Supervised Learning of Image Restoration with Convolutional Networks

Viren Jain[1], Joseph F. Murray[1,2], Fabian Roth[1,2], Srinivas Turaga[1], Valentin Zhigulin[1,2], Kevin L. Briggman[3], Moritz N. Helmstaedter[3], Winfried Denk[3], and H. Sebastian Seung[1,2]

[1]Brain & Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA, USA

[2]Howard Hughes Medical Institute, Cambridge, MA, USA

[3]Biomedical Optics, Max Planck Institute for Medical Research, Heidelberg, Germany

Abstract

Convolutional networks have achieved a great deal of success in high-level vision problems such as object recognition. Here we show that they can also be used as a general method for low-level image processing. As an example of our approach, convolutional networks are trained using gradient learning to solve the problem of restoring noisy or degraded images. For our training data, we have used electron microscopic images of neural circuitry with ground truth restorations provided by human experts. On this dataset, Markov random field (MRF), conditional random field (CRF), and anisotropic diffusion algorithms perform about the same as simple thresholding, but superior performance is obtained with a convolutional network containing over 34,000 adjustable parameters. When restored by this convolutional network, the images are clean enough to be used for segmentation, whereas the other approaches fail in this respect. We do not believe that convolutional networks are fundamentally superior to MRFs as a representation for image processing algorithms. On the contrary, the two approaches are closely related. But in practice, it is possible to train complex convolutional networks, while even simple MRF models are hindered by problems with Bayesian learning and inference procedures. Our results suggest that high model complexity is the single most important factor for good performance, and this is possible with convolutional networks.

1. Introduction

Convolutional networks have been used with great success for high-level visual tasks such as object recognition [15, 16]. In this paper, we argue that they can also can be used as a general method for low-level image processing. Our approach will be illustrated in the context of image restoration, the problem of recovering a "true" image from an observed image that has been corrupted by some noisy process. The result of image restoration can be used to aid human interpretation, or as the input to other computer vision tasks such as recognition or segmentation.

We test the performance of our approach using a database of electron microscopic (EM) images of neural tissue [5]. It is natural to divide the image pixels into two classes, intracellular ("in") or extracellular ("out"), which is a binary image restoration problem. In order to establish ground truth, as well as provide data for training our convolutional networks, humans manually restored the images by drawing boundaries between "in" and "out" regions. To provide a performance baseline, we first tried restoring the images using simple thresholding, and anisotropic diffusion followed by thresholding. Both methods yielded roughly the same performance.

A convolutional network was then trained on the dataset. Although the network architecture was very complex, containing over 34,000 adjustable parameters, we were able to train it using gradient learning. After training, the network provided significantly more accurate reconstructions on the test set than did thresholding.

For comparison, we also trained a Markov random field (MRF) model on the same dataset. When first introduced to image processing, MRF models were fairly simple, with just a few parameters that were adjusted by hand [9]. Recently there has been significant interest in training more sophisticated MRF models using machine learning methods [24]. Drawing on this research, we trained our MRF using the pseudolikelihood algorithm to generate the noisy training image and the restored training image. Its performance on the test set was not significantly better than simple thresholding.

Some researchers have argued that better results can be obtained from MRF models by training them discriminatively, i.e. by directly optimizing the transformation from noisy to restored images [11, 13]. This is called the conditional random field (CRF) approach. We also trained a CRF on our dataset, but its performance on the test set was no better than the MRF.

To understand the failure of the MRF/CRF approach, we also trained a convolutional network with a very simple architecture. This network could be viewed as mean field inference for a CRF. We added more constraints to the architecture of the network to make it match the CRF as closely as possible. When crippled in this way, the performance of the simple network matched that of the CRF and thresholding.

We suggest that convolutional networks and CRFs should give equivalent performance in principle, as long as the models have equivalent complexity. But our empirical results suggest that a more complex model does better than simpler one. One might worry that a highly complex model should suffer from overfitting of its numerous parameters. However, even with over 34,000 free parameters, our complex convolutional network does not seem to overfit—the gap between training and test error is fairly small. Therefore, high model complexity appears to be an important prerequisite for good image restoration.

While convolutional networks and MRF/CRF models are roughly equivalent in principle, in practice we feel that the former approach is superior because highly complex models can be trained. In the MRF/CRF approach, even training simple models is problematic, because of technical difficulties surrounding Bayesian learning and inference procedures.

