Supervised scale-regularized linear convolutionary filters

Loog, Marco; Lauze, François

Publication date
2017

Document Version
Final published version

Published in
British Machine Vision Conference 2017, BMVC 2017

Supervised Scale-Regularized Linear Convolutionary Filters

Loog, Marco; Lauze, François

Publication date
2017

Published in
Proceedings of BMVC 2017

Citation (APA)
Loog, M., & Lauze, F. (2017). Supervised Scale-Regularized Linear Convolutionary Filters. In Proceedings of BMVC 2017 (pp. 1-11)

Supervised Scale-Regularized Linear Convolutionary Filters

Marco Loog
http://prlab.tudelft.nl/

Pattern Recognition Laboratory
Delft University of Technology
Delft, NL
&
The Image Section
University of Copenhagen
Copenhagen, DK

François Lauze
http://www.diku.dk/

The Image Section
University of Copenhagen
Copenhagen, DK

Abstract

We start by demonstrating that an elementary learning task—learning a linear filter from training data by means of regression—can be solved very efficiently for feature spaces of very high dimensionality. In a second step, firstly, acknowledging that such high-dimensional learning tasks typically benefit from some form of regularization and, secondly, arguing that the problem of scale has not been taken care of in a very satisfactory manner, we come to a combined resolution of both of these shortcomings by proposing a technique that we coin scale regularization. This regularization problem can also be solved relatively efficient. All in all, the idea is to properly control the scale of a trained filter, which we solve by introducing a specific regularization term into the overall objective function. We demonstrate, on an artificial filter learning problem, the capabilities of our basic filter. In particular, we demonstrate that it clearly outperforms the de facto standard Tikhonov regularization, which is the one employed in ridge regression or Wiener filtering.

1 Introduction

Nowadays, many computer vision and image analysis tasks are tackled by means of pattern recognition and machine learning techniques. This work makes a step in the opposite direction. It does not reject the learning approach to computer vision, but it shows how tools form computer vision—in particular those variational methods standard in this field, can aid in efficiently solving some of the basic estimation tasks machine learners and pattern recognizers come across. The specific issue we consider is the problem of scale that, one way or the other, emerges in any learning tasks involving images or videos [9, 15, 17]. As so often, however, it is overlooked or at least dealt with in a way that leaves much to be desired. The idea in this work is to properly control the scale of a trained filter. This is solve by introducing a specific regularization term into the overall learning objective.

This work is really at the interface of computer vision and learning techniques and we will draw on terminology from both fields. The main concept from machine learning and pattern recognition that we use are learning (or training) from examples (especially in relation to linear regression) and the ideas underlying artificial neural network, and convolutional neural networks in particular (see, for instance, [2, 3, 10, 16]). The main computer vision concepts employed are from scale space theory [9, 15, 17]. This conceptually interesting, multiresolutional theory shaped our line of reasoning to a large extent. On the computational side, we draw from variational methods standard in classical image analysis and computer vision. Though originally developed as a response to how scale is treated in deep neural networks, our way of including scale should be of more general interest.

Pixel-based classification and regression. Supervised classification and regression techniques have been applied to a broad range of challenging image processing and analysis tasks. Learning-based pixel classification has been around at least since the 1960s. Early studies have been conducted particularly within the field of remote sensing and abutting areas [8]. Though these approaches initially have focussed primarily on the use of the multiple spectral bands that the observations consisted of, later work also include spatial features based on derivative operator, texture measures, and the like. An early overview of the general applicability of pixel classification can be found, for instance, in [11]. Training image filters on the basis of given input-output pairs by means of regression techniques seems to have been considered less often. The problem, as opposed to pixel classification, may be to obtain proper input and output image pairs for training. Also for this reason, possibly the most often studied application is the prediction of (supposedly) noiseless images from images corrupted with a known noise model [6, 12]. More advanced applications found their way into medical image analysis, in particular for filtering complex image structures out of chest radiographs [18, 20].

