Impulse Noise Image Restoration Using Nonconvex Variational Model and Difference of Convex Functions Algorithm

Benxin Zhang, Guopu Zhu, *Senior Member, IEEE*, Zhibin Zhu, Hongli Zhang, *Member, IEEE*, Yicong Zhou, *Senior Member, IEEE*, and Sam Kwong, *Fellow, IEEE*

***Abstract*—In this article, the problem of impulse noise image restoration is investigated. A typical way to eliminate impulse noise is to use an L_1 norm data fitting term and a total variation (TV) regularization. However, a convex optimization method designed in this way always yields staircase artifacts. In addition, the L_1 norm fitting term tends to penalize corrupted and noise-free data equally, and is not robust to impulse noise. In order to seek a solution of high recovery quality, we propose a new variational model that integrates the nonconvex data fitting term and the nonconvex TV regularization. The usage of the nonconvex TV regularizer helps to eliminate the staircase artifacts. Moreover, the nonconvex fidelity term can detect impulse noise effectively in the way that it is enforced when the observed data is slightly corrupted, while is less enforced for the severely corrupted pixels. A novel difference of convex functions algorithm is also developed to solve the variational model. Using the variational method, we prove that the sequence generated by the proposed algorithm converges to a stationary point of the nonconvex objective function. Experimental results show that our proposed algorithm is efficient and compares favorably with state-of-the-art methods.**

***Index Terms*—Difference of convex functions algorithm (DCA), image restoration, impulse noise, nonconvex optimization model.**

Manuscript received 29 May 2022; revised 26 August 2022 and 25 October 2022; accepted 21 November 2022. Date of publication 15 December 2022; date of current version 18 March 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB3102900; in part by the National Natural Science Foundation of China under Grant 11901137, Grant 62172402, Grant 61872350, and Grant 61967004; in part by the Hong Kong Innovation and Technology Commission (InnoHK Project CIMDA); in part by the Hong Kong General Research Fund (GRF)-University Grants Committee (UGC) under Grant 9042816 (CityU 11209819) and Grant 9042958 (CityU 11203820); in part by the Science and Technology Development Fund, Macau, SAR, under Grant 0049/2022/A1; in part by the University of Macau under Grant MYRG2022-00072-FST; and in part by the Fundamental Research Funds for the Central Universities under Grant FRFCU5710011322. This article was recommended by Associate Editor Y. Xia. *(Corresponding author: Guopu Zhu.)*

Benxin Zhang is with the School of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin 541004, China (e-mail: bxzhang@guet.edu.cn).

Guopu Zhu and Hongli Zhang are with the School of Cyberspace Security, Harbin Institute of Technology, Harbin 150001, China (e-mail: guopu.zhu@hit.edu.cn; zhanghongli@hit.edu.cn).

Zhibin Zhu is with the School of Mathematics and Computing Science, Guilin University of Electronic Technology, Guilin 541004, China (e-mail: zhuzb@guet.edu.cn).

Yicong Zhou is with the Department of Computer and Information Science, University of Macau, Macau, China (e-mail: yicongzhou@um.edu.mo).

Sam Kwong is with the Department of Computer Science, City University of Hong Kong, Hong Kong, China (e-mail: cssamk@cityu.edu.hk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2022.3225525.

Digital Object Identifier 10.1109/TCYB.2022.3225525

I. Introduction

IMAGE restoration is a classical inverse problem and plays an important role in the field of image processing. Specifically, we consider the problem of restoring an image corrupted by impulse noise. Mathematically, the image degradation model can be formulated as $f = N_{\text{imp}}(Ku)$, where $f \in R^m$ is the damaged image, N_{imp} means that the degradation is caused by impulse noise, $K \in R^{m\times n}$ is an operator, such as convolution and wavelet transform, and $u \in R^n$ is the original image. Then, our main aim is to recover the original image from the damaged image.

Impulse noise often emerges due to transmission errors or faulty memory locations of hardware. It will damage only a part of the pixels of original image while not affecting the other image pixels. A popular denoising method is the median filter [1], which has high computational efficiency and favorable denoising power. However, the edge information of restored images is lost after median filtering. In order to preserve the edges, variational approaches have been proposed as an important class of image restoration methods, by which the original image u is recovered by minimizing the energy function. Generally, the energy function of the minimization problem consists of the data fidelity term and regularization term.

The data fidelity term is given by the noise type. According to the statistical property of noise, we can derive the corresponding form. For Gaussian noise, the data fidelity function is usually L_2 norm [2]. For non-Gaussian noise, L_1 norm [3] is suitable. It is well known that for impulse noise, the fidelity term is L_1 norm [4]. But, L_1 norm yields biased estimators in statistics. Especially, for high-level impulse noise, the data fidelity term with L_1 norm performs poorly. Gu et al. [5] introduced a smoothly clipped absolute deviation (SCAD) function for data fitting term, which processes the desirable oracle property. Some other nonconvex fidelity terms have been discussed in [6], [7], and [8], including the exponential type, Geman function, log penalty, and L_0 norm. These nonconvex terms are particularly suitable for restoring the images corrupted with high-level impulse noise.

The regularization term is related to the priori knowledge of the image, such as textures and edges, which are important information and structures. A general form of this term is a composition of the potential function and gradient operator. The classical one is Tikhonov regularization [9], of which the function is quadratic and smooth. It can be easily minimized using a smooth optimization method. However, this regularization function often eliminates edges and texture details. To deal with this shortcoming, total variation (TV) regularization was proposed in [2] and has been proved very successful. TV term is composed of the L_1 norm and gradient operator. It captures the gradient sparsity of an image and has the edge-preserving ability. Also, some variants of TV have been studied in the past twenty years. Recently, the Potts model [10], which adopts L_0 norm to ensure sparseness, has received much attention. However, the global optimal solution of the model is difficult to obtain. So, it is expected to find new penalties that have the advantages of L_1 and L_0 norm while avoiding their disadvantages. There are many choices for nonconvex penalties, for example, capped-L1 [11], the minimax concave penalty (MCP) [12], and Log-norm penalty [13]. These regularization functions were introduced to enhance gradient sparsity. Theoretical analysis and experimental results indicate that nonconvex regularization can obtain a better performance than convex regularization.

By using TV and L_1 norm data fitting terms, the TVL1 model is proposed for impulse noise removal [3]. Some efficient algorithms, including the alternating direction method of multipliers (ADMMs) [14] and the primal–dual method [15], have been designed for solving the TVL1 optimization problem. Nikolova et al. [4] proposed a new model that combines L_1 norm data fitting and nonconvex regularizer. The nonconvex regularizer offers more possibilities to recover high-quality image. By adopting TV as the regularization term and SCAD as the data fitting function, Gu et al. introduced a new model TVSCAD in [5]. This nonconvex model can achieve higher performance than TVL1, and a difference of convex functions algorithm (DCA) was designed to solve it. Zhang et al. [7] proposed a simple optimization model that makes use of a nonconvex log function for data fitting term and TV regularization. Compared to TVSCAD, this model is easy to choose the scalar parameter. They also developed a new DCA with an adaptive proximal parameter. Zhang et al. [6] introduced some nonconvex-TV models with a nonconvex potential function, and gave a proximal linearized minimization algorithm. Yuan and Ghanem [8] proposed a sparse optimization method, which applies TV regularization and L_0 norm data fidelity for impulse noise removal. To solve the nonconvex optimization model, the authors reformulated the problem as a mathematical programming with equilibrium constraints. Very recently, Cui and Fan [16] proposed a nonconvex regularization term plus nonconvex data fitting term model. Further, the authors designed an alternating direction minimization method to solve the optimization problem.

In this article, the issue of impulse noise image restoration is investigated. To solve this issue, a new nonconvex variational model is developed considering that the nonconvex plus nonconvex model may be more suitable for reducing artifacts and eliminating impulse noise. Generally, it is a challenging task to find an efficient algorithm to solve the optimization problem related to nonconvex variational model. To address this, a novel DCA is presented. The main contributions of this work are as follows:

First, a new variational model, which is nonconvex in terms of both TV term and data fidelity term, is proposed for the removal of impulse noise. In the proposed model, the nonconvex TV eliminates artifacts while preserving the sharpness and smoothness of the restored image, meanwhile the nonconvex data fidelity effectively removes the impulse noise.

Second, a novel DCA with an L_1 norm proximal term is presented to solve the proposed nonsmooth and nonconvex variational model. To the best of our knowledge, L_1 proximal technique has not yet been used in DC programming. With the help of the η stationary point, the convergence of the proposed algorithm is also proved. The convergence analysis extends the existing results of general DC programming, thus also makes a theoretical contribution to this kind of issues.

Third, our new model can deal with different image processing tasks, such as image deblurring, image recovery, and image denoising (note that the noise types include salt-and-pepper (SP) impulse noise, random-valued (RV) impulse noise, Gaussian plus SP and RV impulse noise, and Gaussian mixture noise). Experimental results show that the proposed algorithm outperforms other competitors for suppressing impulse noise.

The remainder is organized as follows. In Section II, the new nonconvex model is shown. In Section III, a novel DCA is proposed to deal with this optimization model. The convergence of DCA is established in Section IV. In Section V, experimental results for the proposed model and algorithm demonstrate encouraging performance. In Section VI, the conclusion is presented.

II. Motivation and New Model

By combining TV regularization and L_1 norm penalized data term, the minimization of TVL1 model for impulse noise removal can be written as

$$\min_u \ \|u\|_{TV} + \mu\|Ku - f\|_1 \tag{1}$$

where $\mu > 0$ is a regularization parameter, and $\|\cdot\|_{TV}$ is TV norm, that is, $\|\nabla u\|_1$. Using the finite difference operation, TV can be discretized into isotropic and anisotropic forms

$$\|u\|_{TV_{\text{iso}}} := \|Du\|_2, \ \ \|u\|_{TV_{\text{aniso}}} := \|Du\|_1$$

where ∇ is gradient operator and D denotes a first-order finite difference of u at every pixel. It is known that TVL1 model works reasonably well only for low-level impulse noise, and performs poorly for high-level impulse noise. The reason may be that both the corrupted data and the noise free data are equally penalized in data fitting, leading to significant difficulty in balancing regularization and data fitting. Moreover, the L_1 norm penalty is known to yield biased estimators in statistics. The solution of the TVL1 model substantially deviates from both the data acquisition model and the prior model and, thus, is suboptimal. Then, some nonconvex approaches and correction procedures are proposed.

