



HAL
open science

Embedding Blake-Zisserman Regularization in Unfolded Proximal Neural Networks for Enhanced Edge Detection

Hoang Trieu Vy Le, Marion Foare, Audrey Repetti, Nelly Pustelnik

► **To cite this version:**

Hoang Trieu Vy Le, Marion Foare, Audrey Repetti, Nelly Pustelnik. Embedding Blake-Zisserman Regularization in Unfolded Proximal Neural Networks for Enhanced Edge Detection. 2024. hal-04771534

HAL Id: hal-04771534

<https://hal.science/hal-04771534v1>

Preprint submitted on 7 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Embedding Blake–Zisserman Regularization in Unfolded Proximal Neural Networks for Enhanced Edge Detection

Hoang Trieu Vy Le¹, Marion Foare^{2,3}, Audrey Repetti⁴, and Nelly Pustelnik⁵

¹ Univ. Paris-Saclay, CEA, List, F-91120, Palaiseau, France

² CPE Lyon, Villeurbanne, 69100, France

³ Univ. Claude Bernard Lyon 1, ENS de Lyon, CNRS, Inria, LIP, UMR 5668, 69342, Lyon cedex 07, France

⁴ School of Mathematical and Computer Sciences, and the School of Engineering and Physical Sciences,
Heriot-Watt University, Edinburgh, EH14 4AS

⁵ Laboratoire de Physique, ENSL, CNRS UMR 5672, F-69342, Lyon, France

*

November 7, 2024

Abstract

In this paper, we present a new edge detection model based on proximal unfolded neural networks. The architecture relies on unfolding proximal Blake–Zisserman iterations, leading to a composition of two blocks: a smoothing block and an edge detection block. We show through simulations that the proposed approach efficiently eliminates irrelevant details while retaining key edges and significantly improves performance with respect to state-of-the-art strategies. Additionally, our architecture is significantly lighter than recent learning models designed for edge detection in terms of number of learnable parameters and inference time.

Keywords. *Mumford–Shah functional, Blake–Zisserman functional, edge detection, proximal neural networks, optimization.*

1 Introduction

Image edges represent a specific image feature encoding valuable information playing a critical role in a variety of downstream tasks. For example, in experimental physics, edges are used to estimate the phases of contact areas, helping at the identification of hydrodynamic regimes [18].

Traditional edge detection methods rely on hand-crafted features using low-level visual cues such as gradients, which could precisely localize edges but often struggled with accuracy in noisy scenes, shadows or texture. This problem highlights the need for image regularization that removes insignificant and minor details from the scene, while balancing two opposing objectives: smoothing the homogeneous areas of the image and preserving the boundaries of these regions. Methods based on these principles are known as variational approaches, which have been a significant area of research, particularly in the context of joint image denoising and edge detection. This line of work was initiated by the contributions of Mumford and Shah [14], as well as similar ideas proposed by Geman and Geman [9] or Blake and Zisserman [4].

The main drawbacks of variational approaches include the large number of iterations to ensure its convergence. Hence it necessitates a meticulous tuning of the algorithms and hyperparameters to achieve optimal performance.

Nowadays, alternative solutions benefiting from the power of deep neural networks, and fully decorrelated from the Mumford–Shah analysis, have been proposed for edge preserving. Many architectures have been

*This work is funded by the Fondation Simone et Cino Del Duca - Institut de France. The authors thank the Centre Blaise Pascal of ENS Lyon for the computation facilities. The platform uses SIDUS, which was developed by Emmanuel Quemener.

developed in this context. Among methods using convolutional neural networks, we can refer to DeepEdge [3] and DeepContour [19] based on image patches and associated features to identify edges. These methods demonstrated performance improvement over the state-of-the-art Structured Edges method [6]. Recently, [20] proposed a method relying on a VGGnet, called HED, which combines edges obtained from different scales into a final fused edge map. To achieve effective results, methods such as BDCN [10] is designed with less parameters compared to HED, by computing the loss for each layer associated with a specific scale. These methods significantly improve accuracy by integrating features from multiple layers. Recently, inspired by the success of vision transformers [7], a transformer-based edge detector (EDTER) was proposed in [15], which splits images into two sequences of 16×16 and 8×8 patches, then passing through a transformer encoder for detecting global and local features, to finally predict object and local boundaries respectively. However, the resulting edge map tends to be too thick for downstream tasks and are highly dependent on post-processing. To address this, [21] introduced a diffusion model for edge detection (DiffEdge) producing sharp edges without the need for post-processing. Despite these improvements, these learning-based models have become increasingly complex, with larger architectures and slower inference speeds. They also require a resource-intensive training phase and substantial support frameworks to improve inference performance.