The past decade has seen a trend of so-called representation learning [1]. The idea is to avoid any initial (explicit) bias in the learning that entails from prespecifying the particular image features that are going to be used. Rather, one relies on raw input data (pixels in our case) and a complex learner, e.g. deep networks, that is capable of simulating the necessary filtering based on the raw input. Our results are not "deep" as we will deal with shallow networks with a single linear convolutional neuron [1], which is a basic element in the more complex deep structures referred to above. The problem we focus on is inferring an image-to-image mapping, which is not necessarily limited to image denoising. The core of the issue we study is how to control the complexity of this mapping. In our case, this is achieved by controlling the aperture scale at which the mapping operates. In current applications of convolutional neural networks [16], the spatial extent from which information is drawn is coarsely modeled by a rectangle with preset dimensions. We move away from such handcrafted filter sizes and propose to not prefix the spatial range explicitly—and to basically have every input pixel intensity in every location have potential influence on any of the output pixel value to be predicted. We integrate the influence of scale by a regularization term into the overall objective function that is used to determine the fit of the mapping to the training data. In this way, we can trade off the influence of the training data and the scale of the aperture in a gradual and controlled way.

Outline. Section 2 formulates the initial problem setting in mathematical terms. The loss on the data term considered is the regular squared error and we basically will be dealing with standard regularized least squares linear regression or ridge regression. The section shows that our regularized prediction problem can be seen as a convolution in which the convolution kernel is to be determined. As it turns out, the formulations allows us to solve ridge regression problems in features spaces with very high dimensionality (as every pixels values is a feature here) and with even larger numbers of observations. Section 3, covering the other essential part of our theory, argues that some form of regularization would typically be necessary, after which it introduces and explains our scale-regularized objective function. It also shows how to reformulate the optimization problem so that its minimization can be performed by means of a variational method and finally sketches a basic scheme to come to an actual solution. Though on an artificial task, Section 4 provides solid and convincing experiments in which we compare to standard ridge regression, which is the de facto standard. Section 5 discusses and concludes our contribution.

A preliminary version of this work appeared as arXiv preprint [19].

2 Regression and Supervised Filter Learning

Let us initially consider a collection of N input-output pairs of images $\{(\alpha_i,\beta_i)\}_{i=1}^N$ defined on the full image domain $\mathbb{R}^d$: $\alpha_i,\beta_i:\mathbb{R}^d\to\mathbb{R}$, where α_i and β_i come from some image spaces A and B, respectively. The variable d equals the dimensionality of the image domain, which can just taken to be 2 in most of what follows.

We do not consider multi-band or multi-spectral images, but out theory is equally applicable to this setting. Given these N pairs, of what we will refer to as training images, we would like to infer a transformation T that can be applied to any new and unseen input image α, such that it optimally predicts its associated, and unobserved, output β. The expected least squares loss L between the true output and the prediction by T is widely used to define optimality of the transformation T: $L[T]=\int_B\int_A p(\alpha,\beta)||T[\alpha]-\beta||^2\,d\alpha\,d\beta$, where we tacitly assume the integrals exist. In the absence of any precise knowledge of p, the prior over pairs of input and output images, the true expected loss must be approximated. If there is training data available, we may rely on the empirical risk, which is determined by substituting the empirical distribution of our observations for p, leading to the objective (see, for instance, [7])

$$L[T]=\frac{1}{N}\sum_{i=1}^{N}||T[\alpha_i]-\beta_i||^2\,. \tag{1}$$

In many a setting, the transformation T would be taken translation invariant. This is the situation we consider here as well. In fact, since we focus on a single linear mapping, T reduces to a simple linear convolution by means of a kernel $u:\mathbb{R}^d\to\mathbb{R}$. Equation (1) therefore simplifies to

$$L[u]=\frac{1}{N}\sum_{i=1}^{N}\int_{\mathbb{R}^d}|(\alpha_i*u)(x)-\beta_i(x)|^2\,dx=\frac{1}{N}\sum_{i=1}^{N}||\alpha_i*u-\beta_i||^2\,. \tag{2}$$

Denoting the Fourier transform by $\mathscr{F}$ or $\hat{}$, the optimal solution $u^\star$ to the above equation can be obtained as

$$u^\star=\mathscr{F}^{-1}\left[\frac{\sum_{i=1}^N\hat{\beta}_i\bar{\hat{\alpha}}_i}{\sum_{i=1}^N\hat{\alpha}_i\bar{\hat{\alpha}}_i}\right]\,. \tag{3}$$

This formulation, in fact, allows us to efficiently solve an image regression problem in, potentially, very high dimensional feature spaces. Similar observations have been made in the tracking literature (c.f. [4]).

