Dynamic Steerable Blocks in Deep Residual Networks

Jörn-Henrik Jacobsen[1]
j.jacobsen@uva.nl
Bert de Brabandere[2]
bert.debrabandere@esat.kuleuven.be
Arnold W.M. Smeulders[1]
A.W.M.Smeulders@uva.nl

[1] Informatics Institute
University of Amsterdam
Amsterdam, NL

[2] ESAT-PSI
KU Leuven,
Leuven, BE

Abstract

Filters in convolutional networks are typically parameterized in a pixel basis, that does not take prior knowledge about the visual world into account. We investigate the generalized notion of frames, that can be designed with image properties in mind, as alternatives to this parametrization. We show that frame-based ResNets and Densenets can improve performance on Cifar-10+ consistently, while having additional pleasant properties like steerability. By exploiting these transformation properties explicitly, we arrive at dynamic steerable blocks. They are an extension of residual blocks, that are able to seamlessly transform filters under pre-defined transformations, conditioned on the input at training and inference time. Dynamic steerable blocks learn the degree of invariance from data and locally adapt filters, allowing them to apply a different geometrical variant of the same filter to each location of the feature map. When evaluated on the Berkeley Segmentation contour detection dataset, our approach outperforms all competing approaches that do not utilize pre-training, highlighting the benefits of image-based regularization to deep networks.

1 Introduction

Deep Convolutional Neural Networks (CNNs) are the state-of-the-art solution to many vision tasks [29]. However, they are known to be data-inefficient, as they require up to millions of training samples to achieve their powerful performance [41, 42]. In this work, we propose a formulation of CNNs that more efficiently learns to exploit generic regularities known to be present in the data a priori.

For images, as well as any other sensory data, CNNs typically learn filters from individual pixel values. In this paper, we show that alternatives to the pixel basis are more natural formulations for learning models on locally well-understood data like images. We show increased classification performance in state-of-the-art ResNets [18] and Densenets [21] on the highly competitive Cifar-10 classification task by replacing the pixel basis with a basis more suitable for natural images. Further, we show that such a replacement naturally leads to powerful extensions of residual blocks as dynamic steerable interpolators that can steer

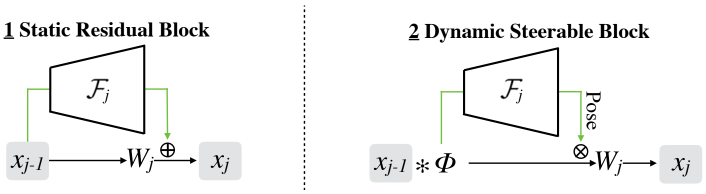


Figure 1: Left is a classical residual block as used in vanilla ResNets [17]. Outputs x_{j-1} of the previous block are combined additively with the output of a small stack of convolutional layers $\mathcal{F}_j$ to form the final output x_j. We augment this formulation in two ways to achieve our dynamic block formulation on the right. First, we apply a change to a steerable frame Φ on the input x_{j-1} and second we replace the addition operation with a multiplication. This permits us to interpret $\mathcal{F}_j$ as a pose estimating network, that directly outputs the linear projection coefficients, transforming the basis and subsequently the effective filters adaptively in a dynamic and, if desired, location dependent manner.

their filters conditioned on the input with respect to pre-defined continuous geometric transformations like rotations or scalings. The proposed block allows the network to adaptively learn the degree of local invariance required for each filter, by decoupling filter learning and local geometrical pose adaption. We show the effectiveness of this approach on the BSD500 boundary detection task, where precise and adaptive local invariances are key. We outperform all competing non-pretrained methods with an approach that converges faster than competing approaches.

Our contributions:

- We introduce the notion of frame bases to CNNs and show that classically used frames from Computer Vision aid optimization, when compared to the commonly used pixel basis. We illustrate this by improving classification performance on Cifar-10+ for multiple ResNet and Densenet architectures.