For many years, there has been a gap between variational approaches and neural network procedures, as the first one was guided by a combination of the data acquisition physics and prior knowledge about the object, while the second one was considered as a very efficient prior-free black-box procedure. However, in the field of inverse problems, model-based neural networks have been designed to help bridging that gap. They are often referred to as Plug-and-Play or unfolded strategies. This class of approaches offers a trade-off between theoretical guarantees and performances. While model-based neural networks appear now as the state-of-the-art in the field of image restoration, their use in the context of edge detection is nonexistent. Hence this is the core of this contribution.

Contributions and outline. Focusing on the Blake–Zisserman (BZ) variational approach for edge detection, we propose an unfolded scheme relying on iterative procedure mimicking BZ minimization strategy. The resulting unfolded method is then composed of two blocks: (i) a smoothing Proximal Neural Network (PNN), and (ii) an edge detection layer. Section 2 recalls the Mumford–Shah functional and its equivalence with BZ. Section 3 is dedicated to the proposed unfolded BZ-PNN strategy, and establishes a link between standard activation functions used for edge detection and BZ thresholding steps. Finally, Section 4 presents the experimental results including comparisons between standard variational procedure, deep neural network strategies, and the proposed BZ-PNN.

2 Blake–Zisserman

Edge detection aims to identify points of intensity discontinuity $\mathbf{e} \in \mathbb{R}^N$ in an image $\mathbf{z} \in \mathbb{R}^{CN}$ composed with C channels and N pixels. This image may result from noisy measurements, for instance in the presence of an additive white Gaussian noise. The Mumford–Shah functional [14], originally designed in the continuous setting, aims to recover both the edges and a smooth image in a context of a degraded image. Its discrete counterpart, proposed in [8], reads

$$\min_{\substack{\mathbf{x} \in \mathbb{R}^{CN} \\ \mathbf{e} \in \mathbb{R}^{CN}}} \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 + \lambda \sum_{c=1}^C \sum_{i,j} (1 - e_{c,i,j})^2 \|(\mathbf{D}\mathbf{x}_c)_{i,j}\|^2 + \beta g(\mathbf{e}), \quad (1)$$

where the first term is the data fidelity term, the second term imposes smoothness everywhere except on the channel-wise edge set $\mathbf{e}_c \in \mathbb{R}^N$, with $\mathbf{D}: \mathbb{R}^N \rightarrow \mathbb{R}^{2N}$ denoting the horizontal and vertical derivative operators, and the last term $g: \mathbb{R}^{CN} \rightarrow (-\infty, +\infty]$ represents the edge length promoting sparsity, with $\beta > 0$. Thus g can be chosen to be the ℓ_0 -pseudo norm, the ℓ_1 -norm or a quadratic ℓ_1 -norm, also known as the BerHu function.

It has been established in [16] that this bivariate minimization problem can be formulated as a univariate minimization problem over \mathbf{x} followed by a thresholding step. In particular, it is equivalent to the BZ two-step procedure [4] when g is the ℓ_0 -pseudo norm. This problem can then be formulated as finding estimates $(\hat{\mathbf{x}}, \hat{\mathbf{e}})$ such that

$$\hat{\mathbf{x}} \in \underset{\mathbf{x} \in \mathbb{R}^{CN}}{\text{Argmin}} \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 + \sum_{c=1}^C \sum_{i,j} \min(\beta, \lambda \|(\mathbf{D}\mathbf{x}_c)_{i,j}\|^2), \quad (2)$$

and $\widehat{\mathbf{e}}_c = (\widehat{\mathbf{e}}_{c,i,j})_{(c,i,j)}$ with

$$\widehat{\mathbf{e}}_{c,i,j} = \begin{cases} 1 & \text{if } \|(\mathbf{D}\mathbf{x}_c)_{i,j}\|^2 > \frac{\beta}{\lambda} \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where $\beta, \lambda > 0$ are hyperparameters controlling respectively the length of the edges and the smoothness of the estimate. The final estimate is then defined as $\widehat{\mathbf{e}} = \left(\min(1, \sum_c \widehat{\mathbf{e}}_{c,i,j}) \right)_{i,j}$.