Yuan and Ghanem [8] proposed the following sparse optimization L0TV model:

$$\min_u \ \mu\|Ku - f\|_0 + \|\nabla u\|_1 \tag{2}$$

which takes L_0 norm as a data fidelity term and TV term as a regularization term. Model (2) is suitable for image restoration under impulse noise. While the problem described by (2) is a combinatorial optimization problem (NP hard), and it is difficult to obtain its optimal solution. In [17], a corrected TVL1 approach was proposed by solving a series equivalence problems of TVL1. Base on the work of Bai et al. [17], Gu et al. [5] introduced the TVSCAD model that includes TV and SCAD function. The TVSCAD model can be described as

$$\min_u \|u\|_{TV} + \mu \Phi_\gamma (Ku - f) \tag{3}$$

where Φ_γ is defined as

$$\Phi_\gamma (y) = \sum_i^m \phi_\gamma (y_i), \;\; y \in R^m$$

and

$$\phi_\gamma (t) = \begin{cases} |t|, & \text{if } |t| \le \gamma_1 \\ \frac{-t^2+2\gamma_2|t|-\gamma_1^2}{2(\gamma_2-\gamma_1)}, & \text{if } \gamma_1 < |t| < \gamma_2 \\ \frac{\gamma_1+\gamma_2}{2}, & \text{if } |t| \ge \gamma_2 \end{cases}$$

where $y = [y_1, \ldots, y_m]^T$ denotes the function variable of $\Phi_\gamma(y)$, t is the function variable of $\phi_\gamma(t)$, $|\cdot|$ denotes the absolute value, $\gamma := (\gamma_1, \gamma_2)$ is a pair of parameters, and $\gamma_2 > \gamma_1 > 0$. Then, a DCA is proposed to solve this model. The resulting subproblem is solved by ADMM, and the global convergence is also established. Zhang et al. [6] proposed a nonconvex model, which is defined as

$$\min_u \;\mu P(Ku - f) + \|u\|_{TV} \tag{4}$$

where P is a potential data fitting function, such as the exponential-type (ET) function. Models (3) and (4) can be considered as the relaxation of (2). Such models can only lead to suboptimal estimation since they give approximate solutions of the problem of (2). Cui and Fan [16] introduced a model by adopting the nonconvex data fitting and regularization terms. Their model is expressed as

$$\min_u \;\mu P(Ku - f) + Q(\nabla u) \tag{5}$$

where P and Q are semialgebraic functions, and both of them can be rewritten as the sum of two functions. By the semiconvergence property of denoising problems, the optimal estimation of (5) converges toward the desired solution first, and then diverges from it after a number of iterations.

The data fitting term containing nonconvex functionals has been used for image restoration under impulsive noise. However, TV regularization was used as the regularizer in these works [5], [6], [8]. Hence, the artifacts may still exist in the restored image. Considering the effectiveness of the artifact removal of nonconvex regularization [16] and the robustness of nonconvex data fidelity to outliers, we propose a new model that combines the nonconvex data fitting term and nonconvex regularization term. The model is given as

$$\min_u \;\mu \Phi_\gamma (Ku - f) + \Psi_s(\nabla u) \tag{6}$$

where Ψ_s is defined as

$$\Psi_s(v) = \sum_i^n \psi_s(v_i), \;\; v \in R^n$$

and

$$\psi_s(t) = \frac{1}{s}\mathrm{Log}(1 + s|t|)$$

where s is a parameter larger than 0, and $v = [v_1, \ldots, v_n]^T$ denotes the function variable of $\Psi_s(v)$. The motivation of using the SCAD function here is to enforce less or even no data fitting and more regularization whenever $(Ku)_i$ deviates significantly from f_i. This is quite reasonable because the ith pixel is more likely to be corrupted in such a case. For the ith pixel such that $(Ku - f)_i$ is sufficiently small, the absolute penalty is kept. The structure of SCAD can discriminate the difference between the corrupted and uncorrupted pixels and provide good restorations. Moreover, the SCAD and Log penalty functions make the resulting estimator possesses three desired properties, that is, continuity, sparsity, and unbiasedness [5]. Hence, the solution of (6) is optimal and robust.

III. Algorithm

In this section, we first introduce some preliminaries that will be used in the proposed approach. Then, a novel DCA is presented.

A. Some Properties of Φ_γ and Ψ_s

We show some properties of Φ_γ and Ψ_s, which are important for a constructive algorithm. First, we consider the functions $\nu_\gamma(t)$ and $\varphi_s(t)$ that can be induced by ϕ_γ and ψ_s, that is

$$\nu_\gamma(t) = |t| - \phi_\gamma(t), \gamma > 0 \tag{7}$$
$$\varphi_s(t) = |t| - \psi_s(t), \;\; s > 0. \tag{8}$$

From [13], $\nu_\gamma(t)$ and $\varphi_s(t)$ are continuously differentiable and convex on R.

Note that in order to apply the difference of convex functions programming to Φ_γ and Ψ_s, $\nu_\gamma(t)$ and $\varphi_s(t)$ are defined in (7) and (8), respectively. Let $\nu_\gamma(t)$ be given by (7), then set $\Xi_\gamma : R^m \to R$ as

$$\Xi_\gamma(y) = \sum_i^m \nu_\gamma(y_i). \tag{9}$$

Similarly, $\varphi_s(t)$ is given by (8). Set Θ_s:$R^n \to R$ as

$$\Theta_s(v) = \sum_i^m \varphi_s(v_i). \tag{10}$$

Note that Ξ_γ and Θ_s are continuously differentiable and convex [13].

From (9) and (10), we also can obtain

$$\Phi_\gamma(y) = \|y\|_1 - \Xi_\gamma(y) \tag{11}$$
$$\Psi_s(v) = \|v\|_1 - \Theta_s(v). \tag{12}$$

Hence, we can rewrite the proposed model (6) as a DC programming.

B. DC Programming and Novel DCA

DC programming is used to minimize a problem which can be written as the difference of convex functions [18], [19]. By (11) and (12), we can decompose SCAD and log functions as the difference of convex functions, that is

$$\Phi_\gamma(Ku-f) = \|Ku-f\|_1 - \Xi_\gamma(Ku-f)$$

and

$$\Psi_s(v) = \|\nabla u\|_1 - \Theta_s(\nabla u).$$

By integrating the above two equations into (6), we obtain

$$\min_u \ \|\nabla u\|_1 + \mu\|Ku-f\|_1 - \big(\Theta_s(\nabla u) + \mu\Xi_\gamma(Ku-f)\big). \quad (13)$$

Let

$$g(u) = \|\nabla u\|_1 + \mu\|Ku-f\|_1$$

and

$$h(u) = \Theta_s(\nabla u) + \mu\Xi_\gamma(Ku-f)$$

then, (13) can be rewritten as the difference of $g(u)$ and $h(u)$

$$\min_u F(u) = g(u) - h(u). \quad (14)$$

So, (14) is a typical DC programming.

It is known that the aim of DCA is to solve the DC programming and its dual problem. Generally, DCA optimizes (14) via linearizing $h(u)$ and solving the following convex problem:

$$u_{k+1} = \arg\min_u\{g(u) - (h(u_k) + \langle\nabla h(u_k), u-u_k\rangle)\} \quad (15)$$

where $\langle\cdot\rangle$ denotes the inner product between two vectors. Gu et al. [5] added an L_2 norm proximal term to (15) and their convex problem is

$$u_{k+1} = \arg\min_u\Big\{g(u) - (h(u_k) + \langle\nabla h(u_k), u-u_k\rangle) + \frac{\eta}{2}\|u-u_k\|^2\Big\} \quad (16)$$

where $\eta > 0$. Different from the above scheme, we introduce a novel DCA with the L_1 norm proximal term as follows:

$$u_{k+1} = \arg\min_u\{g(u) - (h(u_k) + \langle\nabla h(u_k), u-u_k\rangle) + \eta\|u-u_k\|_1\}. \quad (17)$$

Then, our DCA for solving (6) is described as

$$\begin{aligned} u_{k+1} = \arg\min_u\{&\|Du\|_2 + \mu\|Ku-f\|_1 \\ &- \big\langle D^T\nabla\Theta_s(Du_k) + \mu K^T\nabla\Xi_\gamma(Ku_k-f), u-u_k\big\rangle \\ &+ \eta\|u-u_k\|_1\}. \end{aligned} \quad (18)$$

Note that the isotropic discretization TV is adopted in this article and the case of anisotropic TV is completely similar. The objective function of (18) is convex and has a global optimal solution.

In order to use the popular ADMM [20], [21], [22] for solving problem (18), we express (18) as an equivalent form

$$\begin{aligned} u_{k+1} = \arg\min_u\{&\|Du\|_2 - \langle\nabla\Theta_s(Du_k), Du\rangle \\ &+ \mu\|Ku-f\|_1 - \mu\big\langle\nabla\Xi_\gamma(Ku_k-f), Ku-f\big\rangle \\ &+ \eta\|u-u_k\|_1\}. \end{aligned} \quad (19)$$

By introducing free variables w, z, d, p, and q, and defining $w = Du, z = Ku-f, p_k = \nabla\Theta_s(w_k), q_k = \nabla\Xi_\gamma(z_k), d = u$, (19) can be rewritten as

$$\begin{aligned} \min_{u,w,z,d} \quad & \|w\|_2 - \langle p_k, w\rangle + \mu(\|z\|_1 - \langle q_k, z\rangle) + \eta\|d-u_k\|_1 \\ \text{s.t.} \quad & w = Du, z = Ku-f, d = u. \end{aligned} \quad (20)$$

The augmented Lagrangian function of (20) is

$$\begin{aligned} \mathcal{L}(w,z,d,u,\lambda_w,\lambda_z,\lambda_d) &= \|w\|_2 - \langle p_k, w\rangle - \lambda_w^T(w-Du) \\ &+ \frac{\beta_w}{2}\|w-Du\|^2 + \mu(\|z\|_1 - \langle q_k, z\rangle) \\ &- \lambda_z^T(z-(Ku-f)) + \frac{\beta_z}{2}\|z-(Ku-f)\|^2 \\ &+ \eta\|d-u_k\|_1 - \lambda_d^T(d-u) + \frac{\beta_d}{2}\|d-u\|^2 \end{aligned}$$

where $\beta_w, \beta_z, \beta_d > 0$ are penalty factors, and $\lambda_w \in R^{2n}$, $\lambda_z \in R^m$, and $\lambda_d \in R^n$ are multipliers. By (20), we have the following ADMM when initial points start at u_0 and λ_w^0, λ_z^0, and λ_d^0:

$$\Big(w^{j+1}, z^{j+1}, d^{j+1}\Big) = \arg\min_{w,z,d}\mathcal{L}\Big(w,z,d,u^j,\lambda_w^j,\lambda_z^j,\lambda_d^j\Big) \quad (21)$$

$$u^{j+1} = \arg\min_u\mathcal{L}\Big(w^{j+1}, z^{j+1}, d^{j+1}, u, \lambda_w^j, \lambda_z^j, \lambda_d^j\Big) \quad (22)$$

$$\begin{cases} \lambda_w^{j+1} = \lambda_w^j - \alpha\beta_w\big(w^{j+1} - Du^{j+1}\big) \\ \lambda_z^{j+1} = \lambda_z^j - \alpha\beta_z\big(z^{j+1} - \big(Ku^{j+1} - f\big)\big) \\ \lambda_d^{j+1} = \lambda_d^j - \alpha\beta_d\big(x^{j+1} - u^{j+1}\big) \end{cases} \quad (23)$$

where $\alpha \in (0, (1+\sqrt{5})/2)$. In fact, the scheme of (21)–(23) is ADMM for two blocks of variables (w, z, d) and u. Hence, the convergence of this ADMM can be guaranteed by the classical results in [23].

