

# Learning Proximal Operators for Image Restoration

Anonymous CVPR submission

Paper ID 2043

## Abstract

Recently, several discriminative learning approaches have been proposed for effective image restoration, achieving convincing trade-off between image quality and computational efficiency. However, these methods require separate training for each restoration task (e.g., denoising, deblurring, demosaicing) and problem condition (e.g., noise level of input images). This makes it time-consuming and difficult to encompass all tasks and conditions during training. In this paper, we propose to learn generalized proximal operators, which require a single-pass training and allow reuse across various problems and conditions while achieving comparable efficiency as previous discriminative approaches. Furthermore, after being trained, our model can be easily combined with other state-of-the-art priors to further improve image restoration quality.

## 1. Introduction

Low-level vision problems, such as denoising, deconvolution and demosaicing, have to be addressed as part of most imaging and vision systems. Although a large body of work covers these classical problems, low-level vision is still a very active area. The reason is that, from a Bayesian perspective, solving them as statistical estimation problems does not only require models for the likelihood, but relies on natural image priors as a key component.

A variety of models for natural image statistics have been explored in the past. Traditionally, models for gradient statistics [29, 17], including total-variation (TV), have been a popular choice. Another line of works explores patch-based image statistics, either as per-patch sparse model [11, 36] or modeling local similarity between patches [9, 10, 13]. These prior models are general in the sense that they can be applied for various likelihoods, with the image formation and noise setting as parameters. However, the resulting optimization problems are prohibitively expensive, rendering them impractical for real-time and mobile vision.

Recently, a number of works [28, 31, 7] have addressed this issue by truncating the iterative optimization and learn-

ing discriminative image priors, tailored to the likelihood and optimization approach. While these methods allow to trade-off quality with the computational budget for a given application, the learned priors are highly specialized to the image formation and noise parameters, in contrast to optimization-based approaches. Since each individual problem instantiation requires costly learning and storing of the model coefficients, current proposals for learned priors are impractical for vision applications with dynamically changing (often continuous) parameters. This is a common scenario in most real-world vision settings, as well as applications in engineering and scientific imaging that rely on the ability rapidly prototype methods.

In this paper, we combine discriminative learned models with formal optimization methods to learn generic priors that truly share across problem domains. Using proximal optimization methods [12, 24, 3] allows us to decouple the likelihood and prior which is key to learn such shared models. It also means that we can rely on well-researched physically-motivated models for the likelihood, while learning priors from example data. By learning generalized proximal mappings as a prior model, which we dub “proximal fields”, our approach is computationally cheap while being general. We verify our technique using the same proximal field for a variety of diverse low-level image reconstruction tasks and problem conditions, demonstrating the effectiveness of our approach. Benefiting from the proximal splitting techniques, our approach can naturally be combined with existing state-of-the-art priors after being trained to further improve the reconstruction quality. In particular, we make the following contributions:

- We propose proximal fields as a convolutional model for image priors that are computationally cheap to train and are shared across different image restoration tasks and problem conditions.
- Proximal fields are formulated as proximal operators, allowing their use in advanced proximal optimization algorithms.
- We show that our approach is general by demonstrating proximal fields for diverse low-level problems,

108 such as denoising, deconvolution and inpainting, for  
 109 varying noise settings.  
 110

- 111 • We show that our method can naturally be combined  
 112 with existing likelihood and priors after being trained.  
 113

## 114 2. Related work

116 Image restoration aims at computationally enhancing the  
 117 quality of images by undoing the adverse effects of image  
 118 degradation such as noise and blur. As a key area of im-  
 119 age and signal processing it is an extremely well studied  
 120 problem and a plethora of methods exists, see for exam-  
 121 ple [23] for a recent survey. Through the successful ap-  
 122 plication of machine learning and data-driven approaches, image  
 123 restoration has seen revived interest and much progress in  
 124 recent years. Broadly speaking, recently proposed state-of-  
 125 the-art methods can be grouped into three classes: *classical*  
 126 approaches that make no explicit use of machine learning,  
 127 *generative* approaches that aim at probabilistic models of  
 128 undegraded natural images and *discriminative* approaches  
 129 that try to learn a direct mapping from degraded to clean  
 130 images. Unlike classical methods, methods belonging to  
 131 the latter two classes depend on the availability of training  
 132 data.