In this two-step procedure, the first step in (2) aims to recover a smooth image with sharp edges. It is followed by a thresholding step (3) to capture the location of the predominant edges. The penalization term in (2) is a truncated ℓ_2 -norm but it is well-know that other type of penalizations allow to capture similar behaviour, including the log-sum penalization or the weighted- ℓ_1 penalization [17].

3 BZ-PNN: Unfolded proximal neural network for edge detection

Inspired by the two-step BZ procedure (2)-(3), we propose a new model-based neural network relying on a denoising unfolded PNN that we will use as a smoothing estimator, offering a piecewise smooth estimate, followed by an edge detection layer.

3.1 Unfolded smoothing PNN

In [13], the authors proposed a unified framework for building unfolded denoising PNNs with learned linear operators. The proposed architectures share a unifying building block that can be either related to the forward-backward algorithm [2] or Chambolle–Pock algorithm [5], tailored to solve the general variational (Gaussian denoising) problem

$$\widehat{\mathbf{x}}_{\text{MAP}} = \underset{\mathbf{x} \in \mathbb{R}^N}{\text{argmin}} \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 + \nu \|\mathbf{D}\mathbf{x}\|_1, \quad (4)$$

where $\nu > 0$ is a regularization parameter.

In this work, we focus on the unfolded instance relying on Chambolle-Pock algorithm for strongly convex functions (ScCP), since it appears to provide a better compromise between robustness and performance. The associated iterations can be written as

$$\begin{aligned} & \text{for } k = 1, \dots \\ & \begin{cases} \mathbf{u}_k = \text{prox}_{\tau_k(\nu\|\cdot\|_1)^*} \left(\mathbf{u}_{k-1} + \tau_k \mathbf{D} \left((1 + \alpha_k) \mathbf{x}_k - \alpha_k \mathbf{x}_{k-1} \right) \right), \\ \mathbf{x}_k = \frac{\mu_k}{1 + \mu_k} (\mathbf{z} - \mathbf{D}^\top \mathbf{u}_k) + \frac{1}{1 + \mu_k} \mathbf{x}_{k-1}, \end{cases} \end{aligned} \quad (5)$$

where $(\alpha_k, \tau_k, \mu_k)_{k \in \mathbb{N}}$ denote both the step-size and inertial parameters of the algorithm. The resulting unrolled architecture **DScCP-LNO** introduced in [13] can then be described as

$$\mathbf{x}_K = f_{\mathbf{z}, \nu, \Lambda}^{K, \text{DScCP}}(\mathbf{z}) \quad (6)$$

with

$$\begin{cases} \mathbf{x}_k = (1 + \alpha_k) \widetilde{\mathbf{x}}_k - \alpha_k \mathbf{x}_{k-1}, \\ (\widetilde{\mathbf{x}}_k, \mathbf{u}_k) = \mathbf{L}_{\mathbf{z}, \Lambda_k, \mathcal{P}}(\mathbf{x}_{k-1}, \mathbf{L}_{\Lambda_{\nu_k, k, \mathcal{D}}}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})) \end{cases}$$

which combines standard neural network layers (activation function, linear step, bias) both on the dual space (denoted \mathcal{D}) and on the primal space (denoted \mathcal{P})

$$\begin{cases} \mathbf{L}_{\mathbf{z}, \Lambda_k, \mathcal{P}}(\mathbf{x}, \mathbf{u}) = \eta_k, \mathcal{P}(\mathbf{W}_{k, \mathcal{P}} \mathbf{x} + \mathbf{V}_{k, \mathcal{P}} \mathbf{u} + \mathbf{b}_{k, \mathcal{P}}), \\ \mathbf{L}_{\Lambda_{\nu_k, k, \mathcal{D}}}(\mathbf{x}, \mathbf{u}) = \eta_{\nu_k, k, \mathcal{D}}(\mathbf{W}_{k, \mathcal{D}} \mathbf{x} + \mathbf{V}_{k, \mathcal{D}} \mathbf{u} + \mathbf{b}_{k, \mathcal{D}}), \end{cases}$$

