Visual Boundary Prediction: A Deep Neural Prediction Network and Quality Dissection

Jyri J. Kivinen
School of Informatics,
University of Edinburgh, UK

Christopher K. I. Williams
School of Informatics,
University of Edinburgh, UK

Nicolas Heess
DeepMind Technologies[1]
London, UK

Abstract

This paper investigates visual boundary detection, i.e. prediction of the presence of a boundary at a given image location. We develop a novel neurally-inspired deep architecture for the task. Notable aspects of our work are (i) the use of "covariance features" [Ranzato and Hinton, 2010] which depend on the *squared* response of a filter to the input image, and (ii) the integration of image information from multiple scales and semantic levels via multiple streams of interlinked, layered, and non-linear "deep" processing. Our results on the Berkeley Segmentation Data Set 500 (BSDS500) show comparable or better performance to the top-performing methods [Arbelaez et al., 2011, Ren and Bo, 2012, Lim et al., 2013, Dollár and Zitnick, 2013] with effective inference times. We also propose novel quantitative assessment techniques for improved method understanding and comparison. We carefully dissect the performance of our architecture, feature-types used and training methods, providing clear signals for model understanding and development.

1 Introduction

We consider predicting visual boundaries in natural images. Martin et al. [2004] give the definition "A boundary is a contour in the image plane that represents a change in pixel ownership from one object or surface to another." Detecting such changes is a challenging problem and significantly different from simple edge detection: Edge detection, is a low-level technique to detect an abrupt change in some image feature such as brightness or colour. In contrast, boundary detection is involved with detecting abrupt changes in more global properties, such as texture, and therefore needs to integrate information across the image. So, for example, a heavily textured region might give rise to many edges, but there should be no boundary defined within the region. Although difficult, accurate detection of boundaries is important as it subserves many vision tasks including segmentation, recognition and scene understanding.

Accordingly, there has been considerable interest in this problem in the computer vision literature. Recent work on boundary detection makes heavy use of the ground truth provided by the Berkeley Segmentation Data Set (BSDS) [Arbelaez et al., 2011], where each of the 500 images was processed by multiple human annotators. The (deliberately vague) instructions to the annotators were [Martin et al., 2004]: "Divide the image into some number of segments, where the segments represent 'things', or 'parts of things' in the scene. The number of segments is up to you, as it depends on the image. Something between 2 and 30 is likely to be appropriate. It is important that all of the segments have approximately equal importance."

In this paper we ask the question to what extent a general purpose learning architecture such as neural networks can be used to solve this challenging problem and we aim to understand which network properties are critical for good performance. Notable aspects of our best performing architecture are (i) the use of complex-cell like "covariance features" [Ranzato and Hinton, 2010] which depend on the *squared* response of a filter to the input image, and (ii) the integration of image information from multiple scales and semantic levels via multiple streams of interlinked, layered, and non-linear "deep" processing. We further propose two extensions to the commonly used BSDS benchmark

[1]The work reported was carried out while NH was at the Gatsby Unit, UCL, UK.

Appearing in Proceedings of the 17^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2014, Reykjavik, Iceland. JMLR: W&CP volume 33.

protocol which provide further insight into strenghts and weaknesses of different algorithms.

Our results on the Berkeley Segmentation Data Set 500 (BSDS500) show comparable or better performance to the top-performing methods gPb [Arbelaez et al., 2011], SCG [Ren and Bo, 2012], Sketch Tokens [Lim et al., 2013], and those in Dollár and Zitnick [2013]. Additionally, our approach scales effectively with fast prediction times, and avoids several computationally complex hand-crafted designs which are commonly used.

The structure of the paper is as follows: We describe our model and discuss related work in section 2. The benchmark evaluation protocol is discussed in Section 3. Section 4 describes the experiments, including descriptions of the data set, training procedure, and performance evaluation. Section 5 provides a summary and discussion of our main results, and outlines possible extensions.

2 Methods for Visual Boundary Prediction

Our network architecture can be conceptually divided into two parts, the first performs feature extraction, while the second uses the features for boundary prediction. For the first part we rely on unsupervised feature learning techniques and in section 2.1 we describe a variant of the mean-and-covariance restricted Boltzmann machine (mcRBM) architecture of Ranzato and Hinton [2010], and its deep belief net extension, which we use for this purpose. The learned features are combined with one or multiple read-out layers and the full network is trained in a supervised manner. This is described in section 2.2.