133 **Classical models.** While traditional models focus on local  
 134 image statistics and aim at maintaining edges such as total  
 135 variation [29], bilateral [34] or anisotropic diffusion mod-  
 136 els [35], more recent methods exploit the non-local statis-  
 137 tics of images [1, 9, 21, 10, 13, 33]. In particular the highly  
 138 successful BM3D method [9] searches for similar patches  
 139 within the same image and combines them through a col-  
 140 laborative filtering step.

141 **Generative learning models.** Methods of this class seek  
 142 to learn probabilistic models of undegraded natural images.  
 143 A simple, yet powerful subclass include models that ap-  
 144 proximate the sparse gradient distribution of natural im-  
 145 ages [19, 17, 18]. More expressive generative models in-  
 146 clude the fields of experts (FoE) model [27], KSVD [11]  
 147 and the EPLL model [36]. While both FoE and KVSD learn  
 148 a set of filters whose responses are assumed to be sparse,  
 149 EPLL models natural images through Gaussian Mixture  
 150 Models. All of these models have in common to be ag-  
 151 nnostic to the image restoration task, i.e. they can be used  
 152 for any image degradation and can be combined with any  
 153 likelihood and additional priors at test time.

154 **Discriminative learning models.** Recently, discriminative  
 155 models have become increasingly popular for image  
 156 restoration due to their attractive tradeoff between high im-  
 157 age restoration quality and efficiency at test time. Meth-  
 158 ods include trainable random field models such as cas-  
 159 caded shrinkage fields (CSF) [32, 31], regression tree fields  
 160 (RTF) [16], trainable nonlinear reaction diffusion (TRD)  
 161 models [7], as well as deep convolutional networks [15] and

162 other multi-layer perceptrons [4].  
 163

164 Discriminative approaches owe their computational ef-  
 165 ficiency at run-time by defining a particular feed-forward  
 166 structure whose trainable parameters are optimized for a  
 167 particular task during training. Those learned parameters  
 168 are then kept fixed at test-time resulting in a fixed compu-  
 169 tational cost. On the downside, discriminative models do  
 170 not generalize across tasks and typically necessitate sep-  
 171 arate feed-forward architectures and separate training for  
 172 each restoration task (denoising, demosaicing, deblurring,  
 173 etc.) and every possible image degradation (noise level,  
 174 Bayer pattern, blur kernel, etc.).

175 In this work, we propose our *proximal fields* model that  
 176 is able to combine the strengths of both generative and  
 177 discriminative models: it maintains the flexibility of gen-  
 178 erative models but at the same time enjoys the computa-  
 179 tional efficiency of discriminative models. While in spirit  
 180 our approach is akin to the recently proposed method of  
 181 Rosenbaum and Weiss [26], who equipped the success-  
 182 ful EPLL model with a discriminative prediction step, the  
 183 key idea in our approach is to use proximal optimization  
 184 techniques [12, 24, 3] that allow the decoupling of likeli-  
 185 hood and prior and therewith share the full advantages of a  
 186 Bayesian generative modeling approach.

187 Table 1 summarizes the properties of the most promi-  
 188 nent state-of-the-art methods and puts our own proposed  
 189 approach into perspective.

190 Table 1. Analysis of state-of-the-art methods. In the table, “Share-  
 191 able” means the model can be used for different restoration tasks  
 192 and problem conditions; “Collaborative” means the method can be  
 193 combined with other existing priors at test time.

	FoE	EPLL	BM3D	TRD	ours
Runtime efficiency			✓	✓	✓
Easy to parallelize				✓	✓
Shareable model	✓	✓	✓		✓
Collaborative	✓	✓	✓		✓

## 200 3. Proposed method

201 The seminal work of fields-of-experts (FoE) [28] general-  
 202 izes the form of filter response based regularizers in the  
 203 objective function given in Eq. 1. Vector  $\mathbf{b}$  and  $\mathbf{x}$  represents  
 204 the observed and latent (desired) image respectively, matrix  
 205  $\mathbf{A}$  is the sensing operator, matrix  $\mathbf{F}_i$  represents 2D convolu-  
 206 tion with filter  $\mathbf{f}_i$ , and function  $\phi_i$  represents the penalty  
 207 on corresponding filter response  $\mathbf{F}_i\mathbf{x}$ . The positive scalar  
 208  $\lambda$  controls relative weight between the data fidelity (likeli-  
 209 hood) and the regularization term.