We further tested the quality of restorations by using them to generate image segmentations. Since each object within our data is a region of intracellular space separated from all other objects by some extracellular space, a highly clean binary restoration should in principle be sufficient for accurate image segmentation. We demonstrate that the restorations given by convolutional networks can be used to segment the EM images, while the other methods produce restorations that are so error-prone as to be essentially useless for segmentation, at least by naive algorithms.

The present work is related to the extensive literature on applications of neural networks to image processing, which has been reviewed by Egmont-Petersen et al [7]. In this literature, a multilayer perceptron is applied to patches of the input image. Our work is distinct from this literature because our networks are convolutional.

As mentioned previously, we were inspired by previous research on convolutional networks applied to object recognition, as well as one previous study that used convolutional networks to label and segment regions in microscopic imagery [15, 16, 20]. In addition to convolutions, all of these networks also included subsampling, which produced an output representation of much lower resolution than the input image. Subsampling is an important strategy in object recognition, where it helps achieve invariance to distortions of the visual image by discarding positional information about image features and details. But many image processing applications *require* precise positional information. The segmentation of fine branches of neurons in EM images is a good example. Therefore our convolutional networks do not include subsampling, and we expect that this will be appropriate for many other image processing applications. Indeed, for the segmentations described above, we have introduced the use of *supersampling* in our networks to increase the resolution of the output image. This is critical because the spacing between objects sometimes narrows to less than one pixel of the input image.

A further novel aspect of our work is that we emphasize the similarities between convolutional networks and MRF/CRF models, and conduct a comparative study of the two approaches.

2. Restoration of nanoscale brain images

The automated analysis of electron microscopic (EM) images of brain tissue represents an intriguing opportunity for computer vision. The "wiring diagrams" of nervous systems have remained largely unknown due to the lack of imaging techniques of appropriate reliability and resolution. Structures such as axons can be as small as 100 nm, while extending for centimeters across a brain. A recently developed imaging technique, Serial Block Face Scanning Electron Microscopy (SBF-SEM), is capable of generating nanoscale images over a field of view spanning up to potentially 1 mm, which is enough to begin reconstructing structures of biological interest [5, 6]. A block of tissue is alternately imaged and sliced, yielding a 3d image with a voxel resolution of 20-30nm in all three directions. Compared to other serial-section imaging techniques, SBF-SEM generates stacks of 2d images in which successive image slices are extremely thin and very well aligned. However, successful interpretation of this data will require accurate and highly automated methods of image processing: a cube of tissue 300 microns on a side could generate a trillion or more voxels of data, making manual analysis so time consuming as to be impractical.

Restoration of SBF-SEM images poses a difficult computer vision problem due to several issues: (1) small size of certain structures (few pixels in diameter at certain points), (2) dense packing of structures, (3) local noise and ambiguity caused by imperfections in the staining and imaging, and (4) variety of physical morphologies.

For the problem of neural circuit reconstruction, a successful image processing approach will have to overcome such issues with $extremely$ high accuracy in order to maintain an adequate level of automation. Our basic approach is to restore the images by classifying each voxel as being inside or outside a cell. Since cells are usually separated from each other by some amount of "outside" space, an accurate restoration can provide a segmentation of the dataset as well.

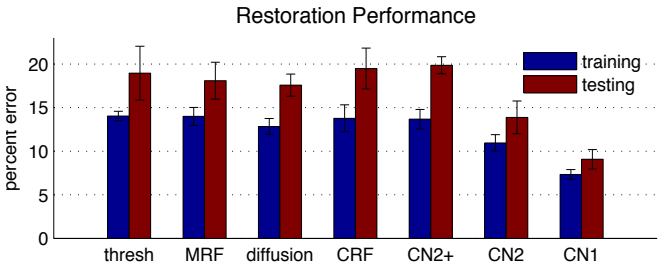

Figure 1. Results on voxel-wise restoration accuracy. The training set has 0.5 million voxels, and the test set has 1.3 million voxels.

3. Creation of the training and test sets

A volume of rabbit retina was imaged at 26.2x26.2x50 nm^3 resolution using the SBF-SEM technique. We used a 20.96x15.72x5μm^3 subset of this volume, yielding a 800x600x100 voxel dataset. The tissue was stained in a manner designed to make image analysis easier for both humans and computers, by attenuating details within the image while enhancing intercellular space [5]. The boundaries of neurons were traced manually by two humans. They provided both object boundaries and consistent object identities from one slice to the next. The tracers were instructed to be careful in drawing boundaries, and generated training data at a rate of roughly 30,000 voxels/hour. An example of an image and its tracing are shown in Figure 3. Tracings were captured at a spatial resolution much higher than the image resolution (the humans traced interpolated images) and were "restored" to an inside/outside binary classification using a point-in-polygon method. Two restorations were generated: one at the same voxel resolution as the images, and one at twice the resolution of the images in each dimension (8 times the number of voxels).