- Exploiting the steerability properties of such alternative bases further, we introduce Dynamic Residual Blocks that are able to continuously transform features in a locally adaptive manner and illustrate the approach on a synthetic tasks to highlight the advantages over competing approaches.

- To evalute the practicality of our proposed approach, we apply a Dynamic Residual Block network on the BSDS-500 contour detection task [2] where we achieve state-of-the-art performance among all published results without pre-training.

2 Related Work

Steerable Filters is a concept established early for signal processing. Initially introduced by [14], the concept was extended to the Steerable Pyramid by [34] and to a Lie-group formulation by [19, 31]. Further, steerability has recently been related to tight frames, presenting Simoncelli's Steerable Pyramid and multiple other Wavelets arising as a special case of the non-orthogonal Riesz transform [36]. Steerable pyramids have been applied to CNNs as a pre-processing step [40], but have not yet been learnable. We incorporate steerable frames in CNNs to increase their de facto expressiveness and to allow them to learn their configurations, rather than picking them a priori.

Convolutional Networks with alternative bases have been proposed with various degrees of flexibility. A number of works utilizes change of basis to stabilize training and increase convergence behavior [3, 33]. Another line of research is concerned with complex-valued CNNs, either learned [35], or fully designed like the Scattering networks [4, 32].

Scattering, as well as the complex-valued networks, rest upon a direct connection between the signal processing literature and CNNs. Inspired by the former, Structured Receptive Field Networks are learned from an overcomplete multi-scale frame, effectively improving performance for small datasets due to restricted feature spaces [23]. Closely related is another line of recent promising work on group-equivariant [7, 12] and steerable CNNs [8, 37]. The latter build steerable representations via appropriately chosen basis functions, illustrating that CNNs with well-chosen geometrical inductive biases consistently outperform state-of-the-art approaches in multiple domains. However, both all these approaches rely on hand-engineered types of representations and none considers locally adaptive filtering with learned degree of invariance, we aim to bridge this gap. Inspired by CNNs learned from alternative bases, we introduce the general principle of Frame-based convolutional networks that allow for non-orthogonal, overcomplete and steerable feature spaces.

Another way to impose structure onto CNN representations and subsequently increase their data-efficiency is by pre-defining the possible transformations, as done in Transforming Autoencoders [20], which map their inputs from the image to pose space through a neural network. The Spatial Transformer Networks [24] learn global, and deformable convolutional networks [9] local transformation parameters in a similar way while applying them to a nonlinear co-registration of the feature stack to some estimated pose. Dynamic Filter Networks [11] move one step further and estimate filters for each location, conditioned on their input. These approaches are all dynamic in a sense that they condition their parameters on the input appearance. Our proposed dynamic residual block can be interpreted as a middle ground that combines the idea of Dynamic Filter Networks with explicit pose prediction into blocks that can locally estimate filter poses from a continuous input space. As such, we overcome the difficulty of estimating local filter pose, while being able to separate pose and feature learning globally without the need for differentiable samplers or locally connected layers.

3 Dynamic Steerable Two-Factor Blocks

Deep Residual Networks (ResNets) are among the best performing current approaches to convolutional networks. Instead of optimizing single layers, they consist of convolutional blocks with skip connections, where the output of block j is defined as $x_j = \mathcal{H}(x_{j-1}) = \mathcal{F}(x_{j-1}) + x_{j-1}$. $\mathcal{F}$ is typically a stack of convolution layers, batch normalizations [22] and nonlinearities [17, 18]. Such residual blocks overcome vanishing gradients and facilitate optimization [15], leading to very simple, but very deep networks with no need for pooling and fully connected layers. In the following sections, we introduce extensions of residual blocks as two-factor models, that overcome the static nature of typical CNNs. A simple extension transforms static residual blocks into dynamic modules, able to change the geometrical pose of their filters conditioned on the input during training *and* inference time in a locally adaptive fashion.