To see that Equation (2) basically formulates a regular linear regression problem, note that $(\alpha * u)(x)$ equals $\langle \alpha(x - \cdot), u \rangle$, where $\alpha(x - \cdot)$ are the explanatory variables or the feature "vectors" indexed by the variable x and u can be interpreted as an estimate for the true regression parameters. Indeed, instead of using patches of limited size to capture the contextual information around every pixel location, this formulation basically takes the whole image (centralized around x) to be the input patch associated to every location x.

Ridge regression. Like in regular least squares, in the case of small samples, one may want to consider ridge regression to avoid overtraining [3, 7, 10]. This basically boils down to including a term in Equation (2) that penalizes the squared norm of u:

$$L[u] = \frac{1}{N} \sum_{i=1}^{N} ||\alpha_i * u - \beta_i||^2 + \lambda \|u\|^2 . \tag{4}$$

The optimal solution $u^\star$ can again be obtained efficiently by means of a minor alteration of the solution in Equation (3):

$$u^\star = \mathscr{F}^{-1} \left[\frac{\sum_{i=1}^{N} \hat{\beta}_i \bar{\hat{\alpha}}_i}{\sum_{i=1}^{N} \hat{\alpha}_i \bar{\hat{\alpha}}_i + N\lambda} \right] . \tag{5}$$

Therefore, also the ridge regression formulation to filter learning can be solved efficiently, even for very high dimensional feature spaces. Incidentally, note the similarity of this solution to those derived in Wiener filtering [13].

Regression problem size, an example. Consider the Brodatz images data set that we are going to experiment with later on. The set consists of 112 images of dimensions 640×640, which we take as the input images (some examples are shown in the top row of Figure 1). Let us assume that we have corresponding output images to all of the 112 Brodatz images, which are the original images corrupted by an unknown convolution filter and additive Gaussian noise. Finding a filter u that is optimal in the empirical least squares sense means that one would actually have to solve a linear regression problem in $640^2 \approx 400$k dimensions, coming from the patch size we consider and equals the size of the full image. The number of instances, we would base the learning on is $640^2 \times 112$, which is more than 45 million, as there are 640^2 locations per image and we have 112 images.

Solving this problem in the standard way by means of linear regression would, among others, mean that we have to invert a covariance matrix sized $640^2 \times 640^2$, which is sheer impossible. Because of the convolutional structure of the problem, however, explicit matrix inversion can be avoided and the computationally most demanding part in Equation (3) are the Fourier transformations. Relying on the fast Fourier transform, the necessary computations to find the more than 400 thousand weights (encoded through u) can be done in less than two seconds, even on a modest 2.40 Ghz laptop.

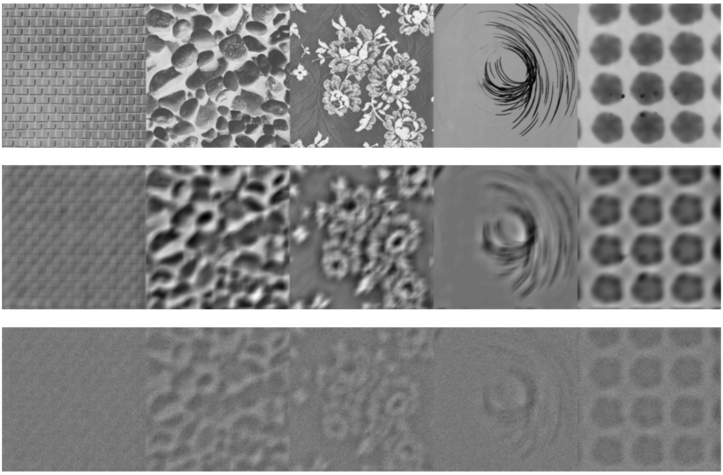

Figure 1: Top row: five example input images from the Brodatz data set. Middle row: corresponding examples (refer to Section 4) of initial output images obtained by convolving the input images in the top row with the kernel depicted in Figure 2. Bottom row: some of the noisy output images (SNR of -14.2 dB) used in the experiments.