The labeled data was split into two regions: a 0.5 megavoxel "training set" volume that was used to optimize each algorithm, and a 1.3 megavoxel testing region that was used to quantitatively evaluate performance. 75.6% of the voxels within a labeled bounding box were classified as 'inside'. The two human labelings disagreed on the classification of 9.38% voxels within a 1 megavoxel region that was traced by both humans. While this may sound significant, the great majority of inter-annatator differences were found in variations in the exact placement of boundaries, rather than in disagreements over the true shape and identity of objects (i.e., segmentation).

4. Thresholding sets baseline performance

Since the "in" regions are usually light and the "out" regions are usually dark, it is natural to attempt binary restoration by simply thresholding the image. The noisy training image was thresholded at various levels to produce a binary restoration, which was compared with the true restoration provided by a human expert. The value of the threshold minimizing error on the training set was found. Then the noisy test image was thresholded at this value and the result was compared with the human restoration. As shown in Figure 1, thresholding yielded a training error of 14.03% and a test error of 18.95%.

An obvious way of improving the simple thresholding algorithm is to preprocess the noisy image by smoothing it. This can be done by linear filtering, but there are also more powerful "edge-preserving" nonlinear techniques that smooth differences in image intensities except at regions of very strong discontinuity. We used a 3d version of the Perona-Malik anisotropic diffusion algorithm [22]. Binary restorations were produced by thresholding the diffused images. The threshold, along with several parameters of the diffusion algorithm, were optimized by grid search on the training set (see supplementary material for additional details).

Thresholding the diffused images did not yield significantly better results than thresholding the raw images. This may be due to the fact that the inside regions of cells were not of uniform intensity. Although "in" regions were generally lighter than "out" regions, some "in" regions were still fairly dark. Anisotropic diffusion smoothed the "in" regions but did not change their average intensity. Therefore some "in" regions still fell below threshold and were erroneously classified as "out."

5. A complex convolutional network outperforms simple thresholding

A convolutional network alternates between linear filtering and nonlinear transformations to produce a transformed version of some input. The architecture consists of an input layer that encodes one or more input images, an output layer that encodes one or more output images, and various intermediate layers with "hidden" images that contain the internal computations and representations of the algorithm. Each layer receives input from only the previous layer. The activity of feature map a in layer k is given by

$$I_a^k = f\left(\sum_b w_{ab}^k \otimes I_b^{k-1} - \theta_a^k\right) \quad (1)$$

where the I_b^{k-1} are feature maps in the previous layer that provide input to I_a^k, and $\otimes$ denotes the convolution operation. The function f is a smooth nonlinearity; we use the sigmoid $f(x) = 1/(1+e^{-x})$. There is also a threshold parameter θ_a^k associated with each feature map.

We trained a convolutional network of the architecture shown in Figure 2 on our dataset of EM images. All filters and biases in the network were optimized using an online version of the backpropagation algorithm (see supplementary material for further details). We used the cross-entropy

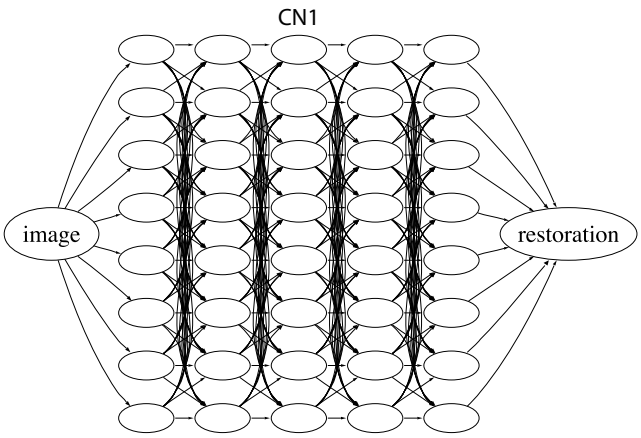

Figure 2. Complex convolutional network architecture (CN1). Between the input image and output image are 5 hidden layers, each containing 8 images. All arrows represent adjustable $5 \times 5 \times 5$ filters, and each node has an adjustable bias parameter. Because there are six convolutions between the input and output, each voxel in the output image is a function of a $25 \times 25 \times 25$ patch of the input image. The total number of free parameters is 34,041.