The key perspective this work relies upon is to change the residual block from being an additive model of the form:

$$\mathcal{H}(x) = \mathcal{F}(x) + Wx, \tag{1}$$

where W is the shortcut projection typically introdcued in a recent improvement [18], to a multiplicative two-factor model of the form:

$$\mathcal{H}(x) = \mathcal{F}(\Phi(x))W\Phi(x). \tag{2}$$

Here, the factor $W\Phi(x)$ represents a canonical feature, while $\mathcal{F}(\Phi(x))$ represents its pose. Note that the purpose of the operator W changes from a tool to increase channels via linear projection in standard resnets to representing the learned canonical filters in our proposed dynamic blocks. The function $\mathcal{F}$ is often called interpolation equation in the steerable filter literature [34]. The space of possible poses can be pre-conditioned by choosing a suitable set of basis functions Φ, that are steerable under pre-defined sets of deformations and thus span an invariant subspace of all transformed versions of the basis itself. In the following, we introduce generalized and steerable bases as alternatives to the in CNNs commonly used pixel basis.

Secondly, we augment the residual block to effectively leverage the transformation properties of the basis, leading us to locally adaptive steerable models with learnable degree of invariance, while being dynamically conditioned on the input during training and inference time.

3.1 CNN Bases Beyond Pixels

Replacing the pixel basis in favour of bases that are steerable, which means equivariant with respect to some group of transformation, linearizes the action of these transformation groups and makes it easier for the network to learn to transform filters dynamically. It also permits to directly learn the degree of invariance to variabilities like rotation, scaling and others, depending on the type of basis used. If $g(\tau) \in G$ is some transformation of the input x and the basis Φ is steerable under this transformation, one can find a linear mapping $\mathcal{F}(\tau)$, such that:

$$WF(\tau)\Phi(x_{j-1}) = W\Phi(g(\tau)x_{j-1}). \tag{3}$$

This means instead of steering the effective filters represented by $W\Phi(x_{j-1})$, it is sufficient to steer the basis Φ and the effective filters will be transformed accordingly. Thus, a given change to a steerable basis Φ, leads to dynamically adapting filters, as well as local and global invariants that can be learned during training efficiently.

The most general set of viable bases to learn filters from are called *Frames* [6]. Frames are a natural generalization of orthogonal bases and are spanning sets that span the same space of functions an orthogonal basis does, while allowing for overcomplete representations and hence more densely sampled parameter spaces. Frames can be seen as a superset of orthogonal bases in the sense that every basis is a frame, but not the reverse, see figure 2. Frames have three main advantages: 1) Many frames are steerable and thus provide us with a signal representation that can be dynamically steered by simple linear projections, i.e. they have the ability to linearize group actions; 2) Frames can spell out signal properties more explicitly, facilitating optimization when good frames for the type of data are known, as is the case for many types of signals like images or video [6]. 3) Frames allow for overcomplete representations of the signal adding robustness and regularization to the optimization procedure, as they are more stable when measurements or updates are noisy.

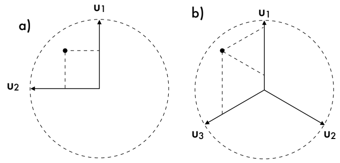


Figure 2: a) Is an orthonormal basis in $\mathbb{R}^2$, u_1 and u_2 are linearly independent and span the space of $\mathbb{R}^2$. A dot in this example represents a filter in a convolutional network with coefficients $\{u_1, u_2\}$. b) A tight frame in $\mathbb{R}^2$. u_1, u_2 and u_3 are linearly dependent. A dot in this example represents a convolutional filter with coefficients $\{u_1, u_2, u_3\}$. The frame is an overcomplete representation, again spanning $\mathbb{R}^2$ and again preserving the norm. Note that the set of filter coefficients as represented by the dot is not unique. Thus even if one u is obstructed by noisy updates or measurements, the filter may still be robust.