2.1 Unsupervised feature learning

The mcRBM-model [Ranzato and Hinton, 2010] is a generative model for images. The variant we consider assigns an energy to the joint configuration of visible units $\mathbf{v}$ and hidden units $\mathbf{h}$ as follows:

$$E = -\sum_j h_j^c \left(d_j - \sum_f \frac{\pi_{fj}}{2} \left[\mathbf{K}_{\cdot f}{}^\top \mathbf{A}\mathbf{v} \right]^2 \right) + \sum_i \frac{(v_i - a)^2}{2\sigma^2} - \sum_\ell h_\ell^m \left(b_\ell + \frac{1}{\sigma^2} \mathbf{M}_{\cdot \ell}{}^\top \mathbf{v} \right), \quad (1)$$

where $\mathbf{K}_{\cdot f}$ denotes a factor-to-image-units filter for a factor with index f, and π is a factor pooling matrix($\pi_{jf} \geq 0 \ \forall \ j, f$; $\sum_f \pi_{jf} = 1 \ \forall \ j$). $\mathbf{M}_{\cdot \ell}$ denotes the mean-hidden-unit-to-visible-unit filter for unit type ℓ. Each of the covariance units h_j^c and mean hidden units h_ℓ^m are associated with biases d_j and b_ℓ, respectively. a is the visible unit bias, and σ is a positive scalar. We simplify the above formulation by setting π to the identity, σ to unity, and the visible unit bias to zero. We also introduce a pre-learned whitening basis $\mathbf{A}$[2], and pre-process input images to have zero mean and unit variance.

In our experiments we use diagonally-tiled parameter sharing, with 8×8 receptive fields, stride of 2 units, and 64/128 (in grey/colour-domain) features of both kinds per site. Figure I in the appendix provides a simplified illustration of such a diagonally-tiled convolutional mcRBM (TmcRBM) model instance.

For feature learning in our deep architectures we develop a deep belief network (called the mcDBN) from the mcRBM, extending it to have an additional layer of binary hidden units on top, similarly to Dahl et al. [2010]. We use the same diagonally-tiled convolutional feature sharing architecture for the additional layer with a stride of one second layer unit (corresponding to 8 units in the visible layer). A top-layer hidden unit layer takes input from a 3×3 region of TmcRBM (64 + 64/128 + 128) hidden unit stacks under each of the 4 shifts, containing 512/1024 input 'channels' in total. Thus each of the second-layer hidden units are directly influenced by a 30×30-region of visible units[3]. We used 512/1024 (in grey/colour-domain) feature planes, and thus the second-layer has 3 sets of 512/1024 hidden units, each with their own sets of weights and biases. Figure II in the appendix shows a schematic of a diagonally-tiled convolutional mcDBN (TmcDBN) model instance.

2.2 Supervised boundary prediction

We consider feedforward sigmoidal neural networks for boundary prediction. The feature planes in first hidden layers of our networks are always formed by the activation probabilities of the mean and covariance hidden units of the TmcRBM. In later hidden layers, for a feature plane k given input $\mathbf{x}^k$, we compute the activation of unit z_i^k as

$$z_i^k = \text{sig}(g^k + \sum_j W_{ij}^k x_j^k), \quad (2)$$

where W^k is the weight matrix, g^k a scalar bias, and $\text{sig}(z)$ the logistic sigmoid $\text{sig}(z) = 1/(1 + e^{-z})$.

The output layer has the same dimensionality as the input image and each pixel i has a corresponding contour unit $u_i = z_i^{\text{out}}$. Its activation is interpreted as

[2]The basis was learned from all 8×8 patches in the training data. Retained dimensions were scaled according to the inverse of the square root of the associated eigenvalue of the scatter matrix, similar to ZCA.

[3]3 replicas of 8×8 filters at all of the diagonal-2-shifts (6 unit offsets between the first and the last).

the predicted probability that pixel i is part of a contour. Depending on the architecture, the output layer receives input from one or more hidden layers in the network.

We consider three architectures of boundary prediction networks, as shown in Fig. 1. The "shallow" network has only a single layer of hidden units, those corresponding to the features of the mcRBM model. The "deep stream" architecture makes contour predictions based on the mcDBN-type hidden units, while the "two-stream" architecture (see Fig. 1 and III) uses the connection patterns of both the shallow and deep streams via skip-layer connections (see e.g. Ripley [1996, page 144] or Sermanet and LeCun [2011]).

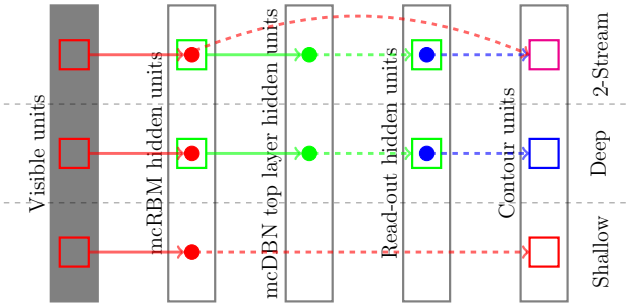

Figure 1: **Streams** of the Considered Networks. Dots in a layer denote hidden units, small blocks receptive fields (no size-consistency). The solid and dashed arrows denote feature-extraction and read-out parameters, respectively.