3 Scale Regularization

As the previous section shows, using basic image processing techniques, for every pixel location in an image, we can actually include all other image values as context in its feature vector and still solve the high-dimensional regression problem efficiently. Nonetheless, there is a good reason why any classifier or regressor would restrict the extent of every filter to an area considerably smaller than the whole image. Estimations in such high-dimensional spaces easily leads to overtraining as there are too many free parameters to be estimated compared to the number of observations that may be available. It seems reasonable to assume, however, that it is more likely that the useful predictive information for a particular location in an image comes from locations nearby rather then pixel values far away.

The current way to exploit this kind of prior knowledge is by explicitly extracting patches of limited size around every pixel location and base the regression on these features only. Equivalently, the convolutional objective function in Equation (2) can be adapted to do the same by simply restricting the support of the filter u, i.e., one can minimize for Equation (2) under the constraint that $\operatorname{supp} u$ is an appropriate subset Ω of $\mathbb{R}^d$. Typically, Ω is just taken to be a square patch. Here, we suggest to take care of scale in a more proper way, stepping away from handcrafted filter sizes. Instead of restricting the influence of surrounding pixel values to a particular region explicitly, we propose to gradually suppress the influence of more and more distant pixel values by means of a scale-sensitive regularization term on the kernel u. We add this as a term to our original least squares objective function in Equation (2). In particular, we consider minimizing the following:

$$L[u] = \frac{1}{N}\sum_{i=1}^{N} ||\alpha_i * u - \beta_i||^2 + \lambda \int ||x||^2 u^2(x)\, dx\,, \qquad (6)$$

where $\lambda \geq 0$ controls the scale.

The primary characteristic of the regularizing term is that larger values for u should be discouraged the further away one gets from the center of the kernel. Clearly, various other formulations would have been possible, but the current suggestion has some appealing properties. Firstly, the polynomial $||\cdot||^2$ is rotationally invariant. Secondly, it is homogenous, so changing the unit in which we measure distance to the kernel center, can equivalently be accommodated by changing λ, i.e., the effect of substituting $||c\cdot||$ for $||\cdot||$, can also be achieved by substituting $c^2\lambda$ for λ. Still, such properties would hold for any choice of power, not only for the square. Choosing the square, however, leads in addition to a relatively easy to solve variational problem, which allows us to retain some of the computationally attractive properties of the original formulation in Equation (2).

Minimization. The choice of the regularization term in Equation (6) makes the minimization easy in the Fourier domain: using the derivation properties of the Fourier transform as well as Plancherel's theorem, one gets for an $L^2(\mathbb{R}^d)$ function u:

$$\|\nabla u\|^2 = \sum_{j=1}^{d}\int_{\mathbb{R}^d}|\partial_{x_j}u(x)|^2\,dx = 4\pi^2\sum_{j=1}^{d}\int_{\mathbb{R}^d}\xi_j^2\hat{u}^2(\xi)\,d\xi = 4\pi^2\int_{\mathbb{R}^d}||\xi||^2\hat{u}^2(\xi)\,d\xi\,. \tag{7}$$

Note that we use $||\cdot||$ to both denote the vectorial norm on $L^2(\mathbb{R}^d)^N$ and the scalar norm on $L^2(\mathbb{R}^d)$. Now, using the properties of the convolution and the Fourier transform, we can rewrite the criterion in Equation (6) as $L[\hat{u}] = \frac{1}{N}\sum_{i=1}^{N}||\hat{\alpha}_i\hat{u}-\hat{\beta}_i||^2 + \frac{\lambda}{4\pi^2}||\nabla\hat{u}||^2$, which is a Tikhonov regularization of the regression problem in the Fourier domain. By letting $a = (\hat{\alpha}_1,\ldots,\hat{\alpha}_N)^\top$ and $b = (\hat{\beta}_1,\ldots,\hat{\beta}_N)^\top$ (so every element in these vectors is a full image), we can rewrite it as

$$L[\hat{u}] = \frac{1}{N}||\hat{u}a-b||^2 + \frac{\lambda}{4\pi^2}||\nabla\hat{u}||^2. \tag{8}$$