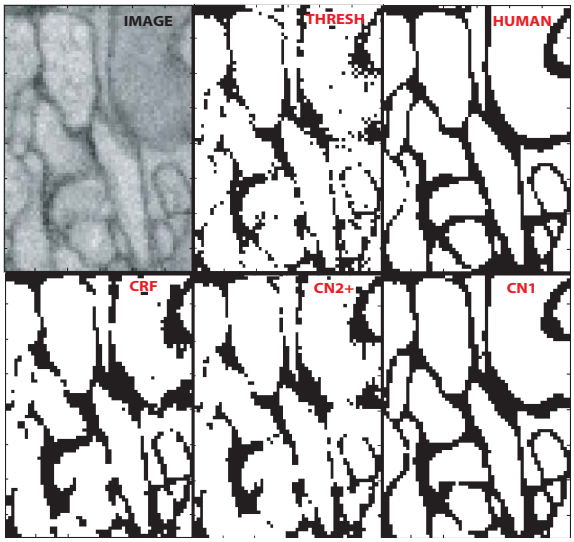

Figure 3. Example of restoration results from the test set (generalization performance). Although only a 2d region is shown here, all methods except thresholding utilize 3d computations.

cost function as our optimization criterion. The network was trained for 100 epochs, where one epoch is defined as exposure to the entire training set of 0.5 megavoxels. This took 48.7 hrs, using a parallel implementation running on 12 cpu-cores operating at 3Ghz. While this may seem significant, it should be noted that our network is a highly complex model by the standards of the image processing field, having over 34,000 adjustable parameters.

Since the output of the convolutional network is analog, it was thresholded to produce a binary image restoration. A threshold value of 0.51 was chosen by optimizing restoration on the training set. The training and test errors are shown in Figure 1. The convolutional network roughly halved the error rate of simple thresholding or thresholded anisotropic diffusion, which was a statistically significant improvement. Visual inspection also shows obvious improvement (Figure 3).

6. MRF performance is similar to thresholding

A conventional approach to image restoration is Bayesian inference and learning using Markov random field (MRF) models [17]. We also applied this approach to our dataset. Let $y = \{y_i\}$ denote the observed image and $x = \{x_i\}$ denote the true image, where i ranges over all voxel locations. The joint density $p(x, y)$ is specified by a prior distribution $p(x)$ over the true image x, and a noise model $p(y|x)$ that describes how noisy images are generated from true images. We considered a prior of the MRF form,

$$p(x) \propto e^{\frac{1}{2}\sum_i x_i (w \otimes x)_i + \sum_i b x_i} \qquad (2)$$

where the x_i are binary variables taking on the values ± 1. Since the MRF is assumed to be translation invariant and local, the interactions between voxels are described by a filter w of size much smaller than the image. We used a $5 \times 5 \times 5$ filter, which permits interactions up 2 pixels away in all directions. The filter is invariant under reflections, $w_i = w_{-i}$, and its central voxel is assumed to be zero (no self-interaction). We used the Gaussian noise model

$$p(y_i|x) = p(y_i|x_i) \propto \sum_{l=\{-1,1\}} \delta(x_i, l) e^{-\frac{(y_i - \xi_l)^2}{2\sigma^2}} \qquad (3)$$

where $\xi_{\pm 1}$ are the means of "in" and "out" voxels, and σ is their standard deviation.

The model was trained on the images by maximizing

$$\log p(x, y) = \log p(y|x) + \log p(x) \qquad (4)$$

with respect to the adjustable parameters, where y was the noisy SBF-SEM image and x was the restored image from the training set. Since the two terms in the sum do not share parameters, they can be optimized independently. The parameters $\xi_{\pm 1}$and σ of the noise model were found by calculating the mean and standard deviation for "in" and "out" voxels of the training data. The parameters of the MRF prior $p(x)$ were determined by pseudo-likelihood learning, which has become popular in MRF research [21, 17, 2, 11].

Once these parameters were trained, we attempted to restore the noisy test image by maximizing the posterior distribution $p(x|y)$ with respect to x, a procedure known as MAP inference. The posterior distribution $p(x|y)$ takes the same form as Eq. (2), except that b is replaced by a quantity that depends on the noisy image y.

