Unfolded discrete Mumford-Shah functional for joint image denoising and edge detection

Hoang Trieu Vy Le*, Marion Foare**, Audrey Repetti‡, Nelly Pustelnik*
*ENS de Lyon, CNRS, Laboratoire de Physique, F-69342
**CPE Lyon, Univ. Lyon, Inria, ENSL, UCBL, CNRS, LIP, UMR 5668, F-69342
‡School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh EH14 4AS, United Kingdom

***Abstract*—This work focuses on joint image denoising and contour detection. On the one hand, contour detection on clean images (without noise) has been extensively studied from both variational formulation and deep learning perspectives. On the other hand, although the task of jointly denoising and contour detection has been largely considered in image processing literature using variational formulations, it has not been handled from the model-based neural network perspective. In this work, we propose an unfolded discrete Mumford-Shah procedure that enables us to bridge the gap between standard variational procedures designed to perform the combined denoising/edge detection task and black-box neural network designed for edge detection purpose.**

***Index Terms*—Edge detection, image denoising, Mumford-Shah, proximal algorithm, model-based neural networks**

I. Introduction

Edge detection plays a pivotal role in image processing and computer vision. It aims at extracting a structural information by delineating objects boundaries and significant intensity transitions, providing a compact yet powerful representation of visual data. Edge maps are then essential for numerous downstream tasks, such as image segmentation, object recognition, and scene understanding. For instance, edge detection is used in experimental physics to identify hydrodynamic regimes by analyzing the phases of contact areas [2]. More recently, in [3], edge information has played an important role as structure-guided priors in diffusion model.

Traditional edge detection methods, such as those relying on gradient computations [4], excel at localizing edges but are often not well adapted to handle challenging cases such as noisy scenes, shadows, or textures. To overcome these limitations, it is then necessary to use image regularization that remove useless details while preserving critical boundaries. In this context, variational approaches have been instrumental, including the Mumford–Shah (MS) model [5] and related methods proposed by Geman and Geman [6] and Blake and Zisserman (BZ) [7]. These methods enable smoothing homogeneous regions while preserving sharp boundaries of these regions. Variational approaches have further the advantage of providing theoretical guarantees for the output solution. However, as they are highly iterative and necessitate hyperparameter tuning, they can appear to be computationally inefficient. Consequently, deep learning (DL) methods for edge detection have emerged in the last decade, shifting from traditional handcrafted methods to data-driven approaches learning hierarchical features. Early CNN-based methods, such as DeepEdge [8] and DeepContour [9], improved upon previous techniques [4], [6], [7] but faced limitations in handling complex scenes. Advances like HED [10] and BDCN [11] introduced multi-scale learning and parameter optimization, enhancing accuracy. Transformer-based models, such as EDTER [12], leveraged self-attention to capture both local and global features but required post-processing to refine thick edge maps. Diffusion models like DiffEdge [13] addressed this issue by producing sharper edges without additional refinement. However, deep learning methods have grown increasingly complex, leading to higher computational costs and slower inference.

In the last decade, hybrid methods that combine variational methods and neural networks have become increasingly popular for image restoration, as they combine network efficiency with variational robustness and interpretability. Inspired by this success, we have recently proposed such a hybrid approach in [14] for edge detection in a noiseless context. Specifically, we derived an unfolded scheme that relies on an iterative procedure that mimics the BZ minimization strategy, combined with an edge detection layer. The resulting network offers a robust method for edge detection with a lighter architecture compared to other recent DL-based models.

In this work, we develop an unfolded network jointly handling the denoising and edge detection tasks, by introducing a novel framework grounded in the MS model. Specifically, we unfold an alternating proximal scheme (dubbed SL-PAM) designed for minimizing the non-convex Discrete MS (DMS) model proposed in [15]. By unfolding SL-PAM, our approach enhances image denoising and edge detection capabilities, improving edge granularity control in a natural manner without requiring additional post-processing step.