with

$$\begin{cases} \mathbf{W}_{k, \mathcal{P}} = \frac{1}{1 + \mu_k}, \\ \mathbf{V}_{k, \mathcal{P}} = -\frac{\mu_k}{1 + \mu_k} \mathbf{D}_{k, \mathcal{P}}, \\ \mathbf{b}_{k, \mathcal{P}} = \frac{\mu_k}{1 + \mu_k} \mathbf{z}, \\ \eta_{k, \mathcal{P}} = \text{Id}, \end{cases} \quad \text{and} \quad \begin{cases} \mathbf{W}_{k, \mathcal{D}} = \tau_k \mathbf{D}_{k, \mathcal{D}}, \\ \mathbf{V}_{k, \mathcal{D}} = \text{Id}, \\ \mathbf{b}_{k, \mathcal{D}} = 0, \\ \eta_{\nu_k, k, \mathcal{D}} = \text{prox}_{\tau_k(\nu_k \|\cdot\|_1)^*}. \end{cases}$$

In the proposed unfolded scheme, the set of learnable parameters is $\Lambda = \{\nu, \Lambda_{k,\mathcal{P}}, \Lambda_{k,\mathcal{D}}\}_{1 \leq k \leq K} = \{\nu, \mathbf{W}_{k,\bullet}, \mathbf{V}_{k,\bullet}, \mathbf{b}_{k,\bullet}\}_{1 \leq k \leq K}$. Since the hyperparameter ν in variational problem (4) controls the smoothness of the estimate, we propose to turn $\nu = (\nu_1, \dots, \nu_K)$ into a learnable hyperparameter, while in the original **DScCP-LNO** architecture (6), $\nu_k \equiv \nu$ is a fixed (i.e., not learned) parameter. It allows to mimic a weighted- ℓ_1 strategy where the regularization parameter is changing from an iteration to the another one, and is thus closer to the original BZ minimization problem (2).

3.2 Edge detection layer

Following the BZ strategy (2)-(3), we propose to add to the **DScCP-LNO** smoother an edge detection layer composed of an additional activation function, denoted η_χ , and a weight matrix designed from BZ formalism, i.e.,

$$\mathbf{e} = \eta_\chi \left(\mathbf{D}_{K+1} \mathbf{x}^{[K]} \right) \quad (7)$$

where $\mathbf{x}^{[K]}$ is a smooth approximation of \mathbf{z} , i.e., the output of the smoother $f_{\mathbf{z}, \nu, \Lambda}^{K, \text{DScCP}}$, and $\mathbf{D}_{K+1} : \mathbb{R}^{CN} \rightarrow \mathbb{R}^N$ is an additional linear operator mapping an image to an edge map. The non-linear function η_χ should be chosen as the activation function $e \mapsto \text{HardTanh}_\chi(e^2)$, to fit the BZ thresholding (3) when $\chi = \frac{\beta}{\lambda}$.

Proposition 3.1. *The truncated quadratic function*

$$\eta_{\frac{\beta}{\lambda}}(e) = \min \left(\frac{\beta}{\lambda}, e^2 \right)$$

is equivalent to $\text{HardTanh}_{\frac{\beta}{\lambda}}(e^2)$.

Proof. This result is directly obtained by using definition of the HardTanh_χ activation function:

$$\text{HardTanh}_\chi(e) = \begin{cases} -\chi & \text{if } e < -\chi, \\ \chi & \text{if } e > \chi, \\ e & \text{otherwise.} \end{cases} \quad (8)$$

□

In order to favor the value of the edge map to be between $[0, 1]$, similarly as in the original Mumford-Shah formulation, we choose $\chi = \frac{\beta}{\lambda} = 1$ (see Fig. 1).