For MAP inference, we first attempted simulated annealing via MCMC sampling on the posterior distribution. However, because of its slow convergence, we could not be sure whether the global optimum was attained. Better results were obtained with the min-cut algorithm of Boykov and Kolmogorov (BK) [3, 4], which has become a popular alternative to sampling. Although the BK min-cut algorithm is fast, it is rather memory-intensive, storing an adjacency graph which in our case was 125 times the size of the already large image. Therefore, the bounding box of the test image was split into four overlapping regions, each of which contained roughly 1.6 million voxels. Then min-cut was run on each region, and the results were stitched together. A further complication is that min-cut is only guaranteed to give the global optimum when the interaction strengths w are nonnegative, but pseudolikelihood training yielded interaction strengths of mixed sign. Nevertheless, min-cut provided superior results to simulated annealing in practice, so its results are shown in Figure 1.

The MRF model did not perform significantly better than simple thresholding. We can only speculate about the reasons for the failure of sophisticated Bayesian methods. First, the pseudolikelihood training procedure might have been problematic. Perhaps true maximum likelihood learning of the MRF prior $p(x)$ would yield better performance. In principle this could be done using the Boltzmann machine learning algorithm [1], but this would have required MCMC sampling, an extremely time-consuming learning procedure on a 0.5 million voxel training region. Second, the min-cut inference procedure might not have given the global optimum, since the interaction strengths were of mixed sign. Third, it might have been misguided to maximize the joint probability $p(x, y)$, a strategy known as generative training. The desired computational task is not generating pairs of noisy and restored images, but rather to transform a noisy image into a restored image. Therefore, it might be better to directly maximize the probability $p(x|y)$, a strategy known as discriminative training. In fact, there is a great deal of current research on MRF models motivated by the belief that discriminative training is superior to generative training.

7. CRF performance is similar to thresholding

Discriminatively trained MRFs are sometimes called conditional random fields (CRFs) [14, 13, 11, 8]. Some have argued that CRFs are superior to MRFs for image analysis [11, 13, 23]. To test this claim, we trained a CRF model of the form

$$p(x|y) \propto e^{\left(\beta\left[\frac{1}{2}\sum_i x_i(w\otimes x)_i + \sum_i x_i(y_i+b)\right]\right)} \qquad (5)$$

We attempted both pseudolikelihood and zero temperature Boltzmann machine learning, which is valid in the limit $\beta \to \infty$. The latter gave superior results. In this approach, min-cut was used to find the global optimum x^* of the CRF at each iteration. Then the contrastive update

$$\Delta w_j \propto \sum_i (x_{i+j}x_i - x^*_{i+j}x^*_i) \qquad (6)$$

was made, where x denotes the human-restored image. A similar update rule was used for the offset parameter b in Eq. (5). Since min-cut is only guaranteed to work for nonnegative filters, whenever the contrastive update drove a filter coefficient negative, it was reset to a value of zero. Once the training had converged, min-cut was used to restore the test image. Again, the test error was not significantly better than that of simple thresholding, as shown in Figure 1.

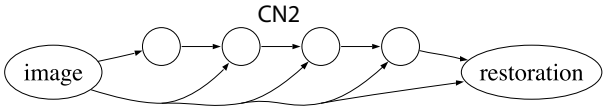

Figure 4. Simple convolutional network with architecture matched to the MRF model used in this paper. A single $5 \times 5 \times 5$ filter and bias are repeated five times. Each layer receives input from the raw image, from which an offset parameter is subtracted. The filter, the bias, and the offset give 127 total adjustable parameters.

8. A simple convolutional network is similar to a CRF

We were surprised that neither the MRF nor CRF models were significantly better than thresholding. How could these empirical results be explained? To explore this issue, we decided to train a convolutional network that was much simpler than the one of Figure 2. Figure 4 depicts the architecture of CN2. We chose the architecture because it can be viewed as mean field inference on the CRF of Eq. (5). Mean field inference is an alternative to MAP inference that yields an approximation to the expectation value of x_i under the CRF. This expectation is then thresholded to obtain a binary value [17]. In the mean field approximation the expectation value μ_i satisfies the equation $\mu_i = \tanh\left(\beta\left[(w \otimes \mu)_i + y_i + b\right]\right)$. A naive method of solving these equations is to iterate them using the dynamics

$$\mu_i(t+1) = \tanh\left(\beta\left[(w \otimes \mu(t))_i + y_i + b\right]\right) \qquad (7)$$

which, for a finite number of iterations, is precisely the convolutional network shown in Figure 4. Each iteration t corresponds to one layer in the network. A single filter w is shared by all layers, and the image serves as input to all layers. Additionally, we restricted the weights in w to be positive, to match the constraint in the min-cut based Boltzmann learning procedure employed for the CRF.

The performance of this network, called CN2$^+$, was nearly identical to the CRF in performance (Figure 1).