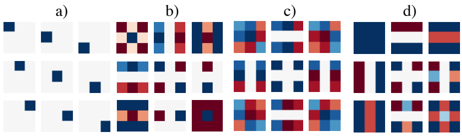


Figure 3: An illustrative plot of multiple 3x3 spanning sets Φ: a) Pixel-basis, b) Gaussian Derivatives, c) Non-orthogonal Framelet, d) Naive Frame. Note the increased symmetries in the three latter.

In a standard convolutional network, a filter kernel is a linear combination over the standard (pixel) basis for $l^2(\mathbb{N})$. This pixel basis is composed of delta functions for every dimension and W_i is the i_{th} filter of the network with parameters w_n^i. Without loss of generalization the orthonormal standard basis can be replaced by a frame to include steerability, non-orthogonality, overcompleteness and increased symmetries into the representation. Changing from the pixel to an arbitrary frame is as simple as replacing the pixel basis e_n with a frame of choice with elements ϕ_m as follows:

$$W_j \Phi(x_{j-1}) = \sum_{n=1}^{N} w_n^j e_n \star x_{j-1} \equiv \sum_{m=1}^{M} w_m^j \phi_m \star x_{j-1}, \tag{4}$$

where $w_1^i, \ldots, w_m^i$ are again the filter coefficients being learned. Note that after optimization with an overcomplete frame is done, the resulting network can be rewritten in terms of the standard pixel basis. Therefore frames only act as a regularizer during training, they do not increase the effective parameter cost of the network.

In practice for CNNs working on images we investigate four typical choices of frames: i) the vanilla orthogonal pixel basis, ii) Gaussian derivatives, one of the most widely used overcomplete frames from the computer vision literature (also used in SIFT) [13, 23, 26, 30], iii) Framelets, a non-orthogonal but not overcomplete basis, designed for images [10] and iv) A "naive" frame of the form $x^p y^q$ derived from steerability requirements [19] but with no image properties in mind. See figure 3 for this selection of frames.

When training a CNN the functions represented by each feature naturally change from update to update. It is desirable to separate the frame functions from the effective features as learned by the network. To be able to separate a feature's pose from its canonical appearance, we are interested in a steerable version of an arbitrary filter $W_i(x, y)$ under a k-parameter Lie group. From equation 4 follows:

$$g(\tau)W_i(x,y) = \sum_{n=1}^{N} w_n^i g(\tau)\phi_n^i. \tag{5}$$

And by substituting according to equation 3 it follows:

$$g(\tau)W_i(x,y) = \sum_{n=1}^{N} w_n^i \sum_{m=1}^{M} \mathcal{F}_m(\tau)\phi_m(x). \tag{6}$$

Thus it is sufficient to determine the group action on the fixed frame by steering it to separate the canonical feature itself from its k-parameter variants, i.e. ϕ_n^i govern the weight of each frame coefficient to form a feature $W_i(x,y)$ and $\mathcal{F}(\tau)_m$ are the steering functions governing the transformation of $g(\tau)$ acting on $W_i(x,y)$ as a whole. To achieve precise geometrical regularization, one can further derive the steering equations for the particular steerable frame at hand and use the resulting trigonometric functions as activation functions, which is suitable for learning in a CNN. While these activation functions can be omitted in more general tasks like classification where the demand on local transformations is not precisely defined, we will show below that they serve as important regularization in tasks where precise geometric adaption is needed, as examplified in the boundary detection experiments. For brevity we moved the derivation of steerability properties of the frames i)-iv) to the supplementary material.