Contributions and outline – The contributions of this work is then the development of a novel unfolded optimization framework for edge detection for noisy images (Section III) relying on DMS solved with SL-PAM (Section II). Note that the proposed unfolded network hence solves the non-convex problem of jointly estimating the denoised image and its contours. We further evaluate the effectiveness of the proposed

This work is partly funded by the Fondation Simone et Cino Del Duca - Institut de France. The work of AR was partly funded by the EPSRC grant EP/X028860/1. The authors thank the Centre Blaise Pascal of ENS Lyon for the computation facilities. The platform uses SIDUS [1], which was developed by Emmanuel Quemener.

method through extensive experiments in Section IV, achieving competitive results compared to state-of-the-art variational approaches and deep learning models.

Notations – In the remainder of this paper we will use the following notations. Let $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_C) \in \mathbb{R}^{N\times C}$ be a matrix containing a vector-reshaped image $\mathbf{x}_c \in \mathbb{R}^N$ in each column $c \in \{1, \dots, C\}$ (corresponding to a channel such that $C = 3$ for RGB images) and N the number of pixels in a channel. Let $\mathbf{e} = (\mathbf{e}_1, \dots, \mathbf{e}_L) \in \mathbb{R}^{N\times L}$ be a matrix containing an edge map $\mathbf{e}_\ell = (\mathrm{e}_{n,\ell})_{1\le n\le N}$ in each column $\ell \in \{1, \dots, L\}$ where L is the number of edges. Let $\psi\colon \mathbb{R}^N \to (-\infty, +\infty]$ be a proper convex, lower semicontinuous function. The proximity operator of ψ at x is given by $\mathrm{prox}_\psi(\mathrm{x}) = \underset{\mathrm{v}\in\mathbb{R}^N}{\mathrm{argmin}}\ \psi(\mathrm{v}) + \frac{1}{2}\|\mathrm{v} - \mathrm{x}\|_2^2$.

II. MS functional and minimization algorithms

A. Discrete Mumford-Shah

The continuous MS model [5] aims to recover both a smooth image and its edges, from a possibly degraded input image. In its discrete version (DMS), proposed in [15], edges are defined between neighbouring pixels, leading to horizontal and vertical edges in the image. Specifically, considering a noisy image $\mathbf{z}$ to analyse, the DMS formulation is expressed as

$$\min_{\mathbf{x},\mathbf{e}} \mathcal{L}(\mathbf{x}, \mathbf{e}) := f(\mathbf{x}) + \beta h(\mathbf{x}, \mathbf{e}) + \lambda g(\mathbf{e}). \tag{1}$$

where $\mathcal{L}$ is the loss function to minimize, $f\colon \mathbb{R}^{N\times C} \to \mathbb{R}\colon \mathbf{x} \mapsto \frac{1}{2}\|\mathbf{x} - \mathbf{z}\|_2^2$ is the data fidelity term for denoising, $\beta, \lambda > 0$ are regularization parameters, and h and g are regularization terms defined as follows. The first term h imposes regularity on $\mathbf{x}$, penalizing variations everywhere except on the edge map $\mathbf{e}$, i.e.,

$$h(\mathbf{x}, \mathbf{e}) = \sum_{c=1}^{C} \|(1 - \mathbf{e}) \odot (\mathbf{D}\mathbf{x}_c)\|_2^2 \tag{2}$$

where $\odot$ denotes the element-wise multiplication, and $\mathbf{D}\colon \mathbb{R}^N \to \mathbb{R}^{N\times 2}$ denotes the linear operator computing the concatenation of the horizontal and vertical discrete gradient operators (i.e., the total variation linear operators). In other words, $\mathbf{D}$ maps an image $\mathbf{x}_c$ to the edge domain. The second term $g\colon \mathbb{R}^{N\times 2} \to (-\infty, +\infty]$ aims to promote sparsity on $\mathbf{e}$ by penalizing the edge length. For instance 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 [15].