Importantly we can think of each stream having two parts: image feature extraction, and hypothesis propagation/read out (solid vs. dashed lines in Fig. 1)[4]. In our networks, we use a mirrored connectivity structure for these two parts. For example, in the shallow networks each hidden unit receives input from an 8×8 region of visible units, and sends information to an 8×8 region of contour units, with the same relative positioning. Adding an additional encoding layer thus adds another read-out hidden layer. One reason for doing so is the reduction in the hidden unit grid locations in the deeper networks. For example for a 142×142 input image, there are only 5×5 hidden unit stack positions at the second hidden layer, for each of the 3 shifts. In these deeper streams, the number of feature planes is also applied in an anti-symmetrical fashion, and so the first and the last hidden layers in the stream have equal numbers of planes. More complicated structures could be considered, see i.e. Fig. IV.

[4]Their boundary is expected to be blurred when network-wide parameter changes are allowed in training.

The multiple streams are motivated by the need to capture image information at multiple scales and semantic levels. The shallow networks analyze the input image very locally, each site being affected by a 14×14 pixel area around it. Such a network is expected to only detect very local image discontinuities such as edges. Deeper networks taking input from larger areas are likely to be needed for distinguishing texture discontinuities; our results are in line with this notion.

Due to the sparse spatial application of the filters throughout the networks, the deeper scales are expected to have only much coarser-scale information about the data, and contour sites with very local discontinuities cannot be detected efficiently. Combining the streams is expected to be beneficial; our results demonstrate enhanced recall-rates for the two-stream over deep-only networks.

2.3 Related work

The Canny edge detector [Canny, 1986] computes the edge response magnitude $\sqrt{G_x^2 + G_y^2}$ at each pixel, where G_x (resp. G_y) denotes the response of a Gaussian first derivative in the x (resp. y) direction, followed by stages of non-maximum suppression and hysteresis thresholding. Note that like our method it involves a squared filtering operation followed by nonlinear processing, but in contrast there are a small, hand-crafted set of filters and post-processing steps.

An important reference method for boundary detection is gPb [Arbelaez et al., 2011]. gPb is based on a Pb (probability of boundary) predictor that considers differences between histograms of brightness, colour and texture in two half-circles around an oriented edge. These results are then combined across multiple scales, and a "globalization" step based on spectral clustering is added. These operations are mainly hand-crafted, although there is some optimization of cue combination coefficients.

There have been a number of papers that have considered more wholesale learning approaches, taking as input an image patch and predicting the presence/absence of a boundary at the centre pixel of the patch. For example Dollar et al. [2006] consider a large number of generic features such as gradients and differences between histograms at multiple locations, orientations and scales, and a probabilistic boosting tree is used as a classifier. Both Mairal et al. [2008] and Ren and Bo [2012] have used representations based on sparse coding, either directly (the former) or via pooling over oriented half-discs (the latter) to train linear classifiers. Recent works by Lim et al. [2013] and Dollár and Zitnick [2013] both train decision forests that learn to map a large number of low level im-

age features onto edge predictions for a given image patch. While the former employs a discrete set of edge-patches ("sketch tokens") as labels for the leaves of the trees which are fixed prior to decision tree training, the structured decision tree framework of the latter learns the leaf labels as part of the tree training.

In contrast to these approaches our method builds on the idea of filters with a squaring non-linearity to learn a set of image features that form a distributed representation of local and global image properties which are then mapped onto a contour prediction image via several layers of learned adaptive "deep" nonlinear processing.

Another line of work has addressed linking together edge fragments in order to create extended smooth contours. See e.g. Parent and Zucker [1989] for early work, and Zhu et al. [2007] for a more recent approach. We also note that boundary detection is closely related to region segmentation, a problem which has recently been adressed using convolutional neural networks (e.g. Farabet et al. [2012]), although contour detectors do not necessarily produce closed contours which partition the image into regions.

3 Methods for Visual Boundary Prediction Quality Assessment

In section 4 we consider several experiments to compare prediction performance across algorithms. Here, we first describe the standard evaluation protocol and then discuss our extensions.

The BSDS assessment protocol involves computing a precision-recall (P-R) curve, as explained in Martin et al. [2004]. A P-R curve can be summarized by computing the maximal F-measure score (the harmonic average of precision and recall) out of the points corresponding to thresholding with a particular value. This threshold can either be optimized across the data set (ODS), or on a per image basis (OIS). The F-score (ODS) is considered the main metric of the benchmark (see for example Martin et al. [2013a] and Martin et al. [2013b]). The P-R curve can also be summarized by the average precision (AP).