The blocks used in the boundary detection experiments are based on first order Gaussian derivatives, steerable with respect to continuous rotations and small ranges of isotropic scalings (σ =0.8-1.5). The Dynamic Two-Factor Block consists of four processing steps. 1) Change from input to frame space by convolving with frame, 2) The interpolator network $\mathcal{F}(\Phi(x))$ estimates local pose from this invariant subspace, outputting a set of pose variables for each location in the image and for each input/output channel (this can be changed depending on application). For the interpolator network (F) we used a small network with 8-16 units and three layers with tanh nonlinearities as they seemed suitable to approximate trigonometric steering equation solutions. For scalings, we found softplus and relu nonlinearities to work well. Optional: 3) the steering functions derived in section 2.4 are applied to the pose variable maps and effectively act as nonlinear pose-parametrized activation functions that regularize the interpolation network to output an explicitly interpretable pose space. 4) A 1x1 convolution layer is applied to the already transformed frame outputs, this convolution represents the weights w_j^i of each feature governing the canonical appearance of the i_{th} feature map in the j_{th} layer.

4 Experiments

The experimental section is organized in two parts. The first part illustrates that our proposed approach does work as intended and shows that we can even outperform highly optimized and flexible approaches like ResNets and Densenets and fully convolutional networks, in domains where they excel. The second part focuses on illustrating our dynamic steerable block mechanism and applying it on a difficult real-world task of boundary detection, where we achieve state-of-the-art results among all other non pre-trained methods.

Method	ResNet56	ResNet110	DensenetK12L40	DensenetK12L100
Pixel Basis	6.70± 0.16%	5.88± 0.22%	5.26± 0.19%	4.16 ± 0.18%
Image Frame	**6.11**± 0.19%	**5.33**± 0.15%	**4.97**± 0.18%	**3.78** ± 0.17%
Naive Frame	7.29± 0.31%	6.93± 0.29%	6.40± 0.21%	5.25 ± 0.20%

Table 1: Results reported as average over 5 runs with standard deviation on Cifar10+. We evaluate multiple models with a standard pixel-basis, a steerable frame basis designed for natural images and a naive steerable frame as an example of a frame that does not take natural image statistics into account. The natural image statistics based frame outperforms the pixel-basis consistently, while the naive frame consinstently performs about 1% worse than the baseline, highlighting the benefit of a frame suitable for the type of input data.

4.1 Generalized Bases on Cifar-10+

To show the validity of generalized basis representations, we compare different bases in multiple state-of-the-art deep Residual Network [18] and Densenet [21] architectures on the Cifar-10 [28] dataset with moderate data augmentation of crops and flips.

We evaluated our approach on two different network sizes that still comfortably train on one GPU over night each. The setup used for the ResNet is as described in [18]. The batch size is chosen to be 64 and we train for 164 epochs with the described learning rate decrease. The ResNet architectures used are without bottlenecks having 56 and 110 layers. For the Densenets we follow [21] and evaluate on the K=12 and L=40, and the K=12 and L=100 models. We run our experiments in Keras [6] and Tensorflow [1]. In the first experiment, we run the models on the standard pixel basis to get a viable baseline. Results are in line or better with the numbers reported by the authors. Secondly, we replace the pixel-basis with widely-used frames that take natural image statistics into account, namely non-orthogonal, overcomplete Gaussian derivatives [13] and non-orthogonal framelets [10] in an alternating fashion, yielding superior performance compared to the pixel-basis by replacement only. We also show that the naively derived $x^p y^q$ frame (see supplementary material for derivation) performs consistently worse than the other two choices, as it does not take natural image properties into account. The frames used, are shown in figure 3. The results are reported in table 1. The fact that the pixel-basis can be replaced by steerable frames with well-understood properties while performance improves is remarkable. Frame-based CNNs run at approximately the same runtime as vanilla CNNs.

4.2 Dynamic Steerable Blocks for Boundary Detection

In this section we evaluate our proposed dynamic steerable two-factor blocks on boundary detection, a natural task for locally adaptive filtering. In the first part, we illustrate properties of the proposed mechanism, while in the second part we apply it to a challenging real-world problem, where we outperform competing approaches.