Computing the first variation of Equation (8) gives the optimality condition

$$a^*(a\hat{u}-b) - \frac{N\lambda}{4\pi^2}\Delta\hat{u} = a^*a\hat{u} - a^*b - \frac{N\lambda}{4\pi^2}\Delta\hat{u} = 0\,, \tag{9}$$

where a^* denotes the Hermitian adjoint of a, $a^* = \bar{a}^\top$. Note that because the $\hat{\alpha}_i$ and $\hat{\beta}_i$ are Fourier transforms of real functions they are Hermitian, i.e., they satisfy the equation

$$\hat{f}(-\xi) = \bar{\hat{f}}(\xi)\,. \tag{10}$$

Of course the solution will be Hermitian, and therefore u, recovered by the inverse Fourier transform of $\hat{u}$, is real-valued, as expected.

More importantly from a computational perspective, note that $a^*a = \sum\alpha_i^*\alpha_i$ and $a^*b = \sum\alpha_i^*\beta_i$ are, next to the value of λ, the only inputs to the optimization one needs. The size of both a^*a and a^*b equals the original image size and does not depend on the number of training images. So no matter how many training images one uses, once we have a^*a and a^*b, the computational complexity of getting to a solution for Equation (9) remains the same.

Now, the actual numerical minimization for the 2-dimensional images in our experiments is carried out using a standard 5-points stencil for the Laplacian. We use the periodic boundary conditions as imposed by the discrete Fourier transform that we of course resort to in our experiments. The resulting system is solved by Jacobi relaxation and reads as follows:

$$\hat{u}_{ij}^{n+1} = \frac{(a^*b)_{ij} + \frac{N\lambda}{4\pi^2}\left(\hat{u}_{i+1j}^n + \hat{u}_{i-1j}^n + \hat{u}_{ij+1}^n + \hat{u}_{ij-1}^n\right)}{(a^*a)_{ij} + \frac{N\lambda}{\pi^2}} \tag{11}$$

(omitting boundary conditions). Though faster solvers are possible, the use of the Jacobi solver automatically enforces at each iteration the discrete counterpart of the Hermitian relation in Equation (10) and therefore $\hat{u}^n$ remains the Fourier transform of a real signal at each iteration. Finally note that in Equation (11), the regularizing effect of a positive λ can, in part, be seen back, as it keeps the denominator bounded away from zero.

4 Experimental Setup and Results

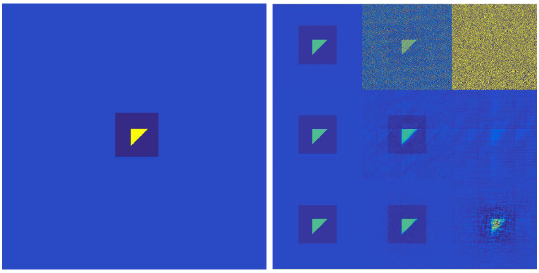

Figure 2: On the left: convolution kernel used to create output images from original Brodatz images. On the right, top row: central cropped part of the three filters estimated by means of unregularized regression: one filter for every one of the noise levels considered. Right, middle row: same as top row, but with approach based on optimally regularized ridge regressions. Right, bottom row: same for the scale-regularized formulation, in which the regularization parameter is taken to be 10^2, 10^4, and 10^6, respectively.

To demonstrate the potential of our scale-regularized filter learning over unregularized and ridge regression, we set up some elementary experiments on the Brodatz image collection. We take the 112 images in this database as our input images (see the top row in Figure 1) and corrupt them to create 112 matching 640×640 output images. In order to come to our corrupted images, we first construct a kernel to convolve the original images with. Here we already note that the asymmetric and flatly shaped filter that we are going to use leads to a relatively difficult behaving Fourier transform (e.g. strongly non-band limited, causes large high-frequency components etc.). Our optimal filter is therefore more challenging to estimate than if we would have used more regular, smoothly varying filters.