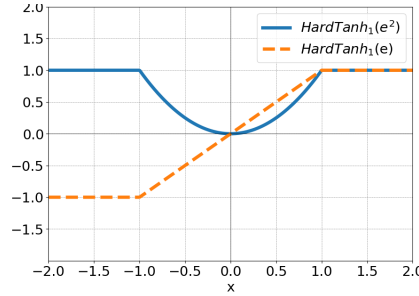


Figure 1: **Truncated quadratic function.**

3.3 BZ-PNN hybrid architecture

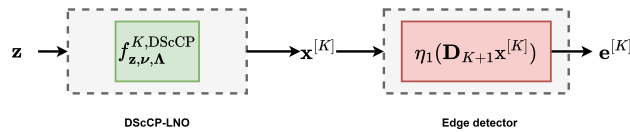


Figure 2: **Proposed BZ-PNN hybrid architecture.**

The proposed **BZ-PNN** hybrid architecture then combines the **DScCP-LNO** smoother (6) with the BZ-like edge detection layer (7). The resulting full architecture is illustrated in Fig. 2 and is summarized as, for any input image \mathbf{z} , $f_{\Theta}(\mathbf{z}) = (\mathbf{x}^{[K]}, \mathbf{e}^{[K]})$ with

$$\begin{cases} \mathbf{x}^{[K]} = f_{\mathbf{z}, \nu, \Lambda}^{K, \text{DScCP}}(\mathbf{z}) \\ \mathbf{e}^{[K]} = \eta_1 (\mathbf{D}_{K+1} \mathbf{x}^{[K]}) \end{cases} \quad (9)$$

and learnable parameters $\Theta = \{\Lambda, \mathbf{D}_{K+1}\}$.

4 Numerical experiments

Training dataset – Our experiments are performed on the BSD500 dataset [1], which provides multiple hand-drawn groundtruth contours for each image. We consider two sets of images: the *training set* $(\bar{\mathbf{e}}_s, \mathbf{z}_s)_{s \in \mathbb{I}}$ of size $|\mathbb{I}|$ and the *test set* $(\bar{\mathbf{e}}_s, \mathbf{z}_s)_{s \in \mathbb{J}}$ of size $|\mathbb{J}|$. For both sets, $\bar{\mathbf{e}}_s$ are the exact edges obtained by stacking all the provided groundtruth annotations, and \mathbf{z}_s consists of an input image (without noise).

Choice of the loss function – Following the literature dedicated to deep learning approaches for edge detection, the learning stage relies on the minimization of the weighted cross entropy loss [20]:

$$\hat{\Theta} \in \underset{\Theta}{\text{Argmin}} \sum_s \sum_{i,j} -\omega_{s,i,j} \left(\bar{\mathbf{e}}_{s,i,j} \log(f_{\Theta}(\mathbf{z}_s)_{i,j}) + (1 - f_{\Theta}(\mathbf{z}_s)_{i,j}) \log(1 - \bar{\mathbf{e}}_{s,i,j}) \right), \quad (10)$$

where $\omega_s \in \mathbb{R}^N$ denotes the class-balance weights between the edge pixel set $E_{s,+} = \{(i, j) : \bar{\mathbf{e}}_{s,i,j} = 1\}$ and non-edge pixel set $E_{s,-} = \{(i, j) : \bar{\mathbf{e}}_{s,i,j} = 0\}$, that is, $\omega_{s,i,j} = |E_-|/|E|$ when $\bar{\mathbf{e}}_{s,i,j} = 1$ and $\omega_{s,i,j} = |E_+|/|E|$ when $\bar{\mathbf{e}}_{s,i,j} = 0$.

The loss (10) is optimized in Pytorch with Adam algorithm [12].

Architecture comparison – We compare the proposed **BZ-PNN** architecture, for $K = 20$ layers and $J = 32$ filters in the smoothing block, with the state-of-the-art BZ variational model with the implementation provided by [11], and with two deep learning-based models, BDCN [10] and DiffusionEdge [21]. Table 1 summarizes the comparison in terms of learnable parameters (i.e. Θ) and inference time (in seconds) highlighting the much lighter architecture of our model. Regarding the BZ model, the inference time includes the Golden gridsearch hyperparameter tuning, as described in [13].

Table 1: **Architecture comparison.** Number of parameters $|\Theta|$ and inference time of state-of-the-art variational approach BZ and DL-based approaches, versus the proposed **BZ-PNN** architecture for $K = 20$ layers and $J = 32$ filters.