4.2.1 Evaluating Boundary Detection Properties on Textured Blobs

In this illustration we empirically validate the effectiveness of our approach with a single dynamic two-factor block by showing that it indeed learns adaptive and non-trivial invariants, that are conditioned on local neighborhoods in the the image. To show this, we create an

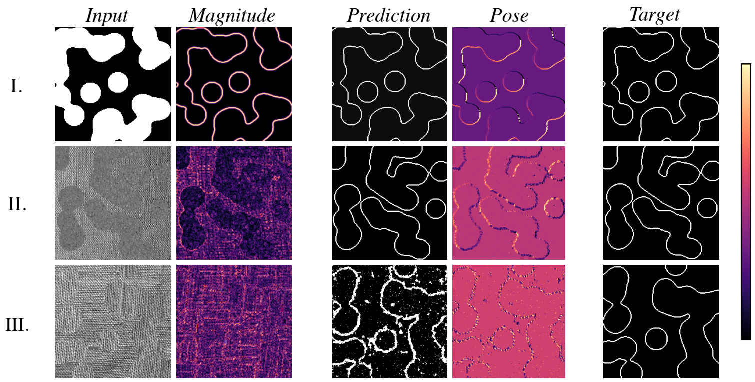
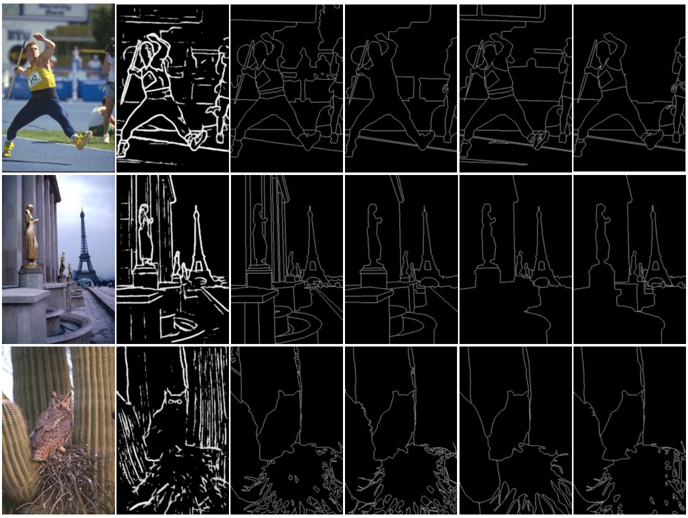


Figure 4: Results on the illustrative boundary detection task. From left to right: Input to the network; Gradient magnitude of the inputs, here used for illustration of an example of a simple invariant obtained from the underlying frame (this is neither input nor output of the network, sole illustration purpose). Pixel-wise predictions of the network and pose variables discovered by the interpolator network that led to them; Pixel-wise labels. I) Perfectly discovers the simple gradient magnitude invariance. II) More complex local relationships that can not be solved with simple invariants are recovered. Note, the two small blobs on the right, that are barely visible in input and gradient magnitude map, but are still correctly segmented. III) Most challenging scenario, as blobs and backgrounds can hardly be distinguished and the gradient magnitude does not contain much useful visible information for the task either. Note, that the interpolator network finds an alternating pattern of rotation, that overcomes the irregular local gradients and recovers most of the targets successfully.

artificial dataset of random blobs, whose boundaries have to be detected by a dynamic two-factor block in a pixel-wise classification task. The dataset is infinite and created on the fly. In the first case, where blob and background are binary, this task can be solved with a simple gradient magnitude invariant and presents no challenge to our algorithm and a fully convolutional network baseline of the same size achieves almost the same performance. In the second example we sample textures from the KTH TIPS dataset [16] and fill blobs as well as background with different textures each. Here, gradient magnitude either only gives weak clues, or in many cases is unable to find any outline given by the target. In both cases, the fully convolutional baseline fails to converge. Remarkably, even though the dynamic two-factor block does only receive two Gaussian derivative gradient filters as an input, it still manages to find highly non-linear steering patterns to recover the boundaries. The results are shown in figure 4. We show some unseen inputs alongside the manually calculated gradient magnitude, predictions and associated pose maps (only rotation variable shown) as estimated by the dynamic block and labels. Even in very hard cases, the dynamic block manages to largely recover the labeled boundaries. Results are evidence of the ability of our proposed method to learn adaptive invariants conditioned on the local context in the image and can do so in ways that go way beyond simple gradient magnitude guided steering.