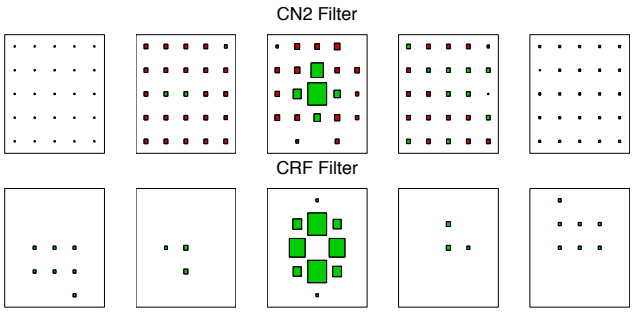

Figure 5. Filters learned by CN2 and the CRF. Each box displays one layer of weights. Green and red boxes signify positive and negative weights, respectively, while size indicates strength. Our results show that the negative surround in CN2 is important for good image restoration. Both the CRF and CN2$^+$filter were constained to be nonnegative, which yielded poor performance.

When inspected visually, the CN2$^+$and CRF restorations were similar (Figure 3). This is consistent with the idea that convolutional networks and CRFs are closely related to each other. When the models are exactly matched, they should yield roughly the same performance.

But we hypothesized that a more complex model should yield better performance than a simpler model. To test this idea, we relaxed the non-negativity constraint of CN2$^+$. This network, called CN2, was significantly better than thresholding, the MRF, and the CRF, but not as good as the complex CN1 network of Figure 2. The weights of the CN2 and the CRF are compared in Figure 5. CN2 has a positive center and a negative surround, suggesting that CN2 outperforms CN2$^+$ and the CRF because of its negative filter coefficients. This example shows that a seemingly trivial increase in the representational capability of a model can lend it superior performance.

9. Performance differences are much more severe when restorations are segmented

The nature of SBF-SEM images enables a simple approach to image segmentation based on binary image restoration. We tested this by restoring the images and then using a connected components algorithm to produce distinct image domains. Although more sophisticated segmentation algorithms might be considered, this simple procedure establishes a lower-bound on relative segmentation performance between the restorations.

A complication in this strategy arises when two objects are so close together that a binary restoration contains no boundary between them. In this case, even a competely accurate restoration may merge the two objects under a connected components criteria. Thus, we generate restorations at a *super-sampled* resolution twice that of the original images. If a human expert has decided that two objects are distinct despite the lack of a local boundary, the super-sampled representation allows enough room in the image space for there to be out-voxels between the objects. This was confirmed by segmenting the supersampled human restorations using the connected components criteria.

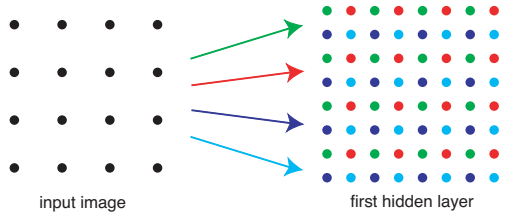

Figure 6. The first layer of CN3, a super-resolution convolutional network, produces a $2\times$ oversampled restoration relative to the input image by convolving 8 filters on 1 location, which are interleaved to form a single, oversampled image. This is illustrated here for the 2d situation, in which there are 4 filters for 1 location.

A convolutional network, CN3, was designed to produce supersampled restorations from the input images. The first layer of the network performs an upsampling by having 8 filters that each look at the same location in the input image, but output a different voxel within the $2 \times 2 \times 2$ cube that each input position now generates (Figure 6). Upsampling is then followed by an additional 4 layers of processing.

For the MRF and diffusion results, we manually upsampled the images using a linear interpolation scheme (other interpolation methods were also tested). However, the simplicity of convolutional networks allows us to easily *learn* the transformation from a "$1\times$" image to a "$2\times$" upsampled restoration.

The results clearly demonstrate the benefit of the convolutional network approach (Figure 7). The restoration from MRF and diffusion based methods contained many low-level errors, particularly in difficult image locations at which two objects were likely to merge together or one object was likely to break apart. Consequently they produced segmentations that were basically useless, even on the *training* set. The thresholded diffusion segmentation lacks many objects and the shapes of those that are present are severely distorted. As for the CRF, almost all objects were merged together into one large blob, due to inadequately preserved object boundaries in the CRF restoration. In contrast, the convolutional network restoration has far fewer errors and thus the segmentation is far more reliable, although not perfect. A video rendering of the segmentation results from CN3 can be found in the supplementary material.

10. Discussion

From our comparison of convolutional networks with MRF methods, several lessons were learned.