Figure 2 depicts this 640×640 filter: the lighter blue, the largest part of the image, takes on the value zero. Dark blue has value -1 and the yellow part takes on the value $\frac{9748}{861}$, which makes sure the filter integrates to zero. The filter's nonzero support is 103×103 pixels. Figure 1, middle row, gives the output images after convolving the corresponding input images in the top row of the same image. As the second and also final step, we add i.i.d. Gaussian noise to every output image. The signal to noise ratios (SNRs) we experiment with are 65.8 dB, 25.8 dB, and -14.2(!) dB. These noise levels are somewhat arbitrary, but it should be clear that if the outputs were noiseless, solving the regression without regularization would provide us with a perfect reconstruction of the original convolution kernel. The bottom row of Figure 1 displays the final noisy output images with the worst SNR of -14.2 dB, in which case we would expect the worst performance for the unregularized learning scheme.

We tested our method with learning set sizes of 1, 2, and 4 image pairs to get an impression of the behavior w.r.t. this aspect as well. The values of λ considered are 0 (no

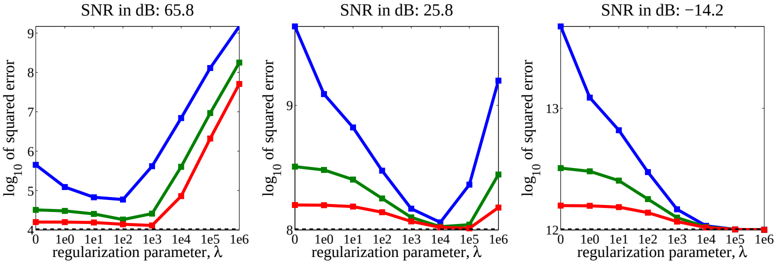

Figure 3: Plots of the squared errors obtained by the non-regularized and scale-regularized regression procedures for the three different noise levels. λ is on the horizontal axis, while the vertical axis shows the 10-log of the squared error. The blue lines are obtained when training on a single image pair; the green line uses two image pairs; the red line four. The black, dashed line provides the error's lower bound due to the intrinsic noise.

regularization), 10^0, 10^1, ..., 10^5, 10^6. We redid all experiments 25 times, every time with different training images, and report the averaged results.

In order to solve this task using the contemporary strategy of extracting patches, we would need patches no smaller than the nonzero support of the filter, which would mean that the feature dimensionality to deal with equals $103^2 = 10609$. Given that, even with just a single training image, we would be dealing with more than 400k patches, the regular approach becomes computationally burdensome quite rapidly. The methods we test, however, are computationally fairly easy to deal with.

Filtering results. Figure 3 plots the results for our regularization scheme in three subplots. We note that all differences with unregularized learning ($\lambda = 0$) are significant according to a paired t-test, never giving p-values higher than 10^{-3}. The best performing regularized filter typically achieves improvements of about an order of magnitude, especially in case the learning is based on one image only (the blue lines in the plots).

For the extreme case of just a single training pair, we also display some of the filters that have been inferred for the different noise levels. Figure 2, on the right, shows the unregularized filters in the top row. The optimal regularized versions for ridge regression and the optimal results of our scale-regularized approach can be found in the middle and bottom rows of that same image, respectively. To properly display these images, we decided to clip their values at twice the minimum and twice the maximum of the values attained by the original filter from Figure 2. For an SNR of 65.8 dB, all three procedures recover the original filter very well, though upon closer inspection the unregularized filter clearly is most noisy in its off-center parts. Also the ridge regressor displays some noise in these areas. The noisiness of the unregularized filter solution is also reflected in the inferior performance displayed in Figure 3.

When the SNR reaches −14.2 dB, the unregularized filter becomes excessively noisy and its values get clipped for over 98% of the pixels, while in the case of scale-regularization this is less than 0.23%. No clipping is necessary for the optimal ridge solution, as the regularization suppressed all values in all locations so strongly that they hardly differ from zero. This is also visually clear: while the scale-regularized solution, irrespective of the immense noise level, still resembles the original filter, the unregularized one has basically been reduced to noise and the ridge regressor returns an image that looks almost uniformly blue.