$$\frac{\lambda}{2} \|\mathbf{b} - \mathbf{Ax}\|_2^2 + \sum_{i=1}^N \phi_i(\mathbf{F}_i\mathbf{x}) \quad (1)$$

211 The well-known anisotropic total-variation regularizer can  
 212 be viewed as a special case of the FoE model where  $\mathbf{f}_i$  is the  
 213

216 derivative operator  $\nabla$  and  $\phi_i$  the  $\ell_1$  norm.  
 217

218 It is difficult to directly minimize Eq. 1 when the penalty  
 219 function  $\phi_i$  is non-linear and/or non-smooth (e.g.,  $\ell_p$  norm,  
 220  $0 < p \leq 1$ ). Proximal algorithms [3, 12, 5] instead, relax  
 221 Eq. 1 and split the original problem into several easier  
 222 subproblems that are solved alternately until convergence.  
 223

224 In this paper we employ the half-quadratic-splitting  
 225 (HQS) algorithm [12] to relax Eq. 1, as it typically requires  
 226 much fewer iterations to converge compared with other proximal  
 227 methods such as ADMM [3] and PD [5]. The relaxed objective  
 228 function is given in Eq. 2:  
 229

$$\frac{\lambda}{2} \|\mathbf{b} - \mathbf{Ax}\|_2^2 + \frac{\rho}{2} \|\mathbf{z} - \mathbf{x}\|_2^2 + \sum_{i=1}^N \phi_i(\mathbf{F}_i \mathbf{z}), \quad (2)$$

230 where a slack variable  $\mathbf{z}$  is introduced to approximate  
 231  $\mathbf{x}$ , and  $\rho$  is a positive scalar. While most related ap-  
 232 proaches [17, 31] relax Eq. 1 by splitting on  $\mathbf{F}_i \mathbf{x}$  rather than  
 233  $\mathbf{x}$ , it would limit the model flexibility in our method. This  
 234 will be explained more clearly in the next sections.  
 235

236 With the HQS algorithm, Eq. 2 is iteratively minimized  
 237 by solving for the slack variable  $\mathbf{z}$  and the latent image  $\mathbf{x}$   
 238 alternately as in Eq. 3 and 4 ( $t = 1, 2, \dots, T$ ).  
 239

$$\mathbf{z}^t = \underset{\mathbf{z}}{\operatorname{argmin}} \left( \frac{\rho^t}{2} \|\mathbf{z} - \mathbf{x}^{t-1}\|_2^2 + \sum_{i=1}^N \phi_i(\mathbf{F}_i \mathbf{z}) \right), \quad (3)$$

$$\mathbf{x}^t = \underset{\mathbf{x}}{\operatorname{argmin}} (\lambda \|\mathbf{b} - \mathbf{Ax}\|_2^2 + \rho^t \|\mathbf{z}^t - \mathbf{x}\|_2^2), \quad (4)$$

240 where  $\rho^t$  increases as the iteration continues. The latter  
 241 forces  $\mathbf{z}$  to become an increasingly good approximation of  
 242  $\mathbf{x}$ , thus making Eq. 2 an increasingly good proxy for Eq. 1.  
 243

### 3.1. Proximal fields

244 The  $\mathbf{z}^t$ -update step in Eq. 3 can be viewed as a proximal  
 245 operation:

$$\mathbf{z}^t := \mathbf{prox}_{\Theta}(\mathbf{x}^{t-1}, \rho^t), \quad (5)$$

246 where  $\mathbf{prox}_{\Theta}$  is called proximal operator with model pa-  
 247 rameters  $\Theta$ , which includes a number of filters  $\mathbf{f}_i$  and cor-  
 248 responding penalty functions  $\phi_i$ . To distinguish it from tra-  
 249 ditional proximal operators which typically contain a single  
 250 filter, we call our generalized proximal operators  $\mathbf{prox}_{\Theta}$  as  
 251 *proximal fields*.