4.2.2 Boundary Detection on BSD500

In this experiment we apply the dynamic two-factor block on a real task in which adaptive invariance is desirable. Recently, a fully equivariant CNN that was engineered to output



Method	DynResNet (Ours)	H-Net [37]	ResNet (Ours-Static)	Kivininen [25]	HED [39]	DCNN Pre-trained [27]
ODS	**0.732**	0.726	0.720	0.702	0.697	0.813
OIS	**0.751**	0.742	0.733	0.715	0.709	0.831

Figure 5: Results on BSD500. Our residual network approach with dynamic steerable blocks outperforms all other methods when no pre-training on large datasets is performed.

locally rotation invariant predictions [37] performed very well on this task. We aim to show, that learned adaptive invariance might be of benefit here, as even though final edge prediction is rotation invariant, intermediate processing might benefit from a context dependent degree of invariance.

We evaluate our method on the contour detection task of the Berkeley Segmentation Dataset. The dataset consists of 500 images divided in a train/val/test split. We tune hyperparameters on the validation set and report results on the test set, following the protocol in [2]. Each image is labelled by 5-7 human annotators, resulting in multiple ground truth maps per image. We follow [37] and merge the different labels by majority vote into one. We minimize the pixelwise binary cross-entropy loss between ground truth and prediction and use class balancing because the classes are heavily imbalanced (many more background pixels than contour pixels).

We use a ResNet model designed for segmentation [38] and reduce its size drastically, due to the limited data-scenario we face. Eventually we use a network with the following structure: Conv2d[64]->DynResBlock[128]->StatResBlock[128]->DynResBlock[128]->StatResBlock[128]->Conv2D[256]. Thus, we alternate static and dynamic blocks as we found that this setup stabilizes training considerably on the validation set. We report the OIS and ODS metrics as in [2], based on F-scores. Due to the subjective aspect of the countour segmentation, the task is ambiguous and the network implicitly has to estimate the level of detail at which to segment the boundaries. While in most cases the network's prediction agrees with at least one of the human annotators (first two rows of figure 5), it sometimes segments boundaries at a too high level of detail (last row of figure 5). Contrary to [37], we find that adding a sparsity penalty to the loss function does not increase performance in a significant manner. Our results show that we outperform all competing approaches that have not been pre-trained on Imagenet, according to [37], including our static ResNet baseline and another explicitly rotation invariant approach [37].

5 Conclusion

We have introduced the notion of Frame-based convolutional networks. Our experiments illustrate that a simple replacement of the standard basis by a frame suitable for natural images leads to increased performance on highly optimized networks. The insight that multiple frames can be considered as viable spanning sets for CNN representations leads us to dynamic steerable two-factor blocks that can transform their filters dynamically in a location-dependent manner. We show the effectiveness of our approach by application on the Berkeley Contour detection benchmark, where we perform favourably.

References

[1] Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011.

[3] Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. *arXiv preprint arXiv:1511.06464*, 2015.

[4] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013.

[5] François Chollet. Keras. https://github.com/fchollet/keras, 2015.

[6] Ole Christensen. *An introduction to frames and Riesz bases*, volume 7. Springer, 2003.

[7] Taco S Cohen and Max Welling. Group equivariant convolutional networks. *arXiv preprint arXiv:1602.07576*, 2016.