Proposed multichannel edge model – In this work, instead of restricting the edge maps to horizontal and vertical edges in the image, we consider a linear operator $\mathbf{D}\colon \mathbb{R}^N \to \mathbb{R}^{N\times L}$ mapping images to a feature domain with L channels , i.e., we consider a multichannel edge map $\mathbf{e} = (\mathbf{e}_1, \dots, \mathbf{e}_L) \in \mathbb{R}^{N\times L}$. In this case, for simplicity, we assume that the regularization on $\mathbf{e}$, $g\colon \mathbb{R}^{N\times L} \to (-\infty, +\infty]$, is additively separable (i.e., $g(\mathbf{e}) = \sum_{\ell,n} g_{n,\ell}(e_{n,\ell})$; for instance the ℓ_1-norm).

B. Variational approach

A strategy to solve the minimization problem (1) is to use alternative minimization algorithms such as Proximal Alternating Linearized Minimization (PALM) [16] or Semi-Linearized Proximal Alternating Minimization (SL-PAM) [15]. While PALM alternates a proximal-gradient step over $\mathbf{x}$ and one over $\mathbf{e}$, SL-PAM relaxes the proximal-gradient step over $\mathbf{e}$ for a proximal one, allowing a wider choice of step-size and providing a more efficient algorithm. Following on from [17] where it was shown that between two algorithms minimizing the same problem, a fast algorithm was more efficient when unfolded, we will focus on SL-PAM in what follows.

The associated iterations can be reformulated as:

$$\begin{aligned}
&\text{For } k = 0, 1, \dots \\
&\left\lfloor \begin{aligned}
&\widetilde{\mathbf{x}}^{[k+1]} = \mathbf{x}^{[k]} - \frac{\beta}{c_k}\nabla_x h(\mathbf{x}^{[k]}, \mathbf{e}^{[k]}) \\
&\mathbf{x}^{[k+1]} = \mathrm{prox}_{c_k^{-1} f}\left(\widetilde{\mathbf{x}}^{[k+1]}\right) \\
&\mathbf{e}^{[k+1]} = \mathrm{prox}_{d_k^{-1}(\beta h(\mathbf{x}^{[k+1]}, \cdot) + \lambda g(\cdot))}\left(\mathbf{e}^{[k]}\right)
\end{aligned}\right.
\end{aligned} \tag{3}$$

where $\mathbf{x}^{[0]} \in \mathbb{R}^{N\times C}$, $\mathbf{e}^{[0]} \in \mathbb{R}^{N\times L}$. On the one hand, the data-term update (first two lines in (3)) simply leads to

$$\begin{cases}
\widetilde{\mathbf{x}}^{[k+1]} = \mathbf{x}^{[k]} - \dfrac{2\beta}{c_k}\mathbf{D}^\top\left((1 - \mathbf{e})^2 \odot \mathbf{D}\mathbf{x}^{[k]}\right), \\
\mathbf{x}^{[k+1]} = \dfrac{\mathbf{z} + c_k\widetilde{\mathbf{x}}^{[k+1]}}{c_k + 1}.
\end{cases} \tag{4}$$

On the other hand, the update of the edge variable in the last line of Algorithm (3) takes a closed form as g is additively separable, i.e., for every $(n, \ell) \in \{1, \dots, N\} \times \{1, \dots, L\}$,

$$e_{n,\ell}^{[k+1]} = \mathrm{prox}_{\frac{\lambda}{\widetilde{\beta}_{n,\ell}^{[k+1]} + d_k} g_{n,\ell}}\left(\frac{\widetilde{\beta}_{n,\ell}^{[k+1]} + d_k \mathbf{e}_{n,\ell}^{[k]}}{\widetilde{\beta}_{n,\ell}^{[k+1]} + d_k}\right) \tag{5}$$

where $\widetilde{\beta}_{n,\ell}^{[k+1]} = 2\beta\left[\mathbf{D}\mathbf{x}_c^{[k+1]}\right]_{n,\ell}^2$.

If, for every $k \in \mathbb{N}$, $c_k > 2\beta\|\mathbf{D}\|^2$ and $d_k > 0$, then the sequence $(\mathbf{x}^{[k]}, \mathbf{e}^{[k]})_{k\in\mathbb{N}}$ converges to a critical point of the minimization problem (1)-(2) [15, Prop. 2].