252 Inspired by the state-of-the-art discriminative meth-  
 253 ods [31, 7], we propose to learn the proximal fields model  
 254  $\mathbf{prox}_{\Theta}$  and the weight scalar  $\lambda$  from training data. With  
 255 the above HQS relaxation, the image prior and data-fidelity  
 256 term in the original objective (Eq. 1) are contained in two  
 257 separate subproblems (Eq. 3 and 4). This makes it possi-  
 258 ble to train an ensemble of diverse tasks (e.g., denoising,  
 259 deblurring, or with different noise levels) each of which has  
 260

261 its own data fidelity term and weight  $\lambda$ , while learning a sin-  
 262 gle prior model  $\mathbf{prox}_{\Theta}$  that is shared across ensemble tasks.  
 263 This is in contrast to state-of-the-art discriminative methods  
 264 such as CSF [31] and TRD [7] which train separate priors  
 265 for each task.  
 266

267 The proximal operator  $\mathbf{prox}_{\Theta}(\mathbf{x}^{t-1}, \rho^t)$  can be inter-  
 268 preted as processing  $\mathbf{x}^{t-1}$  corrupted by zero-mean Gaus-  
 269 sian noise. With this interpretation, we propose to define  
 270  $\mathbf{prox}_{\Theta}(\mathbf{x}^{t-1})$  as a multi-stage non-linear diffusion process  
 271 modified from the TRD [7] model, as given in Eq. 6.  
 272

$$\begin{aligned} \mathbf{z}_k^t &= \mathbf{z}_{k-1}^t - \sum_{i=1}^N \mathbf{F}_i^k \psi_i^k(\mathbf{F}_i^k \mathbf{z}_{k-1}^t), \\ \text{s.t. } \mathbf{z}_0^t &= \mathbf{x}^{t-1}, \quad k = 1, 2, \dots, K. \end{aligned} \quad (6)$$

273 where  $k$  is the stage index, filters  $\mathbf{F}_i^k$ , function  $\psi_i^k$  are  
 274 trainable model parameters at each stage, and  $\mathbf{z}_0^t$  is the  
 275 initial value of  $\mathbf{z}_k^t$ . Note that, different from TRD, our  
 276 model does not contain the reaction term which would be  
 277  $-\rho^t \alpha_k(\mathbf{z}_{k-1}^t - \mathbf{x}^{t-1})$  with step size  $\alpha_k$ . The main reasons  
 278 for this modification are:  
 279

- The data constraint is contained in the  $\mathbf{x}^t$  update in Eq. 4;
- More importantly, our model gets rid of the weight  $\rho^t$  which changes at each HQS iteration. Therefore, our proximal operator  $\mathbf{prox}_{\Theta}(\mathbf{x}^{t-1}, \rho^t)$  is simplified to be:

$$\mathbf{z}^t := \mathbf{prox}_{\Theta}(\mathbf{x}^{t-1}) \quad (7)$$

280 Note that our proximal fields model  $\Theta = \{\mathbf{F}_i^k, \psi_i^k | k \in$   
 281  $\{1, \dots, K\}\}$  is re-used at all HQS iteration  $t$ , making it dif-  
 282 ferent than previous discriminative methods (CSF, TRD).  
 283 The parameters to learn in our method  $\Omega$  includes  $\lambda$ 's for  
 284 each problem class  $p$  (restoration task and problem condi-  
 285 tion), and the proximal fields model  $\Theta$  shared across dif-  
 286 ferent classes, i.e.,  $\Omega = \{\lambda_p, \Theta\}$ . Even though the scalar  
 287 parameters  $\lambda_p$  are trained, our method allows users to ad-  
 288 just them at test time for best performance and non-trained  
 289 problem classes. This contrasts to previous discriminative  
 290 approaches whose model parameters are fixed at test time.  
 291 The subscript  $p$  indicating the problem class in  $\lambda_p$  is omitted  
 292 below for convenience. The values of  $\rho^t$  are pre-selected:  
 293  $\rho^1 = 1$  and  $\rho^t = 2\rho^{t-1}$  for  $t > 1$ .  
 294

295 Note that a multi-stage model as in Eq. 6 is not possible  
 296 if we split on  $\mathbf{F}_i \mathbf{x}$  instead of  $\mathbf{x}$  in Eq. 1 and 2. For clarity, an  
 297 overview of the proposed algorithm is given in Algorithm 1.  
 298

