Scattering perform on par even at the smallest sample size.

Model	Cifar-10	Cifar-100
Roto-Trans Scattering	82.30%	56.80%
RFNiN	86.31%	63.81%
RCNN	91.31%	68.25%

Table 3: Comparison against Scattering on a large complex domain. State-of-the-art comparison is given by RCNN. RFNiN outperforms Scattering by large margins.

Large complex domain. We compare against Scattering on the Cifar-10 and Cifar-100 datasets, as reported by the recently introduced Deep Roto-Translation Scattering approach [30], a powerful variant of Scattering networks explicitly modeling invariance under the action of small rotations. This is a domain where CNNs excel and learning of complex image variabilities is key.

The RFNiN is again a variant of the standard NiN for Cifar-10. It is similar to the model in experiment 1, just that it has one basis layer, two 1x1 convolution layers and one pooling layer less and the units in the 1x1 convolution layers are 192 in the whole network. Furthermore, we show performance of the state-of-the-art recurrent convolutional networks (RCNNs) [16] for comparison.

The results in Table 3 show a considerable improvement on Cifar-10 and Cifar-100 when comparing RFNiN to Roto-Translation Scattering [30], which was designed specifically for this dataset. RCNNs performance is considerably higher as they follow a different approach to which structured receptive fields can also be applied if desired.

RFNNs are robust to dataset size. From these experiments, we conclude that RFNNs combine the best of both worlds. We outperform CNNs and compete with Scattering when training data is limited as exemplified on subsets of MNIST. We capture complex image variabilities beyond the capabilities of Scattering representations as exemplified on the datasets Cifar-10 and Cifar-100 despite operating in a similarly smooth parameter space on a receptive field level.

4.3. Experiment 3: Limiting datasize

To demonstrate the effectiveness of the RF variant compared to the Network in Network, we reduce the number of classes in the ILSVRC2012-dataset from 1000 to 100 to 10, resulting in a reduction of the total number of images on which the network was trained from 1.2M to 130k to 13k and subsequent decrease in visual variety to learn from. To demonstrate performance is not only due to smaller number of learnable parameters, we evaluate two RFNiN versions. RFNiN-v1 is RFNiN-Scale 3rd-order from table1. RFNiN-v2 is one layer deeper and wider [128/128/384/512/1000] version of the RFNiN-v1, resulting in 3 million additional parameters, which is 2,5 million more than NiN.

The results in table 4 show that compared to CNNs the

Model	#Params	1000-class	100-class	10-class
NiN	7.5M	56.78%	67.30%	76.97%
RFNiN-v1	6.8M	50.08%	69.65%	85.00%
RFNiN-v2	10M	54.04%	70.78%	83.36%

Table 4: Three classification experiments on ILSVRC2012 subsets. Results show that the bigger model (RFNiN-v2) performs better than RFNiN-Scale 3^{rd}-order (RFNiN-v1) on the 1000-classes while on 100-class and 10-class, v1 and v2 perform similar. The gap between RFNiN and NiN increases for fewer classes.

RFNiN performance is better relatively speaking when the number of samples and thus the visual variety decreases. For the 13k ILSVRC2012-10 image dataset the gap between RFNiN and NiN increased to 8.0% from 2.4% for the 130k images in ILSVRC2012-100 while the best RFNiN is inferior to NiN by 2.98% for the full ILSVRC2012-1000. This supports our aim that RFNiN is effectively incorporating natural image priors, yielding a better performance compared to the standard NiN when training data and variety is limited, even when having more learnable parameters. Truly large datasets seem to contain information not yet captured by our model.

4.4. Experiment 4: Small realistic data

We apply an RFNiN to 3D brain MRI classification for Alzheimer's disease [4] on two popular datasets. Neuroimaging is a domain where training data is notoriously small and high dimensional and no truly large open access databases in a similar domain exist for pre-training.

We use a 3-layer RFNiN with filters sizes [128,96,96] with a third order basis in 3 scales $\sigma \in \{1, 4, 16\}$. This time wider spaced, as the brains are very big objects and are centered due to normalization to MNI space with the FSL library [8]. Each basis layer is followed by one 1x1 convolution layer. Global average pooling is applied to the final feature maps. The network is implemented in Theano [1] and trained with Adam [10].

The results are shown in table 5. Note that [7, 24] train on their own subset and use an order of magnitude more training data. We follow standard practice [4] and train on a smaller subset. Nevertheless we outperform all published methods on the ADNI dataset. The same 3 layer NiN as our RFNiN model has 84.21% accuracy, more than 10% worse while being hard to train due to unstable convergence. On the OASIS AD-126 Alzheimer's dataset [21], we achieve an accuracy of 80.26%, compared to 74.10% with a SIFT-based approach [3]. Thus, we show our RFNiN can effectively learn comparably deep representations even when data is scarce and exhibits stable convergence properties.

3D MRI classification	Accuracy	TPR	SPC
3D-RFNiN (ours)	**97.79%**	**97.14%**	**98.78%**
ICA [36]	80.70%	81.90%	79.50%
Voxel-Direct-D-gm [4]	-	81.00%	95.00%
3D-CNN [24]	95.70%	-	-
NIB [7]	94.74%	95.24%	94.26%

Table 5: Alzheimer's classification with 150 train and test 3D MRI images from the widely used ADNI benchmark. RFNiN, ICA and Voxel-Direct-D-gm are trained on the subset introduced in [4], 3D-CNN and NIB were trained on their own subset of ADNI, using an order of magnitude more training data. RFNiN outperforms all published results. Reported is accuracy, true positive rate and specificity.