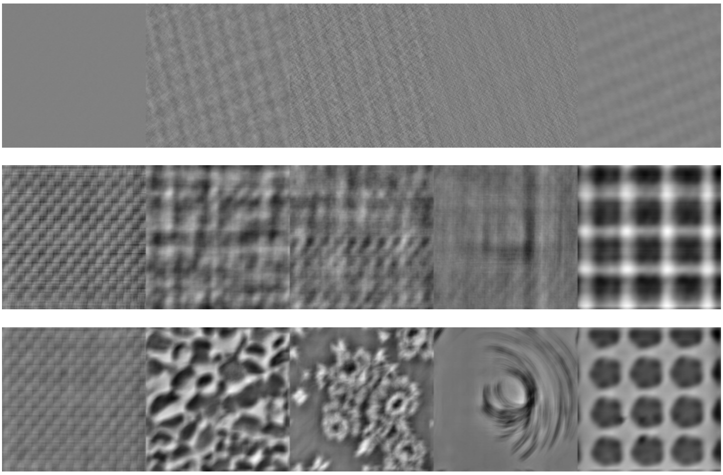

Figure 4: Top row: example prediction results of unregularized regression for the images in Figure 1 in the setting where the maximum noise level was employed and training is based on a single image. Middle row: same but now based on regression with standard, ridge-type regularization with optimal regularization parameter. Bottom row: prediction based on scale-regularized regression; λ was set to the optimal value 10^6.

Figures 4 shows these differences in another way, displaying the images obtained by convolving the input images from the top row in Figure 1 with the unregularized, ridge regularized, and scale-regularized kernels, respectively. While there is little but noise visible in the top row of Figure 4, the middle row vaguely reflects the noise-free output as we can find it in the middle row of Figure 1. The reader should appreciates the rather close resemblance of the reconstructed images in the bottom of the same figure—the ones based on scale regularization, to the same noiseless outputs in the middle of Figure 1. Yet another quantification of the excellent performance of scale regularization can be found in Table 1, which reports on the relative improvement regarding the squared error. As we know the levels of the added noise, we have lower bounds on the errors achievable. Taking that error to be 0 and the measured error for the unregularized and ridge regression approach to be 1, respectively, we can express a relative improvement that scale regularization can achieve compared to these other methods: the closer to zero the number is, the better.

Table 1: Relative improvement in average squared error of scale regularization as compared to the unregularized approach (on the left) and the optimal ridge regressor (on the right) for the three different noise levels and the three different training set sizes. The value is 0 if scale regularization is perfect and 1 if it performs on par with the method compared to.

	unregularized			ridge regression		
	# training images					
	1	2	4	1	2	4
65.8 dB	0.11	0.003	0.0001	0.18	0.04	0.12
25.8 dB	0.37	0.028	0.0006	0.38	0.12	0.12
-14.2 dB	0.51	0.051	0.0019	0.51	0.13	0.18

5 Discussion and Conclusion

The approach is rather elementary, yet the results are striking. To solve more complex, real-world filtering problems, however, we need more complex learners. A basic idea of neural networks is, in fact, that one can build arbitrarily complex regression and classification schemes out of more basic building blocks [1, 2]. What is essential in this, however, is that we do not have to limit ourselves to linear transformation of the data. The next important step in this research should therefore investigate how to incorporate a so-called activation function, which transforms the filter outputs in a nonlinear way, into our setup. Introducing this nonlinearity will take us even further away from the simple-to-solve objective functions in Equations (2) and (4) and it is as yet unknown to what extent computational efficiency can be retained.

Concerning that efficiency, even though the chosen relaxation scheme is sufficiently fast for the image sizes we are currently dealing with, optimization through our simple relaxation scheme does take around 5 minutes (on that same modest 2.40 GHz laptop) when dealing with the computationally most intensive problem, which is for the large-noise case. Given that we have to estimate about half a million parameters this seems very reasonable as we do not rely on highly dedicated GPU implementations and the like. Note that our scheme would hardly suffer from an increase in training examples. It is just the computation of a^*a and a^*b, part of the preprocessing, that takes longer. Nevertheless, further speed up is definitely of interest if we move to even larger images. For this, multigrid schemes seem a natural way to advance as they can be designed to respect the multiresolutional nature of the image data. See, for instance, [21] for general background and theory and [5, 14] for some applications of fast bidirectional schemes within computer vision.