### 3.2. Learning

299 We consider denoising and deconvolution tasks at train-  
 300 ing, where the sensing operator  $\mathbf{A}$  is an identity matrix, or a  
 301

**324 Algorithm 1** Proposed algorithm**325 Input:** degraded image  $\mathbf{b}$ , weight  $\lambda$  (optional)**326 Output:** recovered image  $\mathbf{x}$ 

```

327   1:  $\mathbf{x}^0 = \mathbf{b}, \rho^1 = 1$  (initialization)
328   2: for  $t = 1$  to  $T$  do
329     3:   (Update  $\mathbf{z}^t$  by Eq. 6 below)
330     4:    $\mathbf{z}_0^t = \mathbf{x}^{t-1}$ 
331     5:   for  $k = 1$  to  $K$  do
332       6:      $\mathbf{z}_k^t = \mathbf{z}_{k-1}^t - \sum_{i=1}^N \mathbf{F}_i^k \psi_i^k(\mathbf{F}_i^k \mathbf{z}_{k-1}^t)$ 
333     7:   end for
334     8:    $\mathbf{z}^t = \mathbf{z}_K^t$ 
335     9:   (Update  $\mathbf{x}^t$  by Eq. 4 below)
336    10:    $\mathbf{x}^t = \operatorname{argmin}_{\mathbf{x}} \lambda \|\mathbf{b} - \mathbf{Ax}\|_2^2 + \rho^t \|\mathbf{z}^t - \mathbf{x}\|_2^2$ 
337    11:    $\rho^{t+1} = 2\rho^t$ 
338  12: end for

```

340  
341 block circulant matrix with circulant blocks that represents  
342 2D convolution with blur kernels respectively. In denoising  
343 tasks,  $\mathbf{x}^t$  update in Eq. 4 has closed-form solution:

$$\mathbf{x}^t = \frac{\lambda \mathbf{b} + \rho^t \mathbf{z}^t}{\lambda + \rho^t} \quad (8)$$

344  
345 In deconvolution tasks,  $\mathbf{x}^t$  update in Eq. 4 has closed-form  
346 solution in Fourier domain:

$$\mathbf{x}^t = \mathcal{F}^{-1} \left( \frac{\mathcal{F}(\lambda \mathbf{A}^\top \mathbf{b} + \rho^t \mathbf{z}^t)}{\mathcal{F}(\lambda \mathbf{A}^\top \mathbf{A} + \rho^t)} \right), \quad (9)$$

347  
348 where  $\mathcal{F}$  and  $\mathcal{F}^{-1}$  represent Fourier and inverse Fourier  
349 transform respectively. Note that, different than CSF [31],  
350 our method does not require FFT computation for denoising  
351 tasks.

352 We use L-BFGS solver [30] with analytic gradients for  
353 training. The training loss function  $\ell$  is defined as the  
354 average Peak Signal-to-Noise Ratio (PSNR) of reconstructed  
355 images. The gradient of  $\ell$  w.r.t. the model parameters  
356  $\Omega = \{\lambda, \Theta\}$  is computed by accumulating gradients at all  
357 HQS iterations, as shown in Eq. 10.

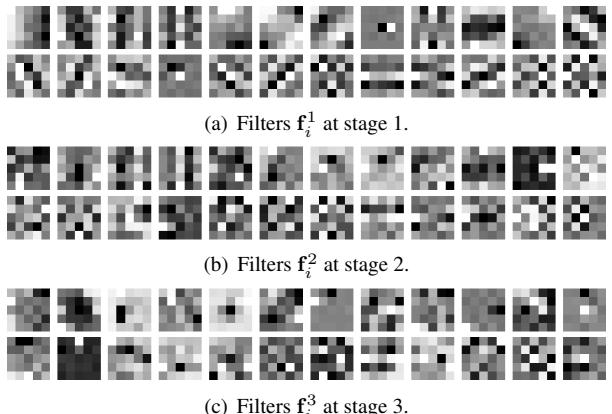
$$\frac{\partial \ell}{\partial \Omega} = \sum_{t=1}^T \left( \frac{\partial \mathbf{x}^t}{\partial \lambda} \frac{\partial \ell}{\partial \mathbf{x}^t} + \frac{\partial \mathbf{z}^t}{\partial \Theta} \left( \frac{\partial \ell}{\partial \mathbf{z}^t} + \frac{\partial \mathbf{x}^t}{\partial \mathbf{z}^t} \frac{\partial \ell}{\partial \mathbf{x}^t} \right) \right), \quad (10)$$