Next, we show how to solve the subproblems (21) and (22). In fact, w^{j+1}, z^{j+1}, and d^{j+1} are the solutions of proximity operators about $\|\cdot\|_2$ and $\|\cdot\|_1$. So, the w, z, d-subproblems in (21) have a closed form as

$$w^{j+1} = \max\Big\{\|Du^j + \frac{\lambda_w^j + p_k}{\beta_w}\|_2 - \frac{1}{\beta_w}, 0\Big\}\frac{Du^j + \frac{\lambda_w^j+p_k}{\beta_w}}{\Big\|Du^j + \frac{\lambda_w^j+p_k}{\beta_w}\Big\|_2} \quad (24)$$

$$\begin{aligned} z^{j+1} &= \max\Big\{|Ku^j - f + \frac{\lambda_z^j + \mu q_k}{\beta_z}| - \frac{\mu}{\beta_z}, 0\Big\} \\ &\quad \cdot\text{sign}\bigg(Ku^j - f + \frac{\lambda_z^j + \mu q_k}{\beta_z}\bigg) \end{aligned} \quad (25)$$

$$d^{j+1} = \max\bigg(|u^j + \frac{\lambda_d^j}{\beta_d} - u_k| - \frac{\eta}{\beta_d}, 0\bigg)\cdot\text{sign}\bigg(u^j + \frac{\lambda_d^j}{\beta_d}\bigg) + u_k \quad (26)$$

where $|\cdot|$ is componentwise absolute value and sign represents the signum function. To address the issue that the divisor is zero and avoid the instability of numerical computation, $0\cdot(0/0) = 0$ is assumed.

Algorithm 1 DCA for Solving (6)

Set $\mu, \beta_w, \beta_z, \beta_d, \epsilon, s, \gamma_2, \gamma_1, \eta > 0$, u_0, $p_0 = \nabla\Theta_s(Du_0)$, $q_0 = \nabla\Theta_s(Ku_0 - f)$, $k_{\max}$, $j_{\max}$.
For $k = 0, 1, 2, \ldots, k_{\max}$
  Given $\lambda_w^0, \lambda_w^0, \lambda_w^0$, and u_k, p_k, q_k.
    For $j = 0, 1, 2, \ldots, j_{\max}$
      Calculate w^{j+1} according to (24).
      Calculate z^{j+1} according to (25).
      Calculate d^{j+1} according to (26).
      Calculate u^{j+1} according to (27).
      Update $\lambda_w^{j+1}, \lambda_z^{j+1}, \lambda_d^{j+1}$ via (23).
      If $\|u^{j+1} - u^j\|_2/(1 + \|u^j\|_2) < \epsilon$,
        break,
      End
    End
  Set $u_{k+1} = u^{j+1}$ and compute $p_{k+1} = \nabla\Theta_s(Du_{k+1})$, $q_{k+1} = \nabla\Theta_s(Ku_{k+1} - f)$.
End

The minimization u-subproblem (22) is a least square. The corresponding normal equation is

$$\begin{aligned}&\left(D^T D + \frac{\beta_z}{\beta_w}K^T K + \frac{\beta_d}{\beta_w}I\right) u^{j+1}\\&= D^T\left(w^{j+1} - \frac{\lambda_w^j}{\beta_w}\right) + \frac{\beta_z}{\beta_w}K^T\left(z^{j+1} - \frac{\lambda_z^j}{\beta_z}\right)\\&\quad + \frac{\beta_z}{\beta_w}K^T f + \frac{\beta_d}{\beta_w}\left(d^{j+1} - \frac{\lambda_d^j}{\beta_d}\right).\end{aligned}\tag{27}$$

When $\beta_w, \beta_z, \beta_d > 0$, the coefficient matrix $D^T D + (\beta_z/\beta_w)K^T K + (\beta_x/\beta_w)I$ is nonsingular. Under the periodic boundary conditions, this matrix can be diagonalized by Fourier transform. So, the solution of normal equation (27) can be given by two fast Fourier transforms. If the matrix has no special structure to use, it can be commonly solved by the conjugate gradient method.

Finally, we show the novel DCA for solving the proposed model (6) in Algorithm 1.

C. Computational Complexity

The computational complexity of the proposed algorithm can be analyzed as follows. In the outer loop, the complexity of calculating Euclidean gradient is $O(n)$, where n is the image size. In the inner loop, the w, z and d subproblems are solved using the shrinkage operators in linear time, so the complexity of each of them is $O(n)$; the computational cost associated with the u subproblem is $O(n \log n)$ if its solution is obtained via fast Fourier transform and inverse transform, and is $O(n^2)$ if its solution is achieved via the conjugate gradient method; the update of the Lagrangian multipliers via (23) can be implemented straightforwardly in $O(n)$ time. Thus, the worst case complexity of the proposed algorithm for solving problem (6) is $O(k_{\max} j_{\max} n \log n)$ or $O(k_{\max} j_{\max} n^2)$. It should be pointed out that the complexity of our new algorithm is the same as those of the methods of [5], [6], and [16], but is lower than that of the classical ADMM presented in [8], [14], and [23] [i.e., $O(Tn \log n)$ or $O(Tn^2)$] if $k_{\max} j_{\max} < T$, where T is the total iteration number of the classical ADMM.

IV. Convergence

Using the tools from variational analysis, we prove that Algorithm 1 is convergence in this section. First, a simple property of subdifferential is established.

Lemma 1: If $\xi_{k+1} \in \partial\|u_{k+1} - u_k\|_1$, then

$$\|u_{k+1} - u_k\|_1 \le \langle \xi_{k+1}, u_{k+1} - u_k \rangle.$$

Proof: Let $J(u) = \|u - u_k\|_1$. By the definition of subdifferential, we obtain

$$\begin{aligned}J(u_k) - J(u_{k+1}) &= 0 - \|u_{k+1} - u_k\|_1\\&\ge \langle \partial J(u_{k+1}), u_k - u_{k+1} \rangle\\&= \langle \xi_{k+1}, u_k - u_{k+1} \rangle.\end{aligned}$$

■

Lemma 2: For the sequence generated by Algorithm 1, we have

$$F(u_k) - F(u_{k+1}) \ge \eta\|u_{k+1} - u_k\|_1 \tag{28}$$

and $\{u_k\}$ is bounded and convergent.

Proof: Because $g(u)$ is a convex function, we know

$$g(u_k) - g(u_{k+1}) \ge \langle \partial g(u_{k+1}), u_k - u_{k+1} \rangle. \tag{29}$$

Similarly, we obtain

$$h(u_{k+1}) - h(u_k) \ge \langle \partial h(u_k), u_{k+1} - u_k \rangle. \tag{30}$$

It follows from (17) that:

$$\partial h(u_k) - \eta\partial\|u_{k+1} - u_k\|_1 \in \partial g(u_{k+1}). \tag{31}$$

Combining (29) and (30), we obtain

$$\begin{aligned}F(u_k) - F(u_{k+1}) &\ge \langle -\partial g(u_{k+1}) + \partial h(u_k), u_{k+1} - u_k \rangle\\&= \langle \eta\xi_{k+1}, u_{k+1} - u_k \rangle\\&\ge \eta\|u_{k+1} - u_k\|_1\end{aligned}$$

where (31) and Lemma 1 are used.

For $k = 0, 1, \ldots$, summing all the inequalities in (28), we deduce that

$$\sum_{k=0}^{\infty} \|u_{k+1} - u_k\|_1 \le \frac{1}{\eta}F(u_0)$$

which implies that $\{u_k\}$ is a Cauchy sequence and is convergent. The assertion is proved. ■

Before we show the convergence result, the definition of η stationary point [24] is given as below.

Definition 1: For $\eta > 0$, u^* is η stationary point of $F(u)$ if there exists $\zeta \in \partial F(u)$ such that $\|\zeta\|_\infty \le \eta$.

Theorem 1: The limit point u^* of $\{u_k\}$ that is generated by Algorithm 1 is an η stationary point of the objective function $F(u)$.

Proof: From the relationship (31), we have

$$\begin{aligned}\partial F(u_{k+1}) &\ni \partial g(u_{k+1}) - \partial h(u_{k+1})\\&= \nabla h(u_k) - \nabla h(u_{k+1}) - \eta\xi_{k+1}.\end{aligned}\tag{32}$$

Due to the boundedness of $\xi_{k+1} \in [-1, 1]$ and $\{u_k\}$, we can find a common subsequence $\{k_j\}$, such that $\xi_{k_j} \to \xi^*$ and $u_{k_j} \to u^*$. Replacing k by k_j in (32), taking the limit and using the continuity of the function $\nabla h(u)$, we obtain $-\eta\xi^* \in \partial F(u^*)$. By $\|\eta\xi^*\|_\infty \leq \eta$, u^* is an η stationary point. ■

V. Experimental Validation

We evaluate the proposed model and algorithm on image restoration problem with impulse noise in this section. The experiments include image deblurring, image denoising and image recovery. All code are performed on a PC with 2.9-GHz processor and 16-GB RAM.