	$ \Theta $	Inference time (sec.)
BZ [4, 11]	–	800 ± 50 sec.
BDCN [10]	16, 302, 120	1.25 ± 0.06 sec.
DiffusionEdge [21]	137, 142, 150	10 ± 2.25 sec.
BZ-PNN	17, 347	0.278 ± 0.004 sec.

Results – As shown in Figures 3 and 4, the results indicate that the **BZ-PNN** strategy enhances edge detection performance compared to both the standard variational approach and the deep-learning ones. In comparison with BDCN, our model highlights local details more accurately without blurring these areas. When compared to a more advanced and complex model such as DiffusionEdge, in regions where there are miss-annotated edges, DiffusionEdge generates some faded edges while our method is able to sharply recover these edges. In terms of cross-entropy (CE), the proposed **BZ-PNN** almost systematically outperforms both variational and neural-network state-of-the-art approaches.

5 Conclusion

In this work, we propose a new model-based neural network architecture relying on the Blake–Zisserman variational model to detect edges. This hybrid model aims to harness the strengths of both traditional and modern deep learning-based approaches, offering a robust solution for edge detection with a lighter

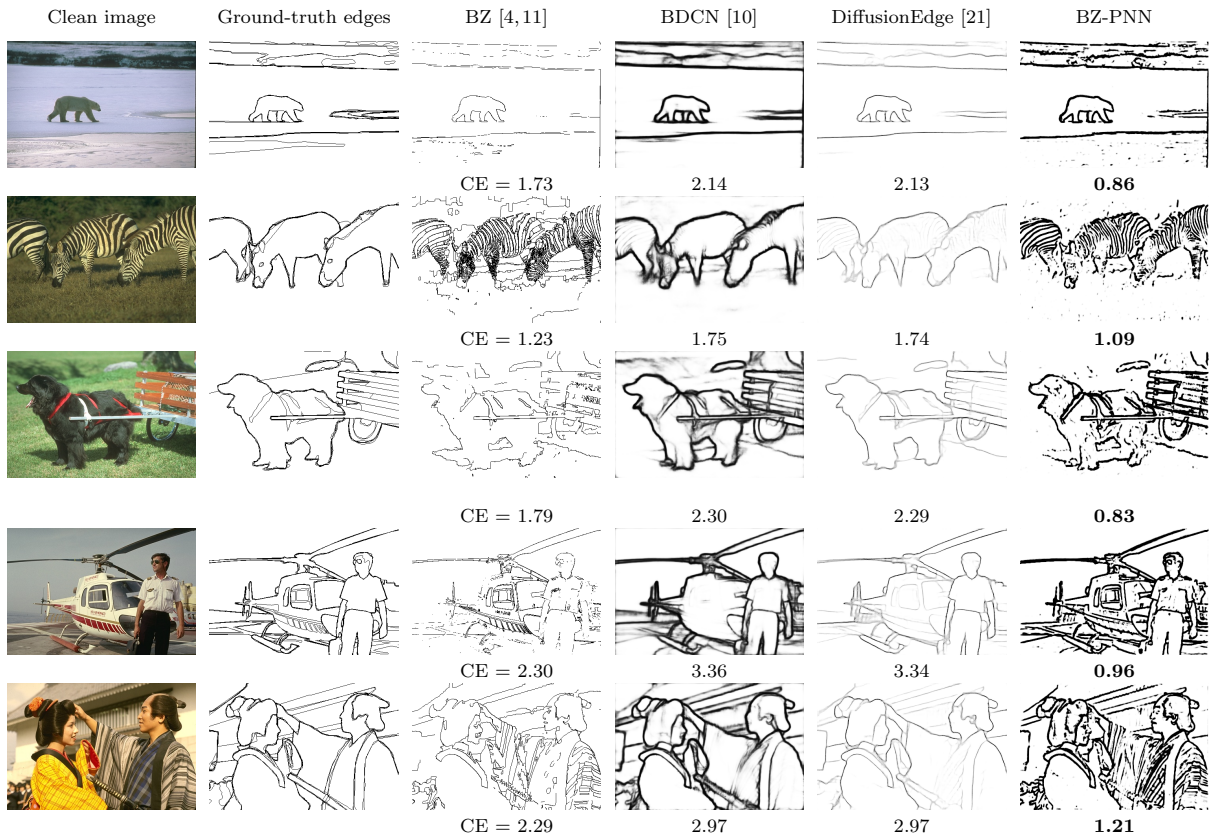


Figure 3: **Edge detection performance comparison.** Example of edge detection and Cross-Entropy (CE) values obtained with the Blake–Zisserman approach [4, 11], BDCN [10], DiffusionEdge [21] and the proposed **BZ-PNN**.