366 where the back-propagation inside our proximal operator is  
367 computed as:

$$\frac{\partial \mathbf{z}^t}{\partial \Theta} = \sum_{k=1}^K \frac{\partial \mathbf{z}_k^t}{\partial \Theta} \frac{\partial \mathbf{z}^t}{\partial \mathbf{z}_k^t} \quad (11)$$

372 The 1D functions  $\psi_i^k$  in Eq. 6 are parameterized as a linear  
373 combination of equidistant-positioned Gaussian kernels  
374 whose weights are trainable.

375 **Progressive training.** A progressive scheme is proposed to  
376 make the training more effectively. First, we set the num-  
377 ber of HQS iterations to be 1, and train  $\lambda$ 's and the model



378  
379 (a) Filters  $f_i^1$  at stage 1.  
380  
381 (b) Filters  $f_i^2$  at stage 2.  
382  
383 (c) Filters  $f_i^3$  at stage 3.  
384  
385

386  
387 Figure 1. Trained filters at each stage ( $k$  in Eq. 6) of the proximal  
388 operator  $\text{prox}_\Theta$  in our model (3 stages each with 24 5×5 filters).  
389  
390

391  
392  $\Theta$  of each stage in  $\text{prox}_\Theta$  in a greedy fashion. Then, we  
393 gradually increase the number of HQS iterations from 1 to  
394  $T$  where at each step the model  $\Omega = \{\lambda, \Theta\}$  is refined from  
395 the result by previous step. The L-BFGS iterations are set  
396 to be 200 for the greedy training steps, and 100 for the re-  
397 fining steps. Fig. 1 shows examples of our learned filters in  
398  $\text{prox}_\Theta$ .

## 4. Results

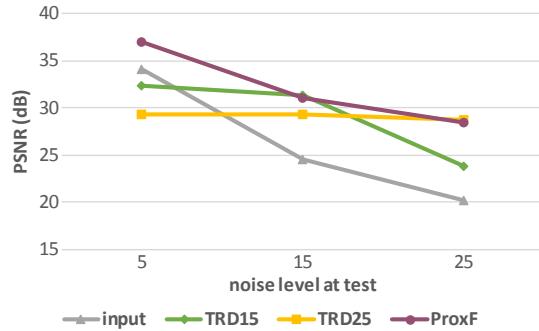
**Denoising.** We compare our method with state-of-the-art image denoising techniques, including KSVD [11], FoE [28], BM3D [9], LSSC [22], WNNM [13], EPLL [36], opt-MRF [6], ARF [2], CSF [31] and TRD [7]. Our method is denoted in short as “ProxF”. The subscript in CSF<sub>5</sub> and TRD<sub>5</sub> indicates the number of cascaded stages (each stage has different model parameters). The subscript and superscript in ProxF<sub>3</sub><sup>5</sup> indicate the number of diffusion stages ( $K = 3$  in Algorithm 1) in our proximal operator  $\text{prox}_\Theta$ , and the number of HQS iterations ( $T = 5$  in Algorithm 1), respectively. Note that the complexity (size) of our model is linear to  $K$ , but independent of  $T$ . CSF, TRD and ProxF use 24 filters of size 5×5 pixels at all stages in this section.

The compared discriminative methods, CSF<sub>5</sub> and TRD<sub>5</sub> both are trained at single noise level  $\sigma = 15$  that is the same as the test images. In contrast, our model is trained on 400 images (100×100 pixels) cropped from [28] with random and discrete noise levels (standard deviation  $\sigma$ ) varying between 5 and 25. The images with the same noise level share the same data fidelity weight  $\lambda$  at training.