A. Image Debluring

In the first example, we show the effectiveness of the proposed method in suppressing impulse noise and restoring the blurred image. Impulse noise can be classified into two types, namely SP noise and RV noise. If a grayscale image is degraded by SP noise, then some of its pixels will change to the minimum (0) or maximum (255) pixel values. More specifically, the noisy f corrupted by the SP noise with noise level r $(0 \leq r \leq 1)$ is given as

$$f_i = \begin{cases} 0, & \text{with probability } \frac{r}{2} \\ 255, & \text{with probability } \frac{r}{2} \\ u_i, & \text{with probability } 1 - r \end{cases}$$

where u denotes the original grayscale image, and i denotes the pixel location. Similarly, if a grayscale image is degraded by RV noise, then some of its pixels will change to the random values which are between the minimum and the maximum pixel values. The noisy f corrupted by the RV noise with noise level r is expressed by

$$f_i = \begin{cases} d_i, & \text{with probability } r \\ u_i, & \text{with probability } 1 - r \end{cases}$$

where d_i is uniformly distributed in [0, 255]. In general, the high-level impulse noise means that more than 50% pixels of an entire image are corrupted. In this section, the experiment results of image deblurring with SP and RV noise are presented.

1) Compared With TVL1 and TVSCAD: In this part, the new model (6) is compared with the TVL1 model (1) and TVSCAD model (3). For high-level impulse noise, TVL1 model works poorly. In order to overcome the limitation of TVL1, TVSCAD is proposed. Moreover, our proposed model (6) is an improved version of TVSCAD, where the nonconvex log function is instead of the L_1 norm in the regularization term of TVSCAD. Moreover, TVSCAD model is solved by the classical DCA with L_2 proximal term, while the new model is solved by a novel DCA with L_1 proximal term. Hence, our approach is called NNDCL1, which means that nonconvex plus nonconvex model is solved by DC programming with L_1 proximal term.

In our experiments, a 9×9 Gaussian blur with standard deviation 10 and an average blur with size 9×9 are considered. Also, SP or RV noises will be added after blur the image. As we know, RV noise is more difficult to removal than SP noise

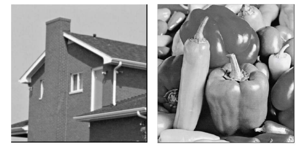

Fig. 1. Tested images.

since the noise value can be arbitrary numbers between the min and max pixel value. Therefore, we test 50%, 70%, and 90% noise levels for SP noise and 40%, 60%, and 80% noise levels for RV noise. The testing images are House (256×256) and Peppers (512×512), as shown in Fig. 1. The quality of the restoration image is evaluated by the signal to noise ration (SNR) and the structure similarity (SSIM) [25]. Obviously, the higher values of SNR and SSIM, the better of the restoration image.

The parameters of the tested models should be specified. These parameters have a great influence on the numerical performances. The best choice of the parameter μ is known to be problem dependent and very hard to find. So, we adjust the parameters one by one for each image. For the regularization parameter μ, we first determine the optimal parameter value μ of TVSCAD by solving a series of problems (3) as done in [5]. Then, the best μ values of TVSCAD are also used in our model (6). For a fair comparison, the time consumed in parameter tuning is not taken into account for all the tested methods. For the TVL1 model, μ takes 1.75 for both the two types of testing noise. For the TVSCAD model and our model, μ takes different values for the two types of noise. Specifically, under the SP noise, μ takes 22 for House image; and under the SP noise with noise levels of 50%, 70%, and 90%, it takes 21, 7, and 17, respectively, for Peppers image. While, under the RV noise with noise level of 80%, μ takes 5 for House and 3.75 for Peppers; under the RV noise with noise levels of 60% and 40%, it takes 5 and 21, respectively, for two test images. In order to make a fair evaluation, the function Φ_γ in (3) and (6) will be taken the same for high-level noise. As suggested in [5], we set $\gamma_1 = 0.08/k$ and $\gamma_2 = \max(0.2 \times 0.85^{k-1}, 0.1)$ for SP noise and $\gamma_1 = 0.0001$ and $\gamma_2 = 0.5$ for RV noise, About s in Ψ_s, it takes 0.035 for SP noise and 0.095 for RV noise.

Next, we consider these parameters in compared and proposed algorithms. For all the compared algorithms, we manually choose the regularization and the algorithmic parameters that yield the best SNR and SSIM in the experiments. For nonconvex problems, it is impossible to make an optimization algorithm achieve its best performance for all tests using a set of fixed parameters. Hence in general, the parameters of the optimization algorithm are tuned case by case to obtain much better results for each test. The proximal term can make the problem well defined and stabilize the method. It mentions that the proximal parameter η in (17) is related to the optimality condition. Hence, we set $\eta = 0.001$ in (16) and (17) throughout. It also noted that both of TVSCAD and NNDCL1 use ADMM to solve the subproblems (16) and (17). Then, we set $\beta_w = 5$, $\beta_z = 10$, $\beta_x = 100$ for SP noise, $\beta_w = 2$, $\beta_z = 5$, $\beta_x = 650$ for RV noise, and $\alpha = 1.618$ in all tests. For the parameters of TVL1 method, we set them as done in [23].

TABLE I
COMPARISON RESULTS OF DIFFERENT ALGORITHMS FOR SP AND RV NOISE

Images	Level	Blur	TVL1			TVSCAD			NNDCL1		
			Time	SNR	SSIM	Time	SNR	SSIM	Time	SNR	SSIM
House	50%(SP)	Gaussian	4.2656	11.6934	0.7674	22.8281	15.5615	0.8261	21.4688	15.9974	0.8358
		Average	4.1719	11.8543	0.7691	22.0625	15.6402	0.8268	20.9844	16.0296	0.8357
	70%(SP)	Gaussian	6.0938	9.5297	0.7227	27.6875	14.9751	0.8209	27.8906	14.9951	0.8211
		Average	5.9603	9.7914	0.7273	29.3281	14.9913	0.8209	28.3438	15.0035	0.8211
	90%(SP)	Gaussian	5.3594	5.3728	0.6326	29.8906	12.1104	0.7717	29.8125	12.1351	0.7723
		Average	6.2969	4.7739	0.6238	28.4219	12.2683	0.7733	29.0313	12.3014	0.7737
	40%(RV)	Gaussian	4.2344	11.8545	0.7726	21.0469	15.3423	0.8258	22.1406	15.9341	0.8352
		Average	4.2031	12.0024	0.7728	22.0156	15.4914	0.8282	21.5000	16.0723	0.8372
	60%(RV)	Gaussian	2.3438	9.4281	0.7237	59.0313	12.9070	0.7838	58.9219	12.9274	0.7842
		Average	2.6094	9.3768	0.7247	57.9375	12.7952	0.7822	57.8750	12.8240	0.7826
	80%(RV)	Gaussian	3.9063	3.3515	0.6195	58.9063	9.7897	0.7265	58.3438	9.8008	0.7269
		Average	3.8750	3.8104	0.6286	57.5313	10.1306	0.7293	58.0156	10.1368	0.7296
Peppers	50%(SP)	Gaussian	20.6094	14.5231	0.7972	103.0625	16.1556	0.8119	102.5938	16.4884	0.8218
		Average	20.5938	14.4890	0.7962	100.3594	16.1971	0.8112	102.2813	16.6093	0.8224
	70%(SP)	Gaussian	25.7031	12.9481	0.7628	133.3281	15.2821	0.8070	134.8125	15.3140	0.8074
		Average	26.0313	12.9786	0.7613	132.6094	15.3822	0.8075	134.4531	15.3898	0.8078
	90%(SP)	Gaussian	24.3281	7.0226	0.6252	133.4688	9.2721	0.7228	134.6250	9.7295	0.7291
		Average	25.4063	6.9395	0.6193	131.3906	8.9744	0.7210	135.8750	9.2049	0.7248
	40%(RV)	Gaussian	20.3281	14.6067	0.7994	101.2656	16.5237	0.8157	102.7969	16.7265	0.8204
		Average	20.6406	14.4556	0.7978	103.6250	16.5486	0.8153	105.2969	16.7887	0.8208
	60%(RV)	Gaussian	16.1875	10.2582	0.7082	268.4375	14.7699	0.7954	272.6875	14.7928	0.7957
		Average	16.1406	10.3871	0.7122	270.9375	14.7950	0.7955	273.1719	14.8176	0.7957
	80%(RV)	Gaussian	20.1250	3.7172	0.5834	273.0469	10.3323	0.7088	272.7813	10.2960	0.7084
		Average	21.3125	3.7399	0.5826	266.9219	10.2885	0.7120	279.4063	10.2507	0.7118

Fig. 2. Visual results for images corrupted by Gaussian blur with 70% SP noise and restored by TVL1, TVSCAD, and NNDCL1.

Fig. 4. Visual results for images corrupted by average blur with 60% RV noise and restored by TVL1, TVSCAD, and NNDCL1.

Fig. 3. Visual results for images corrupted by average blur with 90% SP noise and restored by TVL1, TVSCAD, and NNDCL1.

Fig. 5. Visual results for images corrupted by Gaussian blur with 80% RV noise and restored by TVL1, TVSCAD, and NNDCL1.

For initialization of all algorithms, we set $u_0 = f$. To compute u_{k+1} from u_k in (16) and (17), ADMM starts at u_k and the initial multipliers are zeros when launching ADMM as discussed in [5]. ADMM will terminate when $\epsilon \leq 0.0001$ or the inner iteration number is met. The maximum outer iteration of TVSCAD and NNDCL1 is 5 for SP noise and 10 for RV noise for high-level noise. But, the maximum outer iteration is set as 5 for 40% and 50% low density noise. The reason is that the quality has no much more improvement after several iterations.