5. Discussion

The experiments show that structuring convolutional layers with a filter basis grounded on Scale-space principles improves performance when data is limited. The filter basis provides regularization especially suited for image data by restricting the parameter space to smooth features up to fourth order. The Gaussian derivative basis opens up a new perspective for reasoning in CNNs, connecting them with a rich body of prior multiscale image analysis research that can now be readily incorporated into the models. This is especially interesting for applications where model insight and control is key.

We illustrated the effectiveness of RFNNs on multiple subsets of Imagenet, Cifar-10, Cifar-100 and MNIST. The choice of a third order Gaussian basis is sufficient to tackle all datasets which is in accordance with prior research [12, 2]. While it remains an open problem to match the performance of CNNs on very large datasets like the 1000-class ILSVRC2012, our results show that the RFNN method outperforms CNNs by large margins when data are scarce. It can also outperform CNNs on challenging medium sized datasets while being superior to Scattering on large datasets despite having more parameters as the pre-defined basis restriction allows the network to devote its full capacity to a sensible feature spaces. As a small data real world example, we verify our claims with 3D MRI Alzheimer's disease classification on two datasets where we consistently achieve competitive performance including the best results on the widely used ADNI dataset.

Acknowledgements. We thank Rein van den Boomgaard and Silvia-Laura Pintea for insightful discussions. This work has been conducted with data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) and the Open Access Series of Imaging Studies (OASIS).

References

[1] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*, 2012. 5, 8

[2] J. Bruna and S. Mallat. Invariant scattering convolution networks. *TPAMI*, 35(8):1872–1886, 2013. 2, 3, 4, 5, 6, 7, 8

[3] Y. Chen, J. Storrs, L. Tan, L. J. Mazlack, J.-H. Lee, and L. J. Lu. Detecting brain structural changes as biomarker from magnetic resonance images using a local feature based svm approach. *Journal of neuroscience methods*, 221:22–31, 2014. 8

[4] R. Cuingnet, E. Gerardin, J. Tessieras, G. Auzias, S. Lehéricy, M.-O. Habert, M. Chupin, H. Benali, O. Colliot, A. D. N. Initiative, et al. Automatic classification of patients with alzheimer's disease from structural mri: a comparison of ten methods using the adni database. *neuroimage*, 56(2):766–781, 2011. 8

[5] L. M. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever. Scale and the differential structure of images. *Image and Vision Computing*, 10(6):376–388, 1992. 2

[6] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *TPAMI*, (9):891–906, 1991. 4

[7] A. Gupta, M. Ayhan, and A. Maida. Natural image bases to represent neuroimaging data. In *ICML*, 2013. 8

[8] M. Jenkinson, C. F. Beckmann, T. E. Behrens, M. W. Woolrich, and S. M. Smith. Fsl. *Neuroimage*, 62(2):782–790, 2012. 8

[9] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 5

[10] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 8

[11] J. J. Koenderink. The structure of images. *Biological cybernetics*, 50(5):363–370, 1984. 1, 2, 3

[12] J. J. Koenderink and A. J. van Doorn. Representation of local geometry in the visual system. *Biological cybernetics*, 55(6):367–375, 1987. 1, 4, 6, 8

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2, 5

[14] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 1

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[16] M. Liang and X. Hu. Recurrent convolutional neural network for object recognition. In *CVPR*, 2015. 7

[17] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv:1312.4400*, 2013. 3, 5

[18] T. Lindeberg. *Scale-space theory in computer vision*, volume 256. Springer Science & Business Media, 2013. 1, 2, 3

[19] S. Mallat. *A wavelet tour of signal processing*. Academic press, 1999. 3

[20] S. Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012. 2, 3

[21] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris, and R. L. Buckner. Open access series of imaging studies (oasis): cross-sectional mri data in young, middle aged, nondemented, and demented older adults. *Journal of cognitive neuroscience*, 19(9):1498–1507, 2007. 8

[22] J. Moody, S. Hanson, A. Krogh, and J. A. Hertz. A simple weight decay can improve generalization. *NIPS*, 1995. 2

[23] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014. 2

[24] A. Payan and G. Montana. Predicting alzheimer's disease: a neuroimaging study with 3d convolutional neural networks. *arXiv preprint arXiv:1502.02506*, 2015. 8

[25] P. Perona. Steerable-scalable kernels for edge detection and junction analysis. In *ECCV*, pages 3–18. Springer, 1992. 4

[26] M. A. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR*. IEEE, 2007. 7

[27] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *CVPR*, 2014. 1

[28] B. M. H. Romeny. *Front-end vision and multi-scale image analysis: multi-scale computer vision theory and applications, written in mathematica*, volume 27. Springer Science & Business Media, 2008. 3

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, pages 1–42, 2015. 1, 2, 5

[30] L. Sifre and S. Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *CVPR*, 2013. 2, 3, 6, 7

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014. 3

[32] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 2014. 2

[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv:1409.4842*, 2014. 2, 3, 5

[34] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*. IEEE, 2014. 2

[35] A. P. Witkin. Scale-space filtering. In *International Joint Conference on Artificial Intelligence*, 1983. 1, 2, 3

[36] W. Yang, R. L. Lui, J.-H. Gao, T. F. Chan, S.-T. Yau, R. A. Sperling, and X. Huang. Independent component analysis-based classification of alzheimer's mri data. *Journal of Alzheimer's disease: JAD*, 24(4):775, 2011. 8

[37] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *NIPS*, 2014. 1

[38] M. D. Zeiler and R. Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*, 2013. 7

[39] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 1

[40] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014. 1