All the methods are evaluated on the 68 test images from [28] and the averaged PSNR values are reported in Table 2. Our results are comparable to generic methods such as KSVD, FoE and BM3D, and very close to discriminative methods such as CSF<sub>5</sub>, while at the same time being much more efficient which is demonstrated later. Besides, we simply use  $\lambda$  learned for images with noise  $\sigma = 15$  at

432 Table 2. Average PSNR (dB) on 68 images from [28] for image  
433 denoising.

	KSVD	FoE	BM3D	LSSC	WNNM	EPLL
434	30.87	30.99	31.08	31.27	31.37	31.19
435	opt-MRF	ARF	CSF <sub>5</sub>	TRD <sub>5</sub>	ProxF <sub>3</sub> <sup>3</sup>	ProxF <sub>3</sub> <sup>5</sup>
436	31.18	30.70	31.14	31.30	30.91	31.00



440 Figure 2. Analysis of model generality on image denoising. In this  
441 plot, “TRD15” denotes the TRD model trained at noise  $\sigma = 15$ ,  
442 and “TRD25” at noise  $\sigma = 25$ . “ProxF” denotes our model trained  
443 with mixed noise levels in a single pass.

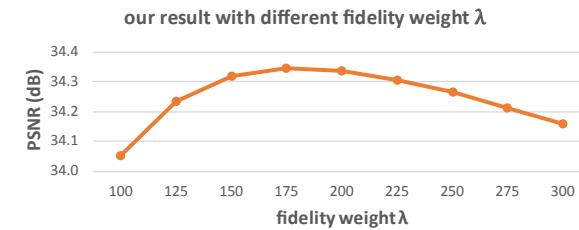
444 training to generate all the test results, although adjusting its  
445 value at test time can expectedly improve our results. Note  
446 that the compared discriminative methods (CSF, TRD) do  
447 not allow for such parameter tuning.

448 To verify the generality of our method on varying noise  
449 levels, we test our model ProxF<sub>3</sub><sup>3</sup> (trained with varying noise  
450 levels in a single pass) and two TRD models (trained at spe-  
451 cific noise levels 15 and 25) on 3 sets of 68 images with  
452 noise  $\sigma = 5, 15, 25$  respectively. The average PSNR val-  
453 ues are shown in Fig. 2. Although performing slightly be-  
454 low the TRD model trained for the exact noise level used  
455 at test time, our method is more generic and works robustly  
456 for various noise levels. In sharp contrast to discriminative  
457 methods, which are inherently specialized for a given prob-  
458 lem setting, i.e. noise level, the proposed approach transfers  
459 across different problem settings. We demonstrate this  
460 generality below for a variety of different image reconstruc-  
461 tion tasks.

462 **Run-time comparison.** In Table 3 we compare the run-  
463 time of our method and state-of-the-art methods. The ex-  
464 periments were performed on a laptop computer with Intel  
465 i7-4720HQ CPU and 16GB RAM. WNNM and EPLL ran  
466 out-of-memory for images over 4 megapixels in our ex-  
467 periments. CSF<sub>5</sub>, TRD<sub>5</sub> and our method ProxF<sub>3</sub><sup>3</sup> all use “par-  
468 for” setting in Matlab. ProxF<sub>3</sub><sup>3</sup> is significantly faster than all  
469 compared generic methods (WNNM, EPLL, BM3D) and  
470 even the discriminative method CSF<sub>5</sub>. Run-time of ProxF<sub>3</sub><sup>3</sup>  
471 is about 1.5 times that of TRD<sub>5</sub>, which is expected as they  
472 use 9 versus 5 diffusion steps in total. In addition, we im-  
473 plement our method in Halide language [25], which has be-  
474 come popular recently for high-performance image process-

486 Table 3. Runtime (seconds) comparison for image denoising on  
487 different size images.

Image size	256 <sup>2</sup>	512 <sup>2</sup>	1024 <sup>2</sup>	2048 <sup>2</sup>	4096 <sup>2</sup>
488 WNNM	157.73	657.75	2759.79	-	-
489 EPLL	29.21	111.52	463.71	-	-
490 BM3D	0.78	3.45	15.24	62.81	275.39
491 CSF <sub>5</sub>	1.23	2.22	7.35	27.08	93.66
492 TRD <sub>5</sub>	0.39	0.71	2.01	7.57	29.09
493 ProxF <sub>3</sub> <sup>3</sup>	0.60	1.19	3.45	12.97	56.19
494 ProxF <sub>3</sub> <sup>3</sup> (Halide)	0.11	0.26	1.60	5.61	20.85



500 Figure 3. Our results with different fidelity weight  $\lambda$  for the non-  
501 blind deconvolution experiment reported in Table 4.