As Hou et al. [2013] demonstrate, the BSDS benchmark protocol is not without problems. While some boundaries are "strong" in the sense that all annotators tend to agree ("consensus boundaries") there is also a significant fraction of "weak" boundaries, which were marked only by a few, or even just a single annotator ("orphan boundaries"). As pointed out by Hou et al. the unreliable orphan boundaries make up a significant fraction of the annotations (with 30.15 % of all annotated boundaries they are almost as frequent as strong consensus boundaries 30.58 %) and the current benchmarking protocol tends to reward algorithms for focusing on these orphans (a phenomenon they refer to as a "precision bubble"). They further find that existing algorithms achieve relatively poor performance on the problem of predicting the unambiguous, strong boundaries only. While their results do not necessarily imply that one boundary type is more valuable than the other, they certainly suggest that a single P-R curve for the full set of annotations provides only limited information about an algorithm's strength and weaknesses for the purposes of different tasks. Motivated by these findings and to provide further insight into the behavior of different algorithms we choose to compute P-R curves not just for the full set of annotations but also compared algorithms with respect to their performance in predicting strong boundaries only (where, for our analysis we use the definition of "consensus" boundaries introduced by Hou et al. [2013]). The results of this comparison can be found in Sec 4.2.1.

This additional analysis (partially) addresses the problem of the ambiguity inherent in the boundary detection task. A second potential problem with the standard evaluation protocol is that the pixel-wise independent computation of hits and misses does not necessarily capture the *perceptual* quality of a boundary prediction (it ignores, for instance, spatial coherence of a prediction). Similar criticisms have previously been raised with regards to standard evaluation metrics for image restoration tasks where pixel-wise mean-squared-error based metrics such as the peak signal-to-noise ratio (PSNR) are widespread. One response to these criticisms has been the mean structural similarity index metric (MSSIM) [Wang et al., 2004] (see Appendix F for more details) which considers several kinds of image information, and, importantly, takes non-local information into account. It is widely regarded as a perceptually more valid metric than e.g. the PSNR. While it may not be immediately obvious that MSSIM is adequate also for comparing boundary prediction images we have found that MSSIM scores (comparing prediction images to human averages) generally agree very well with the (admittedly subjective) perceptual quality of such predictions. We have therefore included an evaluation in terms of this metric in our quantitative comparsion of boundary prediction algorithms, the results of which are discussed in Sec. 4.2.2.

4 Visual Boundary Prediction Experiments

We consider the BSDS500 dataset in our experiments. The dataset consists of 500 natural images and associated boundary annotations by several humans. We

follow strictly the protocol in the benchmark and in our other assessments; this includes not using the test set for model development and selection.

4.1 Training of the models

The mcRBM/mcDBN models were trained using stochastic gradient ascent for approximate maximum likelihood learning using FPCD [Tieleman and Hinton, 2009], with an implementation based on the TmPoT training code [Ranzato et al., 2010]. The mcDBNs were trained in the usual greedy manner, layer-by-layer. (See Appendix B.1 for details.)

For discriminative training of the boundary prediction networks we perform stochastic gradient ascent in the log-likelihood $\mathcal{L}$ of the training data. We maximize the conditional likelihood of a ground-truth contour map $\mathbf{y}$ given an image, $\mathcal{L} = \sum_{n,i} y_i^{(n)} \log u_i^{(n)} + (1 - y_i^{(n)}) \log(1 - u_i^{(n)})$, where $y_i^{(n)}$ is the label of the i^{th} contour unit in n^{th} training image, and $u_i^{(n)}$ the corresponding network prediction (cf. eq. (2)).

The read-out weights were initialized to small random values, and the biases to zeros except for the contour bias g which was set to match the overall probability of a contour in the training data. We use stochastic gradient ascent with mini-batches and momentum, and apply a small amount of L_2 regularization of the weights. Learning rates are initially kept constant and then decayed according to a $\frac{1}{t}$ schedule during the final phase of learning. (See section B.2 for details.)

To further improve prediction performance we employ five 'enhancements' in our full method (called "Enhanced Two-Stream" below; see section B.3 in the Appendix for full details): (1) We standardize each input image to have zero mean and standard deviation one, making the network more robust to shifts and changes in scale of global intensity. (2) During training we average the binary ground truth annotations from different annotators for each image to give a single probability map $\mathbf{y}$, reducing the noise in the gradient. (3) We also encourage sparsity of the hidden unit activations via a cross entropy penalty for deviations from a set target activation level. (4) During prediction, in order to improve transformation equivariance, we perform rotation averaging, applying the network to 16 rotated versions of each image and averaging the prediction results. (5) Finally, as in several previous works [Dollár and Zitnick, 2013, Mairal et al., 2008], we apply non-maximum suppression by the Canny method to the network predictions.