III. Mumford-Shah Proximal Neural Network

In this section, we propose to design an unfolded neural network based on the DMS structure described in Section II-A, which aims to estimate a denoised image $\widehat{\mathbf{x}}$ and associated edges $\widehat{\mathbf{e}} \in \mathbb{R}^N$ from a degraded input image $\mathbf{z}$.

Feedforward networks – The proposed neural network has a feed-forward structure. It means that it can be written as a concatenation of a fixed number K of layer operators $\mathrm{T}_{\mathrm{z},\Theta_k}\colon \mathbb{R}^{N\times C} \times \mathbb{R}^{N\times L} \to \mathbb{R}^{N\times C} \times \mathbb{R}^{N\times L}$, parametrized by the input noisy image z and learnable parameters Θ_k (e.g., hyper-parameters, convolution kernels). For each layer $k \in \{1, \dots, K\}$, the associated operator can be written as

$$\mathrm{T}_{\mathrm{z},\Theta_k}\colon (\mathbf{x}, \mathbf{e}) \in \mathbb{R}^{N\times C} \times \mathbb{R}^{N\times L} \mapsto \eta_k(\mathrm{W}_k(\mathbf{x}, \mathbf{e}) + \mathrm{b}_k)$$

involving a (learnable) operator W_k, a (learnable) bias b_k and a non-linear activation function η_k. In recent works [17]–[19], it has been shown that networks whose layers are based on proximal iterations achieve state-of-the-art performance in denoising or reconstruction. In this work we unfold the SL-PAM method described in Section II-B.

Proposed DMS−PNN network – We are now ready to describe the architecture of our unfolded network, dubbed DMS−PNN, that consists in unrolling the SL-PAM iterations in (3) for solving the general DMS model introduced in Section II-A with $\mathbf{e} \in \mathbb{R}^{N\times L}$. The DMS−PNN can be expressed as

$$\mathrm{d}^K_{\mathrm{z},\Theta}(\mathbf{x}_0, \mathbf{e}_0) = \mathrm{T}_{\mathrm{z},\Theta_K} \circ \cdots \circ \mathrm{T}_{\mathrm{z},\Theta_1}(\mathbf{x}_0, \mathbf{e}_0), \tag{6}$$

where $\mathbf{x}_0 = \mathbf{z}$ and $\mathbf{e}_0 = \mathbf{0}$. For every $k \in \{1, \dots, K\}$, the layer $\mathrm{T}_{\mathrm{z},\Theta_k}$ alternates between a layer dedicated to updating the image, taking into account both the edges and the contents of the image, and a second layer for updating edge information, i.e., for every $k \in \{0, \dots, K-1\}$,

$$\begin{aligned}\left(\mathbf{x}^{[k+1]}, \mathbf{e}^{[k+1]}\right) &= \mathrm{T}_{\mathrm{z},\Theta_{k+1}}(\mathbf{x}^{[k]}, \mathbf{e}^{[k]}) \\ &= \mathrm{T}_{\mathrm{z},\Theta_{\mathbf{e},k}}\Big(\underbrace{\mathrm{T}_{\mathrm{z},\Theta_{\mathbf{x},k}}\left(\mathbf{x}^{[k]}, \mathbf{e}^{[k]}\right)}_{=\left(\mathbf{x}^{[k+1]}, \mathbf{e}^{[k]}\right)}\Big).\end{aligned} \tag{7}$$