Now, we show the compared results of images corrupted by Gaussian and average blur with SP and RV noise. The partial visual comparisons are shown in Figs. 2–5, which include the blurry noisy images and the restored images by different methods. It can be seen that both TVSCAD and NNDCL1 outperform TVL1, especially for high-level noise. Comparing the results of TVSCAD and NNDCL1, we see that NNDCL1 performs competitive with TVSCAD and the two methods adequately restore the images from the very high-level noise. The quantitative evaluations of experiment results are reported in Table I, which gives CPU time (seconds), SNR (dB) and SSIM. Note that each algorithm is tested ten times on the same image under the same random noise level, then each result reported in Table I is obtained by averaging the ten test results. From Table I, one can see that the new model obtains higher SNR and SSIM values in most cases and CPU

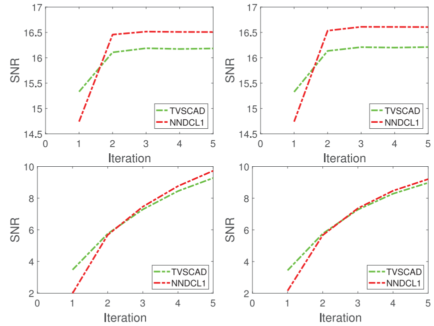

Fig. 6. SNR values versus iteration numbers for Peppers with 50% (first row) and 90% (second row) SP noise.

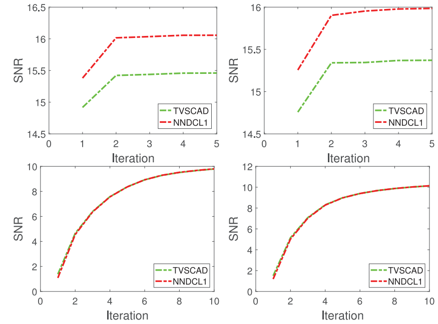

Fig. 7. SNR values versus iteration numbers for House with 40% (first row) and 80% (second row) RV noise.

times of NNDCL1 and TVSCAD are almost the same. The convergence curves of NNDCL1 and TVSCAD are plotted in Figs. 6 and 7. We can see that the curves of SNR values increase as the outer iteration numbers increase. NNDCL1 performs better than TVSCAD in the case of SP noise. For RV noise, NNDCL1 is also competitive.

As shown in Table I, Figs. 6 and 7, the superiority of the proposed algorithm is more evident for low-level noise than for high-level noise. The SNR and SSIM values of NNDCL1 are about 0.4dB and 0.01 higher than those of TVSCAD, respectively. However, with the increase of noise density, the superiority decreases gradually. This may be explained as follows. For low-level noise, both the regularization term and the data fidelity term play equally important roles. In this case, nonconvex regularizer has advantages over convex one. For high-level noise, such as 80% and 90% impulse noise, the data fidelity term plays a far more important role than the regularization term. Therefore, compared with TVSCAD, the superiority of NNDCL1 is no longer obvious. But, in general, by using the nonconvex log penalty to promote sparsity, NNDCL1 can obtain higher quality restored images than TVSCAD.

2) Compared With Nonconvex Models: In this part, the new approach is compared with some nonconvex methods [6], [8], [16]. In [6], ET function is used as data fitting term and TV is taken as regularization term. The model (4) is called ETTV and solved by proximal linearized minimization method. In [8], an L0TV nonconvex model (2) was presented and the proximal ADMM was used to solve the optimization problem. In [16], a nonconvex data fitting term plus nonconvex regularizer model (5) was introduced and the authors also presented an effective algorithm, called NNADM.

In this experiment, the same blur types are adopted as discussed in the last part. But, we test 30% and 80% noise levels of SP noise and 20% and 70% noise levels of RV noise. The tested images are Parrot (256 × 256), Lena (512 × 512), and Man (512 × 512), which are displayed in Fig. 8. The μ value of (6) is taken from $\{3.8, 15, 20, 75\}$, and s is 0.035 for SP noise and 0.09 for RV noise. We choose penalty parameters

$$\beta_w, \beta_z, \beta_d \in \{2, 5, 10, 20, 50, 100, 650, 1000\}$$

in the ADMM. The max outer iteration number is 5 or 10. For ETTV, L0TV, and NNADM, we use the suggested parameters values.

Fig. 8. Tested images.

Fig. 9. Average blur image with 30% SP noise and images restored by ETTV, L0TV, NNADM, and NNDCL1.

Numerical performances of various compared methods are recorded in Table II. The results show that NNDCL1 performs better than ETTV, L0TV, and NNADM in terms of SNR and SSIM in most cases. L0TV and NNADM seem to need more CPU time in seconds than ETTV and NNDCL1 except the case of 70% noise level. Moreover, the performances of ETTV and NNDCL1 are almost similar in terms of CPU times for all noise levels.

The visual comparisons are shown in Figs. 9–12. These figures give restored results of images degraded by average blur kernel with 30% SP noise, Gaussian blur kernel with 80% SP noise, Gaussian blur kernel with 20% RV noise and average blur kernel with 70% RV noise. We see that the four methods can remove the noise well, regardless of low or high-level noise.

The curves of partial SNR and SSIM versus CPU time are plotted in Figs. 13–16. Among the four algorithms, SNR and SSIM values of NNDCL1 are increasing as CPU time and always on the top at the last time. ETTV is in the middle of the four methods in most cases. From Figs. 13 and 14, we can find that the curves of NNADM are far below. The increasing rates of L0TV are slightly slower in Figs. 15 and 16. So, from these

TABLE II
COMPARISON RESULTS OF DIFFERENT ALGORITHMS FOR SP AND RV NOISE

Images	Level	Blur	ETTV			L0TV			NNADM			NNDCL1		
			Time	SNR	SSIM	Time	SNR	SSIM	Time	SNR	SSIM	Time	SNR	SSIM
Parrot	30%(SP)	Gaussian	35.8281	14.2114	0.8865	63.2656	14.6241	0.8768	47.0469	13.8484	0.8814	39.7813	14.9747	0.8829
		Average	35.1719	14.2139	0.8866	61.3438	14.7842	0.8791	47.3125	13.9724	0.8832	36.5958	15.1324	0.8846
	80%(SP)	Gaussian	35.1719	11.6026	0.8170	61.8594	12.0709	0.8140	84.6875	9.6343	0.8006	37.7188	12.0594	0.8271
		Average	34.4844	11.6045	0.8180	61.6563	12.0472	0.8121	82.5781	10.6810	0.8072	38.5625	12.0754	0.8284
	20%(RV)	Gaussian	35.1875	19.4783	0.9397	63.4531	18.8850	0.9350	57.1563	18.8047	0.9385	36.0156	19.5132	0.9425
		Average	34.8281	20.0122	0.9431	63.4531	19.4198	0.9399	57.4531	19.3350	0.9419	36.3906	20.0454	0.9460
	70%(RV)	Gaussian	68.7813	10.7450	0.7815	63.4531	11.2460	0.7657	60.8281	10.9647	0.7660	73.0156	11.0407	0.7860
		Average	67.8594	10.6379	0.7801	64.0625	11.2338	0.7674	58.0625	10.9584	0.7632	72.9219	11.0595	0.7868
Lena	30%(SP)	Gaussian	160.4844	17.5292	0.8782	317.7344	16.9736	0.8614	176.1719	16.8644	0.8652	164.7031	17.7704	0.8756
		Average	163.9219	17.6066	0.8796	320.0625	17.0312	0.8626	179.6563	16.9207	0.8666	165.0469	17.8682	0.8772
	80%(SP)	Gaussian	162.0625	14.9607	0.8210	318.7344	15.1333	0.8189	309.4219	12.6904	0.8071	172.3750	15.3160	0.8233
		Average	162.2500	14.9713	0.8209	319.2656	15.1701	0.8209	315.0625	12.7450	0.8075	171.3906	15.3803	0.8246
	20%(RV)	Gaussian	159.6406	20.8414	0.9048	318.5469	20.5992	0.9034	294.1719	20.2767	0.9012	168.6719	20.7412	0.9084
		Average	159.6563	21.0909	0.9081	319.4844	20.8467	0.9068	295.1719	20.5214	0.9045	170.7656	21.0055	0.9120
	70%(RV)	Gaussian	327.0313	13.5388	0.7831	322.1250	13.3847	0.7702	299.0156	13.4411	0.7789	348.3594	14.0141	0.7855
		Average	326.7969	13.7969	0.7822	320.6719	13.3790	0.7703	294.3906	13.4175	0.7792	341.3488	14.0131	0.7860
Man	30%(SP)	Gaussian	161.9063	15.5185	0.8361	316.8906	14.9178	0.8061	181.0625	15.0034	0.8201	165.4844	15.8707	0.8429
		Average	165.9688	15.6153	0.8390	323.6094	15.0147	0.8095	180.8125	15.0964	0.8234	166.0156	15.9741	0.8457
	80%(SP)	Gaussian	165.0313	13.0031	0.7407	317.2344	13.1090	0.7357	316.5938	12.0420	0.7451	171.9844	13.4770	0.7550
		Average	160.7031	13.0289	0.7416	320.3125	13.1166	0.7366	314.8594	11.9620	0.7432	167.7031	13.4973	0.7561
	20%(RV)	Gaussian	161.0313	19.4947	0.9244	323.1406	19.2115	0.9216	293.1719	18.9691	0.9191	170.1719	19.4400	0.9268
		Average	159.9063	19.7582	0.9277	319.0313	19.4696	0.9253	295.8906	19.1782	0.9221	171.0000	19.6650	0.9295
	70%(RV)	Gaussian	321.6719	11.6699	0.6820	308.7813	11.5663	0.6667	289.9063	12.0401	0.6931	333.1563	12.2584	0.7016
		Average	321.9688	11.5880	0.6810	317.7969	11.4971	0.6649	294.7031	12.0217	0.6922	339.2031	12.1662	0.6988

Fig. 10. Gaussian blur image with 80% SP noise and images restored by ETTV, L0TV, NNADM, and NNDCL1.

Fig. 11. Gaussian blur image with 20% RV noise and images restored by ETTV, L0TV, NNADM, and NNDCL1.

Fig. 12. Average blur image with 70% RV noise and images restored by ETTV, L0TV, NNADM, and NNDCL1.

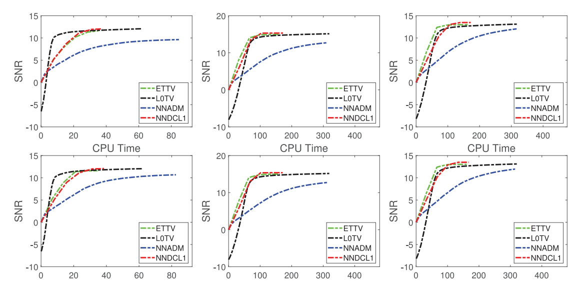

Fig. 13. SNR values versus CPU times for restoring Parrot, Lena, and Man images which are corrupted by Gaussian blur (first row) and average blur (second row) with 80% SP noise.

figures, we observe that NNDCL1 has a good convergence property and is stable.