Convexity comes at the cost of representational power. In general, the problem of MAP inference for the MRF model (2) is an NP-hard combinatorial optimization. But for the special case of nonnegative interaction strengths, MAP inference is equivalent to a network flow problem, which is a convex optimization [10]. This realization has led to a great deal of exciting progress in the form of min-cut/max-flow algorithms for MAP inference [3, 4]. However, researchers may have overlooked the fact that the nonnegativity constraint might compromise the representational capability of their MRF models. In our empirical tests, the CRF model performed no better than the naive algorithm of thresholding, and worse than CN2, which can be viewed as an approximate inference algorithm for the CRF. The reason for the inferior performance of the CRF appears to be the nonnegativity constraint, which was imposed as a requirement for the min-cut algorithm. Our evidence for this comes from the fact that the performance of CN2$^+$,which was just like CN2 but with a nonnegative filter, dropped to the same level as the CRF and thresholding. Although negative interaction strengths are important for good performance (Figure 5), they are not allowed if convex optimization methods are used. More generally, while researchers may be attracted by the prospect of efficient algorithms for convex optimization, they should also be wary that convexity could come at a cost.

Pseudo-likelihood can be a poor approximation True maximum likelihood learning for MRF models depends on MCMC sampling methods. Because these methods can be time-consuming, researchers have looked for approximate methods that are faster, such as the popular pseudolikelihood (PL). But when we used PL to train a CRF, the results were so poor that they were not worth reporting. This is likely because the PL procedure trains the CRF to predict the value of a single output voxel from the values of other output voxels. Evidently this task is not so relevant for predicting output voxels based on the input image. There are other approximate learning procedures, such as contrastive divergence [21, 12], but they are not guaranteed to give good results either. For the simple MRF model (2) one can imagine that MCMC sampling methods would eventually work, given enough computational time. But ML training of an MRF with complexity comparable to CN1 (Figure 2) would be even more difficult. One would be forced to use approximations of questionable accuracy, much like pseudolikelihood. In short, Bayesian inference and learning procedures are beset with technical difficulties. One can argue about whether it is worth putting in the effort to surmount these difficulties, but it is difficult to deny that they exist.

Discriminative training is not always better than generative training In our particular application, a discriminatively trained CRF did not give better results than a generatively trained MRF; both were about the same as thresholding. A possible explanation is that our CRF/MRF model is such an impoverished representation that it does not matter whether discriminative or generative training is used (though it should be noted that the use of a learned $5 \times 5 \times 5$ filter makes our CRF/MRF model richer than many studied in the literature). Perhaps if a more complex CRF/MRF model were trained, there would be more of a difference between discriminative and generative training, but this speculation remains untested.

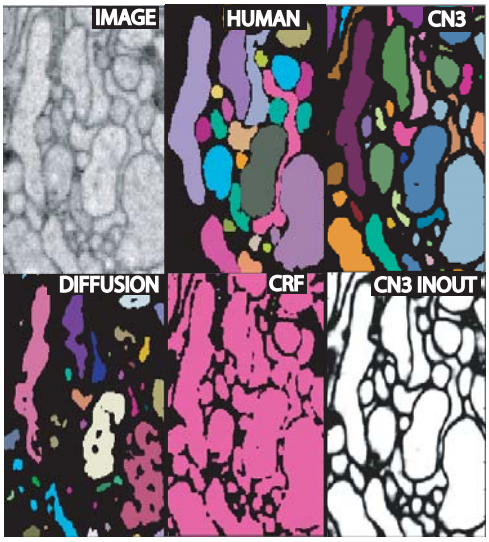

Figure 7. Example of segmentation results on the test set. Diffusion and CRF segmentations are poor due to many errors in the restoration. CN3's superior output ("CN3 inout" shows pre-thresholded restoration) provides a more reliable segmentation.

Convolutional networks avoid many of the technical problems of MRFs As a representation for image processing algorithms, convolutional networks have many of the same virtues as MRFs. Mathematically, the two approaches are closely related: certain convolutional networks can be viewed as mean field inference for discriminatively trained MRFs. However, convolutional networks avoid the technical problems of Bayesian learning and inference that were described above. The gradient of the objective function for learning can be calculated *exactly*, while Bayesian methods of calculating the gradient of MRF likelihoods rely on approximations.

Convolutional network learning is founded not on Bayesian principles but rather on the principle of empirical error minimization [25]. This principle is simple and direct: find a member of the parametrized function class defined by convolutional networks of a given architecture by minimizing error on the training set.