According to Algorithm (3), each sub-layer is defined as

$$\begin{cases}\mathbf{x}^{[k+1]} = \eta_{\mathbf{x},k}\left(\mathrm{W}_{\mathbf{x},k}(\mathbf{x}^{[k]}, \mathbf{e}^{[k]}) + \mathrm{b}_{\mathrm{z},\mathbf{x},k}\right), \\ \mathbf{e}^{[k+1]} = \eta_{\mathbf{e},k,\mathbf{x}^{[k+1]}}\left(\mathrm{W}_{\mathbf{e},k}(\mathbf{x}^{[k+1]}, \mathbf{e}^{[k]}) + \mathrm{b}_{\mathbf{e},k}\right),\end{cases} \tag{8}$$

with

$$\begin{cases}\mathrm{W}_{\mathbf{x},k}(\mathbf{x}^{[k]}, \mathbf{e}^{[k]}) = \frac{c_k}{c_k+1}\left(\mathbf{x}^{[k]} - \frac{2\beta_k}{c_k}\mathbf{D}_k^\top\left((1-\mathbf{e}^{[k]})^2 \odot \mathbf{D}_k\mathbf{x}^{[k]}\right)\right), \\ \mathrm{b}_{\mathrm{z},\mathbf{x},k} = \frac{\mathrm{z}}{c_k+1}, \\ \eta_{\mathbf{x},k} = \mathrm{Id}, \\ \mathrm{W}_{\mathbf{e},k}(\mathbf{x}^{[k+1]}, \mathbf{e}^{[k]}) = \left(\dfrac{\widetilde{\beta}^{[k]}_{n,\ell} + d_k e^{[k]}_{n,\ell}}{\widetilde{\beta}^{[k]}_{n,\ell} + d_k}\right)_{1\le n\le N, 1\le \ell\le L}, \\ \mathrm{b}_{\mathbf{e},k} = 0, \\ \eta_{\mathbf{e},k,\mathbf{x}^{[k+1]}} = \left(\mathrm{prox}_{\frac{\lambda_k}{\widetilde{\beta}^{[k]}_{n,\ell}+d_k} g_{n,\ell}}\right)_{1\le n\le N, 1\le \ell\le L},\end{cases} \tag{9}$$

for $\widetilde{\beta}^{[k]}_{n,\ell} = 2\beta_k\left[\mathbf{D}_k\mathbf{x}^{[k+1]}\right]^2_{n,\ell}$. Hence, by construction DMS−PNN is an unfolded scheme with learnable parameters $\Theta = \{\mathbf{D}_1, \dots, \mathbf{D}_K, (\beta_k, \lambda_k)_{0\le k\le K-1}\}$.

IV. Numerical experiments

A. Training settings

Dataset – Our experiments are performed on the BSD500 dataset [20], containing multiple hand-drawn groundtruth contours for each image. We denote $(\overline{\mathbf{e}}_s, \mathbf{z}_s)_{s\in\mathbb{I}}$ our *training set*, and $(\overline{\mathbf{e}}_s, \mathbf{z}_s)_{s\in\mathbb{J}}$ our *test set*. For both sets, $\overline{\mathbf{e}}_s$ are the exact edges obtained by stacking all the provided groundtruth annotations, and $\mathbf{z}_s$ consists of an input image degraded with Gaussian noise with standard deviation $\delta_s > 0$ distributed according to a uniform distribution in $(0, 0.1)$.

Training setting – We consider a hybrid loss function combining the weighted binary cross entropy (BCE) loss [10], and the MSE loss. These are common choices in the literature for training networks for edge detection and for denoising, respectively. The resulting training problem is thus of the form

$$\widehat{\Theta} \in \operatorname*{Argmin}_{\Theta} \sum_{s\in\mathbb{I}} \mathrm{BCE}(\overline{\mathbf{e}}_s, \mathbf{e}^K_s) + \alpha\mathrm{MSE}(\overline{\mathbf{x}}_s, \mathbf{x}^K_s) \tag{10}$$

where $(\mathbf{x}^K_s, \widetilde{\mathbf{e}}^K_s) = \mathrm{d}^K_{\mathrm{z},\Theta}(\mathbf{z}_s, \mathbf{0})$ is the output of the DMS−PNN, $\mathbf{e}^K_s = \sum_{\ell=1}^L \widetilde{\mathrm{e}}^K_{n,\ell}/L$ is the mean edge map obtained over the L channels of $\widetilde{\mathbf{e}}^K_s$, $\alpha > 0$, and the BCE loss is given by