B. Image Denoising

To further verify the performance of the proposed algorithm in reducing artifacts, image denoising (when K is the identity operator) experiment is tested. In this experiment, the proposed algorithm is compared with two very recent works [26], [27] for impulse noise removal. In [26], under the assumption that both signal and noise are sparse, the authors proposed a fast iterative method, called iterative double thresholding (IDT), to remove impulse noise, which has a low complexity. In [27], a nonconvex model called L0OGSTV was presented, which has an L_0 norm data fidelity term and overlapping group sparse TV (OGSTV) regularization function. This model can be considered as an extension of L0TV. Majorization-minimization method and ADMM were used to solve the L_0-norm and the OGSTV optimization problem.

The test images are Shape image (256×256) and MRI image (320 × 320), as shown in Fig. 17. The SP and RV noises with different noise densities are tested. The noise levels of SP are 40% and 60%, and those of RV are 10% and 50%. For fairness, we manually chose the regularization and algorithm

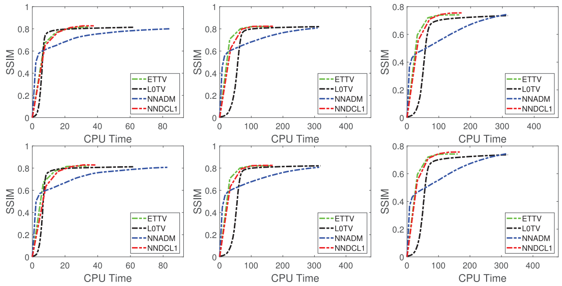

Fig. 14. SSIM values versus CPU times for restoring Parrot, Lena, and Man images which are corrupted by Gaussian blur (first row) and average blur (second row) with 80% SP noise.

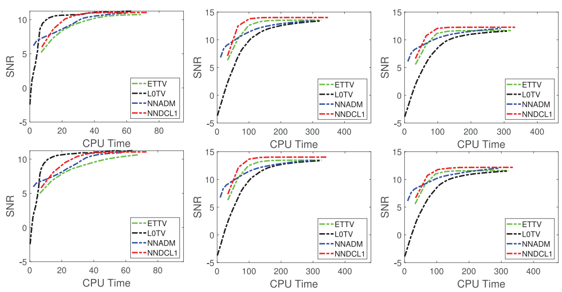

Fig. 15. SNR values versus CPU times for restoring Parrot, Lena, and Man images which are corrupted by Gaussian blur (first row) and average blur (second row) with 70% RV noise.

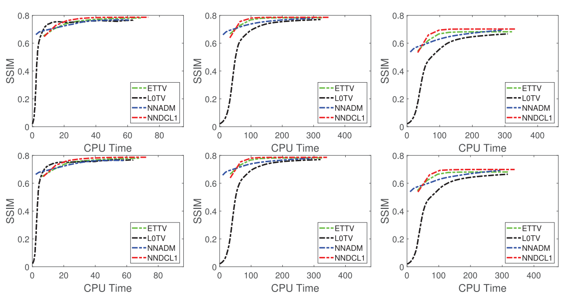

Fig. 16. SSIM values versus CPU times for restoring Parrot, Lena, and Man images which are corrupted by Gaussian blur (first row) and average blur (second row) with 70% RV noise.

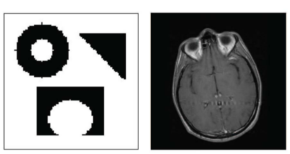

Fig. 17. Tested images.

parameters which yields the best SNR and SSIM for IDT and NNDCL1, while for L0OGSTV, the parameter settings closely follow [27]. The maximum outer iteration of NNDCL1 takes 2 for low-level noise, and takes 3 or 5 for high-level noise.

The experiment results are reported in Table III. From the table, two observations can be made. First, the proposed NNDCL1 method almost always outperforms IDT and L0OSGTV methods for both SP and RV impulse noise removal. This is obvious especially in terms of SNR and SSIM. Second, IDT has the lowest runtime. The reason is that the main computational costs are two thresholding operators, and not involving the matrix–vector product.

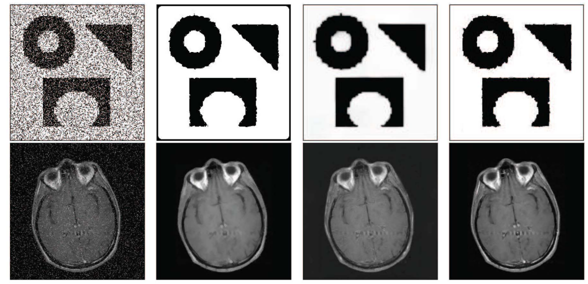

Fig. 18. Denoising results by IDT, L0OGSTV, and NNDCL1 for Shape corrupted by 40% SP noise and MRI by 10% RV noise.

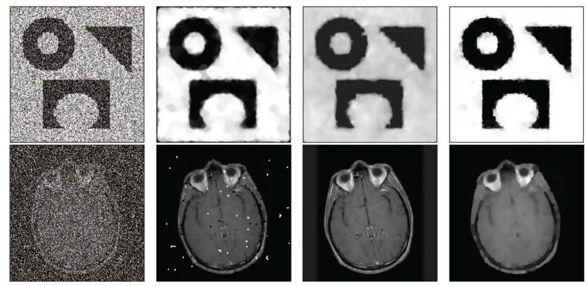

Fig. 19. Denoising results by IDT, L0OGSTV, and NNDCL1 for Shape corrupted by 50% RV noise and MRI by 60% SP noise.

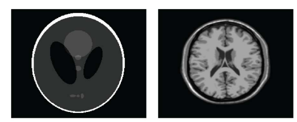

Fig. 20. Original images.

We show the restored images in Figs. 18 and 19. From Fig. 18, we can see that all the test methods can remove the noise well when its density is low. Whereas, as shown in Fig. 19, IDT is not capable of completely removing the noise when the noise density is high or not sparse. Besides, although L0OGSTV preserves the details of the image, it tends to cause blurred edges or produce undesired artifacts, and its execution process usually takes a long time. By comparison, the proposed NNDCL1 method can achieve a good balance between artifact reduction and noise removal.

Besides, it is seen from the comparative experiment that the first iteration of Algorithm 1 produces a high-quality guess for the nonconvex optimization problem, and can be interpreted as a warm-start step. This means that the proposed DCA needs a less number of iterations to achieve the optimal solution, thus improving the overall efficiency of the proposed method.

C. Image Recovery

The third example is given to further show the superiority of the proposed method in image recovery with impulse noise and the effectiveness of the L_1 proximal technique. In this experiment, we apply our model to the image recovery from compressed measurements. The tested images are Shepp-Logan and Brain, and shown in Fig. 20. The size

TABLE III
COMPARISON RESULTS OF DIFFERENT ALGORITHMS FOR SP AND RV NOISE REMOVAL

Images	Level	IDT			L0OGSTV			NNDCL1		
		Time	SNR	SSIM	Time	SNR	SSIM	Time	SNR	SSIM
MRI	40%(SP)	13.0313	14.3147	0.9041	99.2031	22.5821	0.8513	37.2500	14.8918	0.9157
	60%(SP)	20.5469	8.0463	0.8523	88.7031	10.2914	0.7382	62.0469	10.4628	0.8394
	10%(RV)	21.6503	14.8633	0.9144	52.4219	13.2394	0.7839	16.6563	20.9915	0.9743
	50%(RV)	6.5626	7.7387	0.5919	105.9219	3.6139	0.6182	56.9375	10.9142	0.8156
Shape	40%(SP)	9.2344	6.7720	0.8965	52.1719	6.4567	0.8692	18.5938	15.4258	0.9483
	60%(SP)	14.1719	7.5927	0.8666	51.3438	3.8005	0.8424	33.5469	12.5892	0.9115
	10%(RV)	14.6250	10.6736	0.9753	86.3488	7.6116	0.7964	10.3750	23.5858	0.9874
	50%(RV)	6.8906	8.7022	0.6602	55.4688	3.7324	0.7345	35.2656	13.6101	0.8268

TABLE IV
PSNR VALUES OF THE IMAGES RESTORED BY THE COMPARED ALGORITHMS

Noise	Images	L1LS-FISTA	$L_{0.5}$LS-ADMM	YALL1	$L_{0.2}$LA-ADMM	$L_{0.5}$LA-ADMM	$L_{0.7}$LA-ADMM	NNDCL1
Gaussian mixture	Logan	32.3301	32.5683	27.9568	39.3128	40.3454	39.2712	42.7397
	Brain	27.7681	26.5287	26.6500	29.0384	29.2763	28.7730	29.1590
SτS	Logan	12.5396	13.0372	29.5403	31.6272	32.7204	32.6292	33.5979
	Brain	12.1137	11.3154	24.8355	27.1307	27.5435	27.1410	27.2365

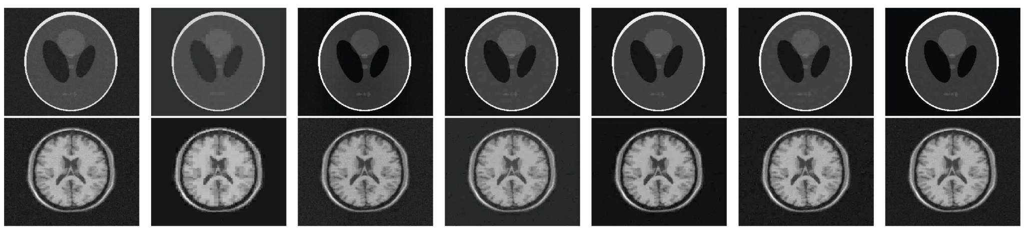

Fig. 21. Recovery results of L1LS-FISTA, $L_{0.5}$LS-ADMM, YALL1, $L_{0.2}$LA-ADMM, $L_{0.5}$LA-ADMM, $L_{0.7}$LA-ADMM, and NNDCL1 on Shepp-Logan and Brain images with Gaussian mixture noise.

of each image is 256×256. The measurement number is $m = \text{round}(0.4n)$, where $n = 65536$. A partial discrete cosine transformation matrix A is employed as the sensing matrix K and the Haar wavelets are used as the basis functions. Then, the model (6) becomes

$$\min_x \ \mu\Phi_\gamma(Ax - b) + \Psi_s(x)$$

where x is a wavelet coefficient and b is the sampled wavelet data with impulse noise. We compare the new model with the L_q regularized least absolute (L_qLA) formulation

$$\min_x \ \mu\|Ax - b\|_1 + \|x\|_q$$

and the L_q regularized least squares (L_qLS) formulation

$$\min_x \ \mu\|Ax - b\|_2 + \|x\|_q$$

where $0 < q \leq 1$.