[8] Taco S Cohen and Max Welling. Steerable cnns. *arXiv preprint arXiv:1612.08498*, 2016.

[9] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. *arXiv preprint arXiv:1703.06211*, 2017.

[10] Ingrid Daubechies, Bin Han, Amos Ron, and Zuowei Shen. Framelets: Mra-based constructions of wavelet frames. *Applied and computational harmonic analysis*, 14(1): 1–46, 2003.

[11] Bert De Brabandere, Xu Jia, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. *arXiv preprint arXiv:1605.09673*, 2016.

[12] Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. Exploiting cyclic symmetry in convolutional neural networks. *arXiv preprint arXiv:1602.02660*, 2016.

[13] Luc MJ Florack, Bart M ter Haar Romeny, Jan J Koenderink, and Max A Viergever. Scale and the differential structure of images. *Image and Vision Computing*, 10(6): 376–388, 1992.

[14] William T Freeman and Edward H Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 13(9):891–906, 1991.

[15] Klaus Greff, Rupesh K Srivastava, and Jürgen Schmidhuber. Highway and residual networks learn unrolled iterative estimation. *arXiv preprint arXiv:1612.07771*, 2016.

[16] Eric Hayman, Barbara Caputo, Mario Fritz, and Jan-Olof Eklundh. On the significance of real-world conditions for material classification. *Computer Vision-ECCV 2004*, pages 253–266, 2004.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. *arXiv preprint arXiv:1603.05027*, 2016.

[19] Yacov Hel-Or and Patrick C Teo. Canonical decomposition of steerable functions. *Journal of Mathematical Imaging and Vision*, 9(1):83–95, 1998.

[20] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer, 2011.

[21] Gao Huang, Zhuang Liu, and Kilian Q Weinberger. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

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

[23] Jörn-Henrik Jacobsen, Jan van Gemert, Zhongyu Lou, and Arnold W.M. Smeulders. Structured receptive fields in cnns. 2016.

[24] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[25] Jyri Kivinen, Chris Williams, and Nicolas Heess. Visual boundary prediction: A deep neural prediction network and quality dissection. In *Artificial Intelligence and Statistics*, pages 512–521, 2014.

[26] Jan J Koenderink and Andrea J van Doorn. Representation of local geometry in the visual system. *Biological cybernetics*, 55(6):367–375, 1987.

[27] Iasonas Kokkinos. Pushing the boundaries of boundary detection using deep learning. *arXiv preprint arXiv:1511.07386*, 2015.

[28] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[29] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.

[30] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[31] Markus Michaelis and Gerald Sommer. A lie group approach to steerable filters. *Pattern Recognition Letters*, 16(11):1165–1174, 1995.

[32] Edouard Oyallon and Stéphane Mallat. Deep roto-translation scattering for object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2865–2873, 2015.

[33] Oren Rippel, Jasper Snoek, and Ryan P Adams. Spectral representations for convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 2449–2457, 2015.

[34] Eero P Simoncelli and William T Freeman. The steerable pyramid: a flexible architecture for multi-scale derivative computation. In *ICIP (3)*, pages 444–447, 1995.

[35] Mark Tygert, Joan Bruna, Soumith Chintala, Yann LeCun, Serkan Piantino, and Arthur Szlam. A mathematical motivation for complex-valued convolutional networks. *Neural computation*, 2016.

[36] Michael Unser and Nicolas Chenouard. A unifying parametric framework for 2d steerable wavelet transforms. *SIAM Journal on Imaging Sciences*, 6(1):102–135, 2013.

[37] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. *arXiv preprint arXiv:1612.04642*, 2016.

[38] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *arXiv preprint arXiv:1611.10080*, 2016.

[39] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1395–1403, 2015.

[40] Tianfan Xue, Jiajun Wu, Katherine L Bouman, and William T Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. *arXiv preprint arXiv:1607.02586*, 2016.

[41] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[42] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.