$$\begin{aligned}\mathrm{BCE}(\mathbf{e}, \mathbf{e}^K) = \sum_{n=1}^N -\omega_n\Big(&\overline{\mathrm{e}}_n \log(\mathrm{e}^K_n) \\ &+ (1-\mathrm{e}^K_n)\log(1-\overline{\mathrm{e}}_n)\Big),\end{aligned}$$

where $(\omega_n)_{1\le n\le N} \in \mathbb{R}^N$ denotes the class-balance weights between the edge pixel set $E_+ = \{n \in \{1, \dots, N\} | \overline{\mathrm{e}}_n = 1\}$ and non-edge pixel set $E_- = \{n \in \{1, \dots, N\} | \overline{\mathrm{e}}_n = 0\}$, that is, $\omega_n = |E_-|/|E_+ \cup E_-|$ when $\overline{\mathrm{e}}_n = 1$ and $\omega_n = |E_+|/|E_+ \cup E_-|$ when $\overline{\mathrm{e}}_n = 0$.

The training is then performed using the Adam optimizer [21] on Pytorch, with learning rate 10^{-3}, batch size 3, and image sizes 321×481 or 481×321.

Architecture choices – We incorporate the noise level of the data z_s in our architecture by multiplying the learned parameters $(\lambda_k, \beta_k)_{0\le k\le K-1}$ by the standard deviation δ_s. Hence the noise level is an additional input to our network, similarly to noise-aware networks such as DRUnet [22] and PNNs [18], [23]. The step sizes parameters $(c_k, d_k)_{0\le k\le K-1}$ are chosen to satisfy theoretical conditions (see Section II-B): $c_k = 2\delta\beta_k\|\mathbf{D}_k\|^2$ and $d_k = 2\delta\lambda_k\eta\|D_k\|^2$ with $\eta = 2.10^{-4}$.

B. Results and analysis

Impact of the training loss – We evaluate the training efficiency for two choices of α in (10): $\alpha = 0$ (loss only based on BCE) and $\alpha = 0.15$ (loss that combined MSE and BCE). The results are displayed in Figure 1. We observe that while the edge detection is only slightly impacted by the choice of α, the performance in terms of denoising are improved when $\alpha > 0$, motivating the use of the proposed hybrid loss.

Impact of the noise-aware parameter δ – In Figure 1 we evaluate the impact of the free parameter δ on the output. Based on the previous conclusions, we focus on the results obtained with $\alpha = 0.1$. We observe that δ acts as a threshold parameter on the edges: smaller is δ, and the more the details in the edge map. We observe that large δ values smooth the denoised images, and hence degrade the PSNR values.

Comparisons with state-of-the-art – In Figure 2, we focus on edge-detection performance and compare the results obtained with standard DMS based on SL-PAM iterations [15] (cf. Section II), DiffEdge [13] (only designed for edge detection), and with the proposed DMS−PNN (with $\delta = 0.001$ and $\delta = 0.1$). The number of parameters for each approach as well as their inference time are summarised in the table below:

	$\lvert\Theta\rvert$	Inference time (s)
DMS [15]	–	12 ± 2.1
DiffusionEdge [13]	$137, 142, 150$	10 ± 2.3
DMS-PNN	$\mathbf{17,320}$	$\mathbf{0.09 \pm 0.03}$

We observe that state-of-the-art methods provide less detailed edges. Especially DiffEdge, a pure black-box neural network architecture whose goal only focuses on estimated

Fig. 1. **Loss impact (choice of paramter α) –** First column: Noisy image $\mathbf{z}$ (top) with standard deviation 0.025 and ground-truth edges $\overline{\mathbf{e}}$ (bottom). Second to fourth columns: Results obtained with `DMS-PNN` for different values of $\delta = \{0.001, 0.01, 0.1\}$ (the noise-aware parameter given to `DMS-PNN` when evaluated). The first two rows show results obtained with $\alpha = 0$ (i.e., purely BCE); and the last two rows show results with $\alpha = 0.1$ (i.e., hybrid BCE-MSE). The performance are evaluated in terms of standard Peak Signal to Noise Ratio (PSNR) and Cross Entropy (CE) computed as $\mathrm{CE}(\mathbf{e}, \mathbf{e}^K) = -\sum_{n=1}^{N} \overline{\mathrm{e}}_n \log(\mathrm{e}_n^K)$.

ground-truth data. In comparison the standard DMS and the proposed `DMS-PNN` capture more detailed edges.