4.2 Results and method comparison

We now discuss the results of our method and the competing methods in the literature. We have considered all main state-of-the-art methods for which results or source code for their computation are available[5]. Example results for one image are given in Fig. 2; other results are shown in Appendix D.

4.2.1 Dissecting BSDS Boundary Prediction Performance

We first focus on the P-R analysis. These results are summarized in Table 1 and Figure 3.

Analyzing performance for gray-scale images (Table 1, bottom) we note that our network performs generally better or at least comparably to those approaches for which gray-level results or source code for their computation was available. This is true especially with respect to the main metric, F-score (ODS).

In the colour-domain (Table 1, top; see also Fig. 3), for the standard BSDS500 benchmark (boundaries of all strength; "any"), we find that our approach achieves similar performance to the strongest methods from the literature in terms of F-score (Dollár and Zitnick [2013] and SCG) although their AP is slightly higher than ours. Sketch Tokens also achieves a higher AP but is inferior in terms of F-scores.

Considering separately the performace for strong boundaries we find in agreement with Hou et al. [2013] that all methods perform significantly worse at predicting strong boundaries only (compared to predicting the full set of annotations); this is explained by the fact that many strong (but not consensus) edges now become false positives. We further notice, however, that relative to other methods our network emphasizes strong boundaries over weak ones: It performs especially well on the task of predicting consensus boundaries only, both in terms of AP and ODS.

4.2.2 Perceptual Quality Performance Analysis

To assess perceptual quality of boundary predictions we compared prediction images to human annotation averages in terms of the MSSIM metric. We first obtained the MSSIM-score for each test image and algorithm. For each image we then computed the pairwise score-difference between our method and those from the literature. Histograms of these difference scores are shown in Figure 4 (see Table I in the appendix for further details). Overall, the MSSIM scores tend to be higher for our method than for the other algorithms on a per-image basis. The most notable difference is between our method and Sketch Tokens. This is interesting as it paints a rather different picture from the P-R

[5]We thank Dollár and Zitnick [2013] for providing their BSDS500 test results via personal communication.