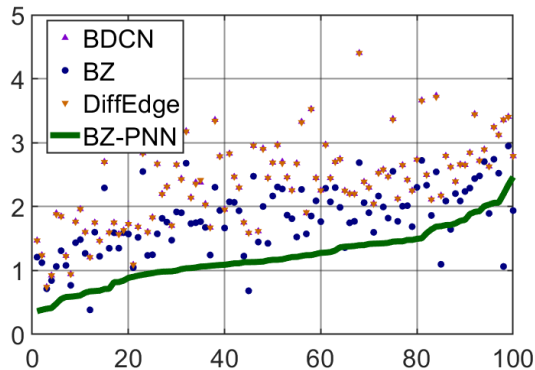


Figure 4: **Comparison of the CE values on 100 test images from the BSSDS500 database.** The x-axis (image number) for each score is sorted in ascending order according to the proposed **BZ-PNN** (solid green line).

architecture compared with other recent DL-based models. Through this approach, we strive to achieve a balance between computational efficiency and edge detection accuracy, pushing the boundaries of what existed in the class of variational approaches.

References

- [1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Match. Int.*, 33(5):898–916, 2010.

- [2] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.
- [3] G. Bertasius, J. Shi, and L. Torresani. DeepEdge: A multi-scale bifurcated deep network for top-down contour detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4380–4389, 2015.
- [4] A. Blake and A. Zisserman. *Visual reconstruction*. MIT press, 1987.
- [5] A. Chambolle and C. Dossal. On the convergence of the iterates of the Fast Iterative Shrinkage/Thresholding Algorithm. *J. Optim. Theory Appl.*, 166(3):968–982, 2015.
- [6] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *IEEE Trans. Pattern Anal. Match. Int.*, 37(8):1558–1570, 2014.
- [7] A. Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv: 2010.11929*, 2020.
- [8] M. Foare, N. Pustelnik, and L. Condat. Semi-linearized proximal alternating minimization for a discrete Mumford–Shah model. *IEEE Trans. Image Process.*, 29(1):2176–2189, 2019.
- [9] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Match. Int.*, pages 721–741, 1984.
- [10] J. He, S. Zhang, M. Yang, Y. Shan, and T. Huang. Bi-directional cascade network for perceptual edge detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3828–3837, 2019.
- [11] K. Hohm, M. Storath, and A. Weinmann. An algorithmic framework for Mumford–Shah regularization of inverse problems in imaging. *Inverse Problems*, 31(11):115011, 2015.
- [12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [13] H.T.V. Le, A. Repetti, and N. Pustelnik. Unfolded proximal neural networks for robust image Gaussian denoising. *IEEE Trans. Image Process.*, 33:4475–4487, 2024.
- [14] D. B. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 1989.
- [15] M. Pu, Y. Huang, Y. Liu, Q. Guan, and H. Ling. EDTER: Edge detection with transformer. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1402–1412, 2022.
- [16] N. Pustelnik. On the primal and dual formulations of the Discrete Mumford–Shah functional. In *Proc. Int. Conf. Acoust., Speech Signal Process.*, pages 1–5, 2023.
- [17] A. Repetti and Y. Wiaux. Variable metric forward-backward algorithm for composite minimization problems. *SIAM Journal on Optimization*, 31(2):1215–1241, 2021.
- [18] M. Serres, M.-L. Zanota, R. Philippe, and V. Vidal. On the stability of taylor bubbles inside a confined highly porous medium. *International Journal of Multiphase Flow*, 85:157–163, 2016.
- [19] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang. DeepContour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3982–3991, 2015.
- [20] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 1395–1403, 2015.
- [21] Y. Ye, K. Xu, Y. Huang, R. Yi, and Z. Cai. DiffusionEdge: Diffusion probabilistic model for crisp edge detection. In *Proc. AAAI Conf. on Artificial Intelligence*, volume 38, pages 6675–6683, 2024.