V. Conclusion

In this work we derive the first model-based neural network for joint image denoising and edge detection, designing a tailored proximal unfolded neural network. We illustrate the impact of the loss and of a noise-aware hyperparameter through simulations. We also illustrate the benefit of the proposed `DMS-PNN` compared to state-of-the-art methods, both in terms of performance and computational efficiency.

References

[1] E. Quemener and M. Corvellec, "SIDUS—the Solution for Extreme Deduplication of an Operating System," *Linux J.*, vol. 2013, no. 235, 2013.

[2] 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*, vol. 85, pp. 157–163, 2016.

[3] C. Cao, Q. Dong, and Y. Fu, "Zits++: Image inpainting by improving the incremental transformer on structural priors," *IEEE Trans. Pattern Anal. Match. Int.*, , no. 45, pp. 12667–12684, 2023.

[4] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Anal. Match. Int.*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[5] D. B. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, 1989.

[6] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Match. Int.*, pp. 721–741, 1984.

[7] A. Blake and A. Zisserman, *Visual reconstruction*, MIT press, 1987.

[8] 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*, 2015, pp. 4380–4389.

[9] 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*, 2015, pp. 3982–3991.

[10] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1395–1403.

[11] 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*, 2019, pp. 3828–3837.

[12] 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*, 2022, pp. 1402–1412.

[13] Y. Ye, K. Xu, Y. Huang, R. Yi, and Z. Cai, "DiffusionEdge: Diffusion probabilistic model for crisp edge detection," in *Proc. AAAI Confe. on Artificial Intelligence*, 2024, vol. 38, pp. 6675–6683.