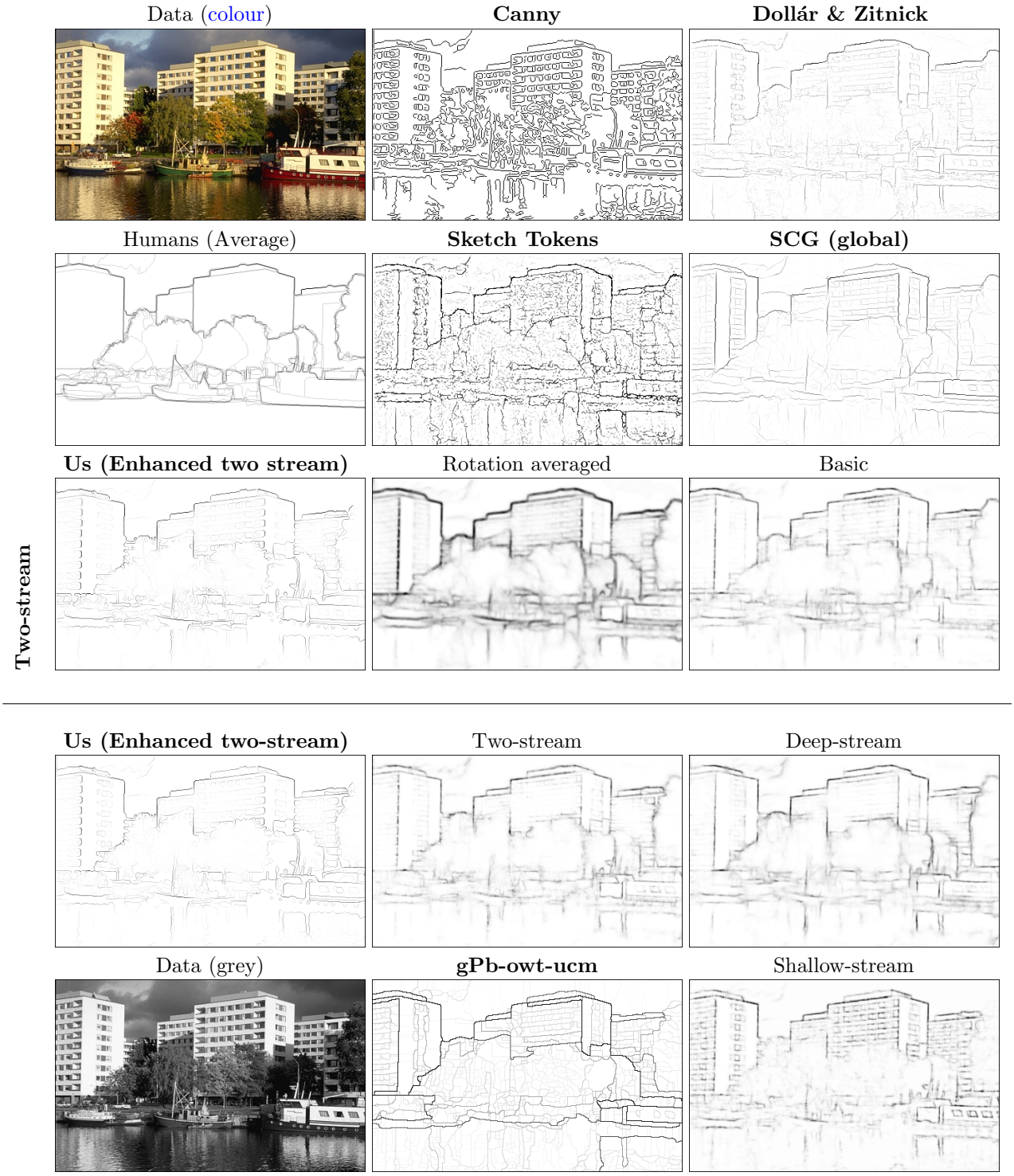

Figure 2: Contour Prediction Examples on the BSDS500. Predictions are individually rescaled to fill full intensity range for visualization purposes. Best viewed on screen.

Boundary Strength	F -score ODS	OIS	AP
Us (Enhanced Two-Stream)			
Any	0.738	0.759	0.758
Consensus	**0.613**	0.642	**0.611**
Dollár and Zitnick [2013]			
Any	**0.741**	**0.760**	**0.780**
Consensus	0.588	0.621	0.586
Sketch Tokens [Lim et al., 2013]			
Any	0.728	0.746	0.780
Consensus	0.569	0.592	0.561
SCG [Ren and Bo, 2012, (Global)]			
Any	0.739	0.758	0.773
Concensus	0.604	0.636	0.561
gPb-owt-ucm [Arbelaez et al., 2011]			
Any	0.727	0.759	0.727
Consensus	0.591	**0.648**	0.491
Us (Enhanced Two-Stream, Grey)			
Any	**0.722**	**0.740**	0.736
Concensus	**0.582**	**0.613**	**0.581**
gPb-ucm-owt [Arbelaez et al., 2011, Grey]			
Any	0.69	0.71	0.67
Consensus	0.543	0.599	0.381
SCG [Ren and Bo, 2012, (Global, Grey)]			
Any	0.71	0.73	0.74

Table 1: **BSDS500 (Colour, Grey)** Prediction Dissection. Highest score for each statistic and boundary category is highlighted.

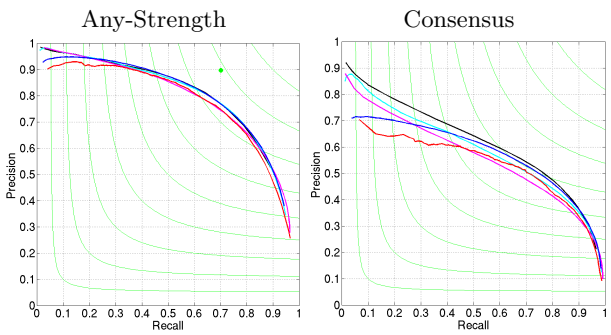

Figure 3: **BSDS500 (Colour) Precision-Recall Curves**. "Any strength" is the full BSDS500 annotation set (standard BSDS500 benchmark evaluation). Colors indicate algorithms: Us, Dollár and Zitnick [2013], Sketch Tokens, SCG (global), gPb-owt-ucm. Best viewed on screen.

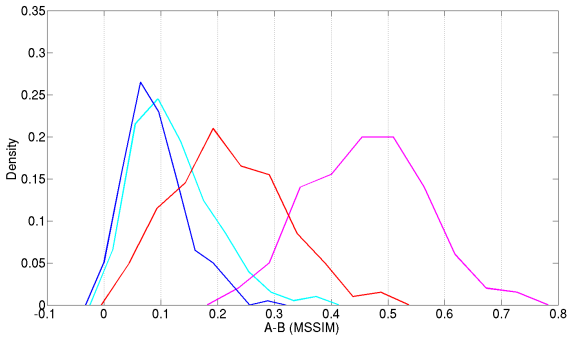

Figure 4: BSDS500 **MSSIM**-Score Difference Densities; Us Minus Competition. The density-curves are method-specific and colour-coded: Dollár and Zitnick [2013], Sketch Tokens, SCG (global), gPb-owt-ucm. Best viewed on screen.

analysis where Sketch Tokens performs well according to the summary results, especially in terms of AP. The differences relative to SCG and Dollár and Zitnick [2013] are much smaller and roughly in line with the P-R comparison. In general, we find the MSSIM scores to be relatively well correlated with the perceptual quality of the contour prediction as is illustrated in section D in the appendix, where individual prediction images and associated MSSIM scores are shown (Figs. XIV- XIX). Notice, for instance, in Fig. XIV how our approach is particularly effective at suppressing non-boundary edges on the tree and in the background.