Two types of impulse noise [28] are tested. The first one is Gaussian mixture noise. The probability density function of two-term Gaussian model is given by

$$(1-\varepsilon)\mathcal{N}\left(0, \sigma^2\right) + \varepsilon\mathcal{N}\left(0, \kappa\sigma^2\right)$$

where $\varepsilon = 0.1$ and $\kappa = 1000$ in our experiments. The first term stands Gaussian thermal noise and the second term means the behavior of impulse noise. The second one is symmetric τ-stable (SτS) noise. The characteristic function of SτS distribution can be expressed as $e^{(j\tau\omega)-\rho^\tau|\omega|^\tau}$, where the characteristic exponent τ is 1 and the scale parameter ρ is 10^{-4}. The smaller the value of τ, the more impulse the noise is.

Our new method NNDCL1 is compared with some representative algorithms, which include L1LS-FISTA [29], L_qLS-ADMM [30], YALL1 [31], and L_qLA-ADMM [28]. L1LS-FISTA solves the L_1LS model. $L_{0.5}$LS-ADMM solves the L_qLS model based on ADMM with $q = 0.5$. YALL1 solves the robust L_1LA model. L_qLA-ADMM with different values of $q \in \{0.2, 0.5, 0.7\}$ solves the L_qLA model. The peak-signal noise ratio (PSNR) is used to evaluate the recovery performance.

Fig. 21 presents the recovery versions of the images corrupted by Gaussian mixture noise. We can see that all algorithms achieve good results in terms of visual quality. Fig. 22 gives the recovery images that have been damaged by SτS noise. One can see that the L_2 data fidelity-based methods, that is, L1LS-FISTA and $L_{0.5}$LS-ADMM, are failed, while the L_1 data fidelity-based algorithms, that is, YALL1, L_qLA-ADMM and NNDCL1, work well. That is because the considered SτS noise contains more impulse noise than Gaussian noise. The performances in terms of PSNR are shown in Table IV. The results show that NNDCL1 outperforms those representative algorithms on reconstructing Shepp-Logan. The improvements are higher as 2.39 dB (Gaussian mixture noise) and 0.87 dB (SτS noise) over the compared algorithms. The reason may be that the L_1 norm proximal term of the proposed algorithm further promotes the sparsity in the iterations. This advantage

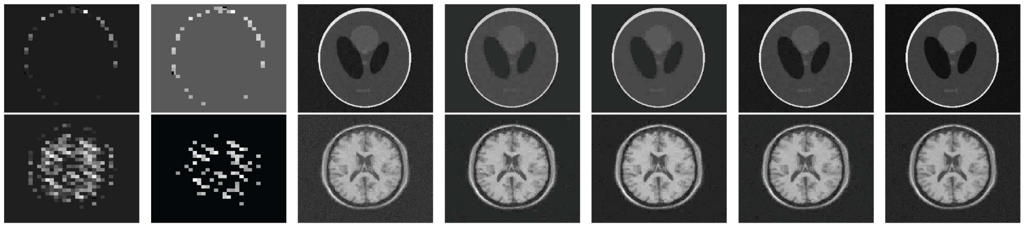

Fig. 22. Recovery results of L1LS-FISTA, $L_{0.5}$LS-ADMM, YALL1, $L_{0.2}$LA-ADMM, $L_{0.5}$LA-ADMM, $L_{0.7}$LA-ADMM, and NNDCL1 on Shepp-Logan and Brain images with SτS noise.

TABLE V
AVERAGE RESULTS ON REMOVING MIXED GAUSSIAN PLUS RV AND SP IMPULSE NOISE

DataSet	Level	BdCNN		NNDCL1	
		PSNR	SSIM	PSNR	SSIM
Pascal	(15,50,30)	25.5198	0.7360	23.8147	0.7089
	(25,40,20)	26.5697	0.7440	24.2687	0.7060
	(15,80,0)	16.3726	0.3790	18.0848	0.5756
	(25,70,40)	19.0974	0.5305	20.4848	0.6079

is decreased on recovering Brain image. The PSNR values attained by NNDCL1 are slightly lower than $L_{0.5}$LA-ADMM, but are higher than L1LS-FISTA, $L_{0.5}$LS-ADMM, YALL1, $L_{0.2}$LA-ADMM, and $L_{0.7}$LA-ADMM.

D. Mixed Noise Removal

Our last example is given to show the application of the new model in removing mixed noise. In this section, Gaussian plus impulse noise and Gaussian mixture noise are considered. In recent years, the deep learning using convolutional neural network (CNN) has attracted many researchers for its wide application in image processing. It has been gradually established as one of the significant techniques. Hence, we compared our new method with deep-learning-based methods in this section. PSNR and SSIM are adopted to evaluate the restoration quality.

1) Gaussian Plus Impulse Noise: First, we test the removal of Gaussian plus impulse noise. In this experiment, we choose randomly 500 images from the Pascal dataset as our test images. The images are corrupted by white Gaussian noise plus RV and SP noise. To demonstrate the performance for different settings of mixed Gaussian, RV, and SP impulse noise, we test on four noise levels, which are shown in Table V. The noise level of (15, 50, 30) in this table means that Gaussian noise ($\sigma = 15$) + 50% RV noise + 30% SP noise, and the other three cases have similar meanings. We compare our proposed method with BdCNN [32], which is a learning-based method. By training different kinds of noisy images with various noise levels, BdCNN can remove mixed noise regardless of the noise level. For BdCNN, we use the network trained by the authors of [32]. For our model, the regularization parameter μ is set as 1.35, and the other parameters are set to $s = 10$, $\gamma_1 = 0.01$, $\gamma_2 = 0.5$. We set $\beta_w = 10$, $\beta_z = 50$, $\beta_x = 100$, $\eta = 0.00001$, and $\alpha = 1.618$ for the ADMM. The maximum number of outer iterations is 5. This parameter setting of the proposed method is used for all test cases.

Table V reports the test results for Gaussian plus impulse noise removal. We can see from Table V that BdCNN achieves higher PSNR than NNDCL1 when the noise level is low. But, our method performs well in terms of PSNR and SSIM when the images are corrupted by heavy mixed noises. This is because our new method use the nonconvex data fitting term, which can detect impulse noise effectively. Hence, it can be seen that compared with the deep-learning-based method, our numerical results are quite competitive.

2) Gaussian Mixture Noise: In the second test, we compare the proposed method with EMCMM [33] on the removal of Gaussian mixture noise. EMCNN is a variational model for mixed noise removal, and is integrated with the CNN deep learning regularization. For EMCNN, image prior is learned by the CNN that is associated with a variational functional.

In this part, Pascal (500 test images, which are the same as those chosen in the previous test) and Berkeley Segmentation Dataset (BSD, 100 test images in total) datasets are used for the comparison. We consider the following Gaussian mixture noise form:

$$\varepsilon\mathcal{N}\left(0, \sigma_1^2\right) + (1-\varepsilon)\mathcal{N}\left(0, \sigma_2^2\right).$$

Then, the test images are corrupted by Gaussian mixture noise with mixture ratio $\varepsilon = 0.4, 0.5, 0.6$ and three noise levels, that is: 1) $\sigma_1 = 10$ and $\sigma_2 = 30$; 2) $\sigma_1 = 15$ and $\sigma_2 = 70$; and 3) $\sigma_1 = 25$ and $\sigma_2 = 60$. For EMCNN, the regularization trained by the authors of [33] is adopted, and the parameters are set as suggested in [33]. For the proposed method, the setting of the parameters is the same as that of the previous test.

Table VI reports the results for Gaussian mixed noise removal. It is seen from Table VI that the proposed method achieves the highest SSIM index except for the case of low noise level tested on BSD, which means that our method produces the best restored results in the most cases. We also observe that EMCNN performs better than NNDCL1 in terms of PSNR in the most cases. Fig. 23 shows the restored images of EMCNN and NNDCL1 for two test images (Bike image from Pascal and Penguin image from BSD) with different noise levels. We can see from Fig. 23 that EMCNN can retain more image details. Thus, the mean square error value of EMCNN is less than NNDCL1, which leads to that EMCNN has a higher PSNR value. However, a small amount of noise or artifacts still exist in the images restored by EMCNN. Although NNDCL1 generates oversmooth edges in the restored images, the noise is completely removed. By the

TABLE VI
AVERAGE PSNR AND SSIM VALUES: EMCNN AND NNDCL1 ON PASCAL AND BSD DATASETS

DataSet		σ_1=10 σ_2=30			σ_1=15 σ_2=70			σ_1=25 σ_2=60		
		0.4:0.6	0.5:0.5	0.6:0.4	0.4:0.6	0.5:0.5	0.6:0.4	0.4:0.6	0.5:0.5	0.6:0.4
Pascal	EMCNN	29.2946	30.6564	31.6237	27.5260	28.1595	29.0134	26.8140	27.5028	28.0263
		0.6834	0.7614	0.8071	0.6339	0.6784	0.7281	0.5958	0.6408	0.6729
	NNDCL1	27.9841	28.2771	28.5647	26.1412	26.6347	27.0935	25.5511	25.8712	26.1714
		0.7938	0.8039	0.8131	0.7416	0.7567	0.7699	0.7080	0.7180	0.7270
BSD	EMCNN	27.7475	28.5534	29.3255	25.1053	25.6313	26.3697	24.3528	24.9035	25.3232
		0.7187	0.7614	0.7968	0.5811	0.6194	0.6705	0.5337	0.5698	0.5957
	NNDCL1	25.3835	25.6220	25.8561	23.9501	24.3713	24.7822	23.4221	23.6832	23.9367
		0.6828	0.6961	0.7087	0.6121	0.6346	0.6553	0.5763	0.5897	0.6025

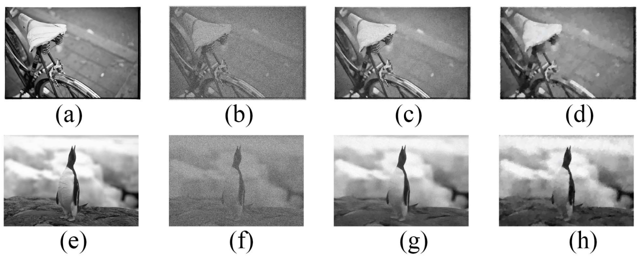

Fig. 23. (a) Bike image, (b) noisy image, $\sigma_1 = 25$, $\sigma_2 = 60$, and $\varepsilon = 0.4$, (c) restored by EMCNN, PSNR = 26.1105, and SSIM = 0.6348, (d) restored by NNDCL1, PSNR = 23.0624, and SSIM = 0.7423, (e) Penguin image, (f) noisy image, $\sigma_1 = 15$, $\sigma_2 = 70$, and $\varepsilon = 0.5$, (g) restored by EMCNN, PSNR = 30.0007,and SSIM = 0.7337, and (h) restored by NNDCL1, PSNR = 29.3246, and SSIM = 0.7981.