The primary drawback of empirical error minimization

is the requirement of databases with labeled examples. Creation of such databases may require substantial labor, particularly in image processing applications. However, as the goal of many low-level vision algorithms lack a robust mathematical specification, the creation of labeled datasets may be the only way to rigorously evaluate and optimize algorithms [18]. Moreover, recent advances in unsupervised learning in neural networks may dramatically reduce the amount of labeled data that is required to achieve good generalization performance [12, 19].

A convolutional network makes use of context to do image processing As mentioned in the introduction, our work is closely related to neural networks that operate on image patches, which have been studied extensively [7]. A convolutional network can be regarded either as a generalization or as a special case of an image patch network. Both viewpoints will be described briefly here, starting with the former.

The filters of CN1 were all 5^3 (Figure 2). If the filters were reduced in size to 1^3 after the first 5^3 layer, then CN1 would be equivalent to a neural network that is applied to each 5^3 image patch. This means that CN1 can be seen as a generalization of an image patch network.

But in fact, each 5^3 convolution in CN1 increases the size of the input patch that is "seen" by a voxel. In the final output layer, a voxel depends on a 25^3 patch of the input image, because of the six intervening convolutions. Consequently, CN1 uses a much larger context than 5^3 to compute the value of each output voxel.

A convolutional network makes efficient use and reuse of computation Alternatively, CN1 can be seen as a special case of a neural network that takes 25^3 patches as input, in which the synaptic weights are constrained to have a convolutional structure. This constraint makes CN1 highly efficient in its use of computational resources. Consider two 25^3 image patches that are displaced by a single voxel. If a general neural network were applied to both patches, the computations would be completely separate. But for CN1, most computations are shared by neighboring image patches, suggesting that convolutional networks are highly efficient algorithms for image processing.

References

[1] D. Ackley, G. Hinton, and T. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 1985. 5

[2] J. Besag. Efficiency of pseudolikelihood estimation for simple Gaussian fields. *Biometrika*, 1977. 4

[3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 2004. 5, 7

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 2001. 5, 7

[5] K. L. Briggman and W. Denk. Towards neural circuit reconstruction with volume electron microscopy techniques. *Curr Opin Neurobiol*, 2006. 1, 2, 3

[6] W. Denk and H. Horstmann. Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure. *PLoS Biology*, 2004. 2

[7] M. Egmont-Petersen, D. de Ridder, and H. Handels. Image processing with neural networks- A review. *Pattern Recognition*, 35, 2002. 2, 8

[8] X. Feng, C. Williams, and S. Felderhof. Combining belief networks and neural networks for scene segmentation. *PAMI*, 24(4):467–483, 2002. 5

[9] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *PAMI*, 1984. 1

[10] D. Greig, B. Porteous, and A. Seheult. Exact Maximum A Posteriori Estimation for Binary Images. *Journal of the Royal Statistical Society, Series B.*, 51(2):271–279, 1989. 7

[11] X. He, R. Zemel, and M. Perpinan. Multiscale conditional random fields for image labeling. *CVPR*, 2004. 1, 4, 5

[12] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 2006. 7, 8

[13] S. Kumar and M. Hebert. Discriminative fields for modeling spatial dependencies in natural images. *NIPS*, 16, 2004. 1, 5

[14] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *ICML*, 2001. 5

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11), 1998. 1, 2

[16] Y. LeCun, F. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *CVPR*, 2004. 1, 2

[17] S. Li. *Markov random field modeling in image analysis*. Springer-Verlag, New York, 2001. 4, 5

[18] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *ICCV*, 02:416, 2001. 8

[19] J. F. Murray and K. Kreutz-Delgado. Visual recognition and inference using dynamic overcomplete sparse learning. *Neural Computation*, 19:2301–2352, 2007. 8

[20] F. Ning, D. Delhomme, Y. LeCun, F. Piano, L. Bottou, and P. Barbano. Toward Automatic Phenotyping of Developing Embryos From Videos. *IEEE Trans. Image Proc.*, 2005. 2

[21] S. Parise and M. Welling. Learning in markov random fields: An empirical study. *Joint Statistical Meeting*, 2005. 4, 7

[22] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *PAMI*, 12(7):629–639, 1990. 3

[23] A. Quattoni, M. Collins, and T. Darrell. Conditional random fields for object recognition. *NIPS*, 17:2004, 2004. 5

[24] S. Roth and M. Black. Fields of Experts: a framework for learning image priors. *CVPR*, 2, 2005. 1

[25] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, 1995. 7