4.2.3 Computational efficiency

Our approach scales well with image size in contrast to gPb and SCG, and provides significantly faster prediction times empirically (prediction time is 240s and 280s per image for global gPb and SCG respectively, and 60 and 100s for their local versions, cf. Lim et al. 2013). As an example, our unoptimized GPU implementation of the two-stream model takes 0.1 to 0.2 s per test image (grey-scale; 3 times for colour-domain with twice the number of features). Our enhanced two-stream inference is currently serial over the orientations but easily parallelized. We note the speeds of Sketch tokens [Lim et al., 2013](1 s) and that of in Dollár and Zitnick [2013] (1/6 s) are comparable to us in speed without the rotation-averaging. See Appendix C.2 for more details.

4.3 Dissecting the deep neural prediction network

In order to understand which features of our architecture were important for achieving good prediction results we performed a careful analysis. Here we will only summarize the main findings, the full results can be found in Appendix E. We focused on (i) the comparison between shallow and different types of deep networks; (ii) the relative importance of mean and covariance units; (iii) the effect of unsupervised pretraining and supervised fine tuning.

	F-score		
Model	**ODS**	OIS	AP
Two-stream:			
Full	0.695	0.709	0.673
Covariance-only	0.692	0.708	0.683
Mean-only	0.679	0.696	0.657
Deep-Stream:			
Full	0.702	0.715	0.654
Covariance-only	0.696	0.710	0.652
Mean-only	0.679	0.694	0.627
Shallow-Stream:			
Full	0.648	0.668	0.637
Covariance-only	0.642	0.660	0.624
Mean-only	0.629	0.652	0.613

Table 2: Network Prediction Performance Dissection on **BSDS500 (grey-scale)**. Mean-only and Covariance-only denote network parts including only and branching from mean and covariance units, respectively. See Table II for an extended dissection.

Shallow vs. deep networks We found that shallow networks were generally outperformed by multilayer networks by a significant margin (compare shallow-stream vs. deep-/two-stream in Table 2). This was expected as shallow networks have very local receptive fields and are thus not able to integrate image information more globally. Deep networks with this capability were much better at suppressing local, non-boundary edge structure (cf. Fig. 2 bottom). Introducing skip-layer connections in deep networks tended to further improve AP (but not having much effect on the F-score; compare two-stream vs. deep in Table 2) consistent with the idea that direct access to low-level features helped improving precision at high recall.

Covariance units Networks with covariance units generally had higher prediction performance than networks with mean units only. This difference was especially pronounced when the weights obtained with unsupervised pre-training were not fine tuned. Using mean units in combination with covariance units provided a small additional advantage relative to the use of covariance units only.

Unsupervised pre-training and fine tuning Unsupervised pre-training and fine tuning both improved final network performance. In particular, we found the positive effect of unsupervised pre-training to increase when moving from shallow to deep networks.

5 Discussion

We have developed a deep neural network architecture for visual boundary prediction built on top of a diagonally-tiled convolutional mcRBM for feature learning. The architecture is very different from previous approaches to this problem. It allows end-to-end optimization, and fast and scalable inference for prediction. We achieve accuracy comparable or better to those of the best-performing methods on the BSDS500 dataset and do not require an expensive "globalization" step, leading to prediction times that are highly competitive with most existing methods.

Our extended evaluation protocol which breaks down the analysis for different boundary strengths and considers visual plausibility highlights some interesting differences between methods that achieve very similar summary performance in terms of F-score and AP. We found our network to be especially good at predicting strong boundaries (Hou et al. [2013]), and to produce visually more plausible boundary predictions than Sketch Tokens despite being outperformed by the latter in terms of mAP.

Our careful analysis of different network architectures emphasizes the importance of network depth for integrating image information across multiple scales which appears to be critical for good performance. Covariance features are also a crucial ingredient, in agreement with previous findings that squaring nonlinearties form an important component of vision pipelines. We finally observe significant benefits of generative pre-training, possibly indicating that the amount of annotated training data is limited relative to the difficulty of the task.

We are currently exploring improved initialization techniques. Above the read-out parameters were initialized to random values. An interesting alternative would be to consider a joint model of the image and contour data. This could take the form of a dual-wing harmonium [Xing et al., 2005], in the simplest case in the form of a RBM that has two sets of visible units: the RGB and boundary images. Such joint models would allow not just the learning of "prediction features" as an initialization step, but also for other kinds of interesting applications, including image-prediction from boundary data (de-sketching) and image completion. Furthermore, the introduction of stochastic hidden units would allow dependencies between the contour units, providing a form of 'globalization' as well as a principled way of dealing with different alternative boundary predictions for a given image.