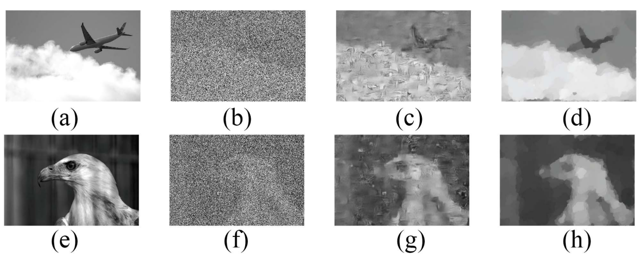

Fig. 24. (a) Plane image, (b) noisy image with the level (20, 80, 30), (c) restored by BdCNN, PSNR = 15.0069, and SSIM = 0.5602, (d) restored by NNDCL1, PSNR = 20.6525, and SSIM = 0.8545, (e) Eagle image, (f) noisy image with the level (35, 80, 20), (g) restored by BdCNN, PSNR = 16.9129, and SSIM = 0.4900, and (h) restored by NNDCL1, PSNR = 19.2452, and SSIM = 0.6694.

definition of SSIM, its value depends both on image mean value and variance. So, NNDCL1 can obtain a higher SSIM value under a lower PSNR value. This may be the reason that the proposed algorithm has a lower PSNR value but a higher SSIM value than EMCNN. Moreover, compared to the EMCNN, our new method shows an improvement in SSIM for high noise level. This shows the potential ability of our new method to improve the performance of high-level mixture noise removal.

In summary, as shown in Tables V and VI, it is obvious that our new method can effectively remove different levels of mixed noise. Though the tested deep-learning-based methods exhibit better performance in terms of PSNR, the proposed method has a higher SSIM value that BdCNN and EMCNN. Since deep-learning-based methods are data dependent, the quality of their restored images are likely unsatisfactory if the noise distribution or noise level is not included in the training samples. To illustrate this, Fig. 24 shows the images restored by BdCNN and NNDCL1 with the noise levels of (20, 80, 30) and (35, 80, 20). Note that the two noise levels are not included in the training dataset of BdCNN. It is seen that the images restored by BdCNN have unexpected artifacts, and the perceptual quality of the restored images are unsatisfactory. Although some details are lost in the images restored by the proposed NNDCL1, the proposed method has a good performance in denoising. We also note that both the PSNR and SSIM values of the proposed method are higher than that of BdCNN in this case. Moreover, compared with the deep-learning-based methods, our proposed method has the advantage that it does not require training data and training processing, and can deal with different image processing tasks. In addition, the use of the nonconvex and nonsmooth function in image processing can preserve piecewise constant regions and match the image data perfectly. This is particularly useful in enhancing restored images and removing impulse noise.

VI. CONCLUSION

In this article, we have proposed a new model for image restoration with impulse noise. Although the resulting optimization problem is nonconvex, a novel and effective DCA has been designed for solving it. We have proved that the limit point of the sequence generated by the proposed algorithm is a stationary point of the nonconvex objective function. The convergence analysis expands the existing results in DC programming. Experimental results of image deblurring, denoising, and recovery demonstrated that the proposed approach is highly competitive compared with the existing popular methods. Note that it remains largely unexplored how to adaptively choose the regularization parameter μ. This is an open and challenging problem, and we leave the problem to further research.

REFERENCES

[1] S. Alliney, "Digital filters as absolute norm regularizers," *IEEE Trans. Signal Process.*, vol. 40, no. 6, pp. 1548–1562, Jun. 1992.

[2] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D, Nonlinear Phenom.*, vol. 60, pp. 259–268, Nov. 1992.

[3] M. Nikolova, "A variational approach to remove outliers and impulse noise," *J. Math. Imag. Vison*, vol. 20, no. 2, pp. 99–120, 2004.

[4] M. Nikolova, M. K. Ng, and C.-P. Tam, "On l_1 data fitting and concave regularization for image recovery," *SIAM J. Sci. Comput.*, vol. 35, pp. 397–430, 2013.

[5] G. Gu, S. Jiang, and J. Yang, "A TVSCAD approach for image deblurring with impulsive noise," *Inverse Problems*, vol. 33, no. 12, 2017, Art. no. 125008.

[6] X. Zhang, M. Bai, and M. K. Ng, "Nonconvex-TV based image restoration with impulse noise removal," *SIAM J. Imag. Sci.*, vol. 10, no. 3, pp. 1627–1667, 2017.

[7] B. Zhang, G. Zhu, and Z. Zhu, "A TV-log nonconvex approach for image deblurring with impulsive noise," *Signal Process.*, vol. 174, Sep. 2020, Art. no. 107631.

[8] G. Yuan and B. Ghanem, "l_0TV: A sparse optimization method for impulse noise image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 352–364, Feb. 2019.

[9] O. Scherzer, *Handbook of Mathematical Methods in Imaging*. New York, NY, USA: Springer, 2010.

[10] P. Cascarano, L. Calatroni, and E. L. Piccolomini, "Efficient l_0 gradient-based super-resolution for simplified image segmentation," *IEEE Trans. Comput. Imaging*, vol. 7, pp. 399–408, Apr. 2021.

[11] T. Zhang, "Analysis of multi-stage convex relaxation for sparse regularization," *J. Mach. Learn. Res.*, vol. 11, no. 3, pp. 1081–1107, 2010.

[12] C. H. Zhang, "Nearly unbiased variable selection under minimax concave penalty," *Ann. Stat.*, vol. 38, no. 2, pp. 894–942, 2010.

[13] I. Selesnick and M. Farshchian, "Sparse signal approximation via nonseparable regularization," *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2561–2575, May 2017.

[14] J. Yang, Y. Zhang, and W. Yin, "An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise," *SIAM J. Sci. Comput.*, vol. 31, no. 4, pp. 2842–2865, 2009.

[15] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imag. Vison*, vol. 40, pp. 120–145, May 2011.

[16] Z. Cui and Q. Fan, "A 'nonconvex+nonconvex' approach for image restoration with impulse noise removal," *Appl. Math. Model.*, vol. 62, pp. 254–271, Oct. 2018.

[17] M. Bai, X. Zhang, and Q. Shao, "Adaptive correction procedure for TVL1 image deblurring under impulse noise," *Inverse Problems*, vol. 32, no. 8, 2016, Art. no. 85004.

[18] T. H. Le and T. P. Dinh, "DC programming and DCA: Thirty years of developments," *Math. Program.*, vol. 169, no. 1, pp. 5–68, 2018.

[19] T. Xie, S. Li, and J. Lai, "Adaptive rank and structured sparsity corrections for hyperspectral image restoration," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 8729–8740, Sep. 2022, doi: 10.1109/TCYB.2021.3051656.

[20] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[21] Y. Jia, H. Liu, J. Hou, S. Kwong, and Q. Zhang, "Semisupervised affinity matrix learning via dual-channel information recovery," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 7919–7930, Aug. 2022, doi: 10.1109/TCYB.2020.3041493.

[22] H. Zhang, X.-L. Zhao, T.-X. Jiang, M. K. Ng, and T.-Z. Huang, "Multiscale feature tensor train rank minimization for multidimensional image recovery," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 13395–13410, Dec. 2022, doi: 10.1109/TCYB.2021.3108847.

[23] R. H. Chan, M. Tao, and X. Yuan, "Constrained total variation deblurring models and fast algorithms based on alternating direction method of multipliers," *SIAM J. Imag. Sci.*, vol. 6, no. 1, pp. 680–697, 2013.

[24] Y. Gao and C. Wu, "On a general smoothly truncated regularization for variational piecewise constant image restoration: Construction and convergent algorithms," *Inverse Problems*, vol. 36, no. 4, 2020, Art. no. 45007.

[25] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, pp. 600–612, 2004.

[26] S. Sadrizadeh, N. Zarmehi, E. A. Kangarshahi, H. Abin, and F. Marvasti, "A fast iterative method for removing impulsive noise from sparse signals," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 1, pp. 38–48, Jan. 2021.

[27] M. Yin, T. Adam, R. Paramesran, and M. F. Hassan, "An l_0-overlapping group sparse total variation for impulse noise image restoration," *Signal Process Image Commun.*, vol. 102, Mar. 2022, Art. no. 116620.

[28] F. Wen, L. Pei, Y. Yang, W. Yu, and P. Liu, "Efficient and robust recovery of sparse signal and image using generalized nonconvex regularization," *IEEE Trans. Comput. Imaging*, vol. 3, no. 4, pp. 566–579, Dec. 2017.

[29] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[30] G. Li and T. K. Pong, "Global convergence of splitting methods for nonconvex composite optimization," *SIAM J. Optim.*, vol. 25, no. 4, pp. 2434–2460, 2015.

[31] J. F. Yang and Y. Zhang, "Alternating direction algorithms for l_1-problems in compressive sensing," *SIAM J. Sci. Comput.*, vol. 33, no. 1, pp. 250–278, 2011.

[32] R. Abiko and M. Ikehara, "Blind denoising of mixed Gaussian-impulse noise by single CNN," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Brighton, U.K., May 2019, pp. 1717–1721.

[33] F. Wang, H. Huang, and J. Liu, "Variational-based mixed noise removal with CNN deep learning regularization," *IEEE Trans. Image Process.*, vol. 29, pp. 1246–1258, 2020.