[14] H.T.V. Le, M. Foare, A. Repetti, and N. Pustelnik, "Embedding

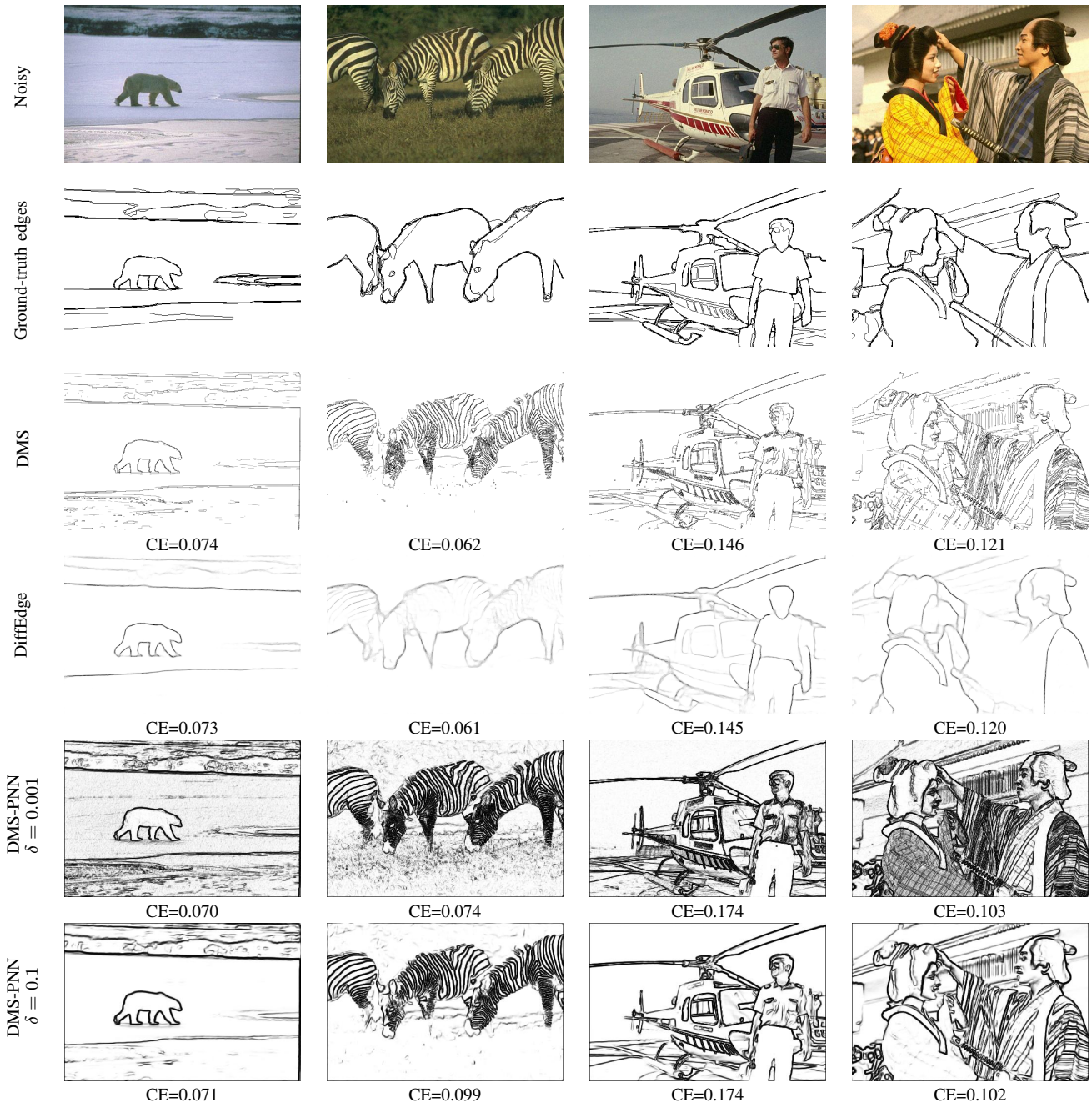

Fig. 2. **State-of-the-art comparison –** Row 1: Noisy image $\mathbf{z}$ with standard deviation 0.025. Row 2: Ground-truth edges $\overline{\mathbf{e}}$. Row 3: Estimated edges with DMS based on SL-PAM [15]. Row 4: Estimated edges with DiffEdge [13]. Rows 5-6: Estimated edges with the `DMS-PNN` for $\delta = 0.001$ and $\delta = 0.1$.

Blake-Zisserman regularization in unfolded proximal neural networks for enhanced edge detection," *IEEE Signal Process. Lett.*, 2025.

[15] M. Foare, N. Pustelnik, and L. Condat, "Semi-linearized proximal alternating minimization for a discrete Mumford–Shah model," *IEEE Trans. Image Process.*, vol. 29, no. 1, pp. 2176–2189, 2019.

[16] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1, pp. 459–494, 2014.

[17] H.T.V. Le, M. Foare, and N. Pustelnik, "Proximal based strategies for solving Discrete Mumford-Shah with Ambrosio–Tortorelli penalization on edges," *Signal Process. Lett.*, vol. 29, pp. 952–956, 2022.

[18] H.T.V. Le, A. Repetti, and N. Pustelnik, "Unfolded proximal neural networks for robust image Gaussian denoising," *IEEE Trans. Image Process.*, vol. 33, pp. 4475–4487, 2024.

[19] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, 2021.

[20] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Match. Int.*, vol. 33, no. 5, pp. 898–916, 2010.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, "Plug-and-play image restoration with deep denoiser prior," *IEEE Trans. Pattern Anal. Match. Int.*, vol. 44, no. 10, pp. 6360–6376, 2021.

[23] A. Repetti, M. Terris, Y. Wiaux, and J.-C. Pesquet, "Dual forward-backward unfolded network for flexible plug-and-play," in *Proc. 30th European Signal Processing Conference (EUSIPCO)*, 2022, pp. 957–961.