Acknowledgements

NH acknowledges funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 270327, and from the Gatsby Charitable foundation.

References

P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. PAMI*, 33(5):898–916, 2011.

J. Canny. A computational approach to edge detection. *IEEE Trans. PAMI*, 8(6):679–698, 1986.

B. Catanzaro, B. Y. Su, N. Sundaram, Y. Lee, M. Murphy, and K. Keutzer. Efficient, high-quality image contour detection. In *Proceedings, International Conference on Computer Vision (ICCV)*, pages 2381–2388, 2009.

G. E. Dahl, M.'A. Ranzato, A.-R. Mohamed, and G. E. Hinton. Phone recognition with the mean-covariance restricted Boltzmann machine. In *Proceedings, Advances in Neural Information Processing Systems (NIPS)*, 2010.

P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *Proceedings, International Conference on Computer Vision (ICCV)*, 2013.

P. Dollar, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. In *Proceedings, Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2006.

C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Scene parsing with multiscale feature learning, purity trees, and optimal covers. In *Proceedings, International Conference on Machine Learning (ICML)*, 2012.

G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.

X. Hou, A. Yuille, and C. Koch. Boundary detection benchmarking: Beyond F-measures. In *Proceedings, Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

V. W. Lee, C. Kim, J. Chhugani, M. Deisher, D. Kim, A. D . Nguyen, N. Satish, M. Smelyanskiy, S. Chennupaty, P. Hammarlund, R. Singhal, and P. Dubey. Debunking the 100X GPU vs. CPU myth: an evaluation of throughput computing on CPU and GPU. *SIGARCH Comput. Archit. News*, 38(3):451–460, June 2010.

J. Lim, L. Zitnick, and P. Dollar. Sketch tokens: A learned mid-level representation for contour and object detection. In *Proceedings, Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

J. Mairal, M. Leordeanu, F. Bach, M. Hebert, and J. Ponce. Discriminative sparse image models for class-specific edge detection and image interpretation. In *Proceedings, European Conference on Computer Vision (ECCV)*, 2008.

D. Martin, C. Fowlkes, D. Tal, and J. Malik. Berkeley Segmentation Dataset and Benchmark: About the Benchmark, 2013a. Page[6] generated 8 Aug. 2013.

D. Martin, C. Fowlkes, D. Tal, and J. Malik. Berkeley Segmentation Dataset and Benchmark: Boundary Detection Benchmark: Algorithm Ranking, 2013b. Page[7] generated 20 Feb. 2013.

D. Martin, C. Fowlkes, D. Tal, and J. Malik. Berkeley Segmentation Data Set and Benchmark: BSDS500, Best Practice Guidelines, 2014. Page[8] accessed 23 Feb. 2014.

D. R. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. PAMI*, 26(5):530–549, 2004.

P. Parent and S. W. Zucker. Trace inference, curvature consistency, and curve detection. *IEEE Trans. PAMI*, 11(8):823–839, 1989.

M.'A. Ranzato and G. E. Hinton. Modeling pixel means and covariances using factorized third-order Boltzmann machines. In *Proceedings, Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

M.'A. Ranzato, V. Mnih, and G. E. Hinton. Generating more realistic images using gated MRF's. In *Proceedings, Advances in Neural Information Processing Systems (NIPS)*, 2010.

X. Ren and L. Bo. Discriminatively trained sparse code gradients for contour detection. In *Proceedings, Advances in Neural Information Processing Systems (NIPS)*, pages 593–601, 2012.

B. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, UK, 1996.

P. Sermanet and Y. LeCun. Traffic sign recognition with multi-scale convolutional networks. In *Proceedings, International Joint Conference on Neural Networks (IJCNN)*, 2011.

T. Tieleman and G. E. Hinton. Using fast weights to improve Persistent Contrastive Divergence. In *Proceedings, International Conference on Machine Learning (ICML)*, pages 1033–1040. ACM New York, NY, USA, 2009.

Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Perceptual image quality assessment: From error visibility to structural similarity. *IEEE Trans. IP*, 13(4):600–612, April 2004.

[6] `http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/`

[7] `http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/bench/html/algorithms.html`

[8] `http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html`

E. P. Xing, R. Yan, and A. G. Hauptmann. Mining associated text and images with dual-wing harmoniums. In *Proceedings, Uncertainty in Artificial Intelligence (UAI)*, 2005.

Q. Zhu, G. Song, and J. Shi. Untangling cycles for contour grouping. In *Proceedings, International Conference on Computer Vision (ICCV)*, 2007.