Of course, our experiments quite fit the assumptions underlying the scale-regularization. In particular, the filter that corrupts the input images, is fairly centralized. The a priori expectation that the most important image information is probably near the image location for which the prediction is being made seems reasonable however. Nonetheless, at times, the desired solution may be a non-centralized filter. Also in this case, we would still expect our approach to work best in general, even though the differences in performance may become less. One should realize that if extreme off-center filters are indeed necessary, any method (including deep nets) would have difficulties with this task.

To conclude: we devised a novel scheme that tackles the problem of scale in the elementary learning setting of inferring a linear filter from a set of input-output pairs. Our approach does not rely on any a priori restriction on the context size taken into account when performing the regression, but it incorporates a way of regulating scale by means of an added scale-regularization term that can be tuned. The approach is rather elementary, yet the results are striking, showing excellent performance for our approach even with SNRs as low as -14.2 dB. Performance could also have been measured in terms of cross-correlation, PSNR, or L_1 distance, but this does not lead to any difference in our conclusions: our approach is clearly to be preferred.

References

[1] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE TPAMI*, 35(8):1798–1828, 2013.

[2] C.M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, 1995.

[3] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] D.S. Bolme, J.R. Beveridge, B.A. Draper, and Y.M. Lui. Visual object tracking using adaptive correlation filters. In *IEEE Conference on CVPR*, pages 2544–2550, 2010.

[5] Andrés Bruhn, Joachim Weickert, Timo Kohlberger, and Christoph Schnörr. A multigrid platform for real-time motion computation with discontinuity-preserving variational methods. *IJCV*, 70(3):257–277, 2006.

[6] H.C. Burger, C.J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with BM3D? In *Proceedings CVPR, 2012*, pages 2392–2399, 2012.

[7] Theodoros Evgeniou, Massimiliano Pontil, and Tomaso Poggio. Statistical learning theory: A primer. *IJCV*, 38(1):9–13, 2000.

[8] K.S. Fu, D.A. Landgrebe, and T.L. Phillips. Information processing of remotely sensed agricultural data. *Proceedings of the IEEE*, 57(4):639–653, 1969.

[9] B.M. Haar Romeny. *Front-End Vision and Multi-Scale Image Analysis*. Springer, 2003.

[10] T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Verlag, 2001.

[11] F. Holdermann, M. Bohner, B. Bargel, and H. Kazmierczak. Review of automatic image processing. *Photogrammetria*, 34(6):225–258, 1978.

[12] Z. Hou and T.S. Koh. Image denoising using robust regression. *IEEE Signal Processing Letters*, 11(2):243–246, 2004.

[13] Anil K. Jain. *Fundamentals of digital signal processing*. Prentice-Hall, 1989.

[14] Ron Kimmel and Irad Yavneh. An algebraic multigrid approach for image analysis. *SIAM Journal on Scientific Computing*, 24(4):1218–1231, 2003.

[15] Jan J Koenderink. The structure of images. *Biological cybernetics*, 50(5):363–370, 1984.

[16] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[17] Tony Lindeberg. *Scale-space theory in computer vision*. Springer, 1993.

[18] M. Loog, B. van Ginneken, and A.M.R. Schilham. Filter learning: application to suppression of bony structures from chest radiographs. *Medical Image Anal.*, 10(6):826–840, 2006.

[19] Marco Loog and François Lauze. Scale-regularized filter learning. *arXiv preprint 1707.02813*, 2017.

[20] K. Suzuki, H. Abe, H. MacMahon, and K. Doi. Image-processing technique for suppressing ribs in chest radiographs by means of massive training artificial neural network (MTANN). *IEEE Transactions on Medical Imaging*, 25(4):406–416, 2006.

[21] Ulrich Trottenberg, Cornelius W. Oosterlee, and Anton Schuller. *Multigrid*. Academic Press, 2000.