502 Table 4. Average PSNR (dB) on 32 images from [20] for non-blind  
503 deconvolution.

Input	Levin [19]	Schmidt [32]	CSF <sub>3</sub> <sup>PW</sup>	ProxF <sub>3</sub> <sup>3</sup>
508 22.86	32.73	33.97	33.48	34.34

512 ing applications, and report the run-time on the same CPU  
513 as mentioned above.

514 **Deconvolution.** Our general model supports image decon-  
515 volution tasks in the HQS framework. In this experiment,  
516 we train a model with an ensemble of denoising and de-  
517 convolution tasks on 400 images (100×100 pixels) cropped  
518 from [28], in which 250 images are generated for denois-  
519 ing tasks with random noise levels  $\sigma$  varying between 5 and  
520 25, and the other 150 images are generated by blurring the  
521 images with random 25×25 kernels (PSFs) and then adding  
522 Gaussian noise with  $\sigma$  between 1 and 5. All input images  
523 are quantized to 8 bits.

524 We compare our method with state-of-the-art non-blind  
525 deconvolution methods including Levin et al. [19], Schmidt  
526 et al. [32] and CSF [31]. Note that TRD [7] does not support  
527 non-blind deconvolution. We test the methods on 32 images  
528 from [20] and report the average PSNR values in Table 4.  
529 The results of compared methods are quoted from [31]. We  
530 run a grid search on the adjustable fidelity weight  $\lambda$  (the  
531 same value for all the 32 test images) and report the best re-  
532 sult in Table 4. In Fig. 3, we show our results with different  
533  $\lambda$ . Within a fairly wide range of  $\lambda$ , our method outperforms  
534 all previous methods.

535 **Collaborative with existing priors.** As shown above, even  
536 though the fidelity weight  $\lambda$  is trainable, our method al-  
537 lows users to adjust its value for better performance at test  
538 time. Moreover, this property makes it possible to combine  
539 our model (after being trained) with existing state-of-the-

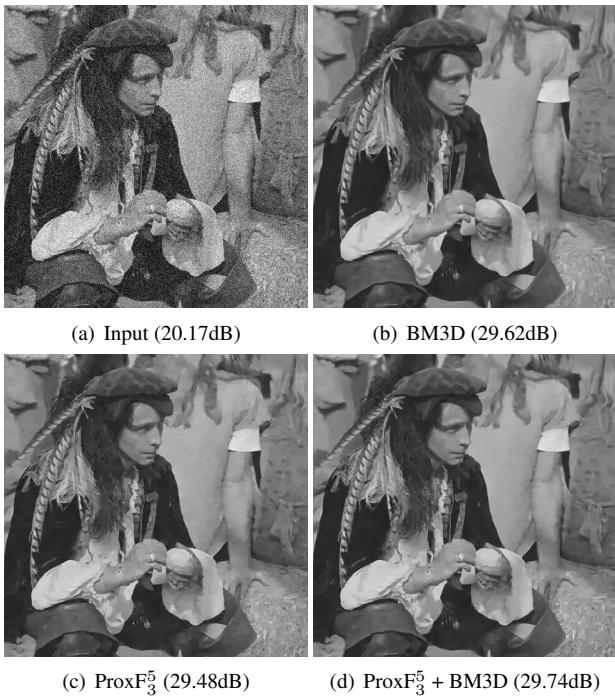


Figure 4. Experiment on incorporating non-local patch similarity prior (BM3D [8]) with our model after being trained. The input noise level  $\sigma = 25$ . Please zoom in for better view.

art priors at test time, in which case  $\lambda$  typically needs to be adjusted. Again, this is not possible with previous discriminative methods (CSF, TRD).

In Fig. 4 we show an example to incorporate a non-local patch similarity prior (BM3D [8]) with our method to further improve the denoising quality. BM3D performs well in removing noise especially in smooth regions but usually over-smoothes edges and textures. Our original model (ProxF<sup>5</sup><sub>3</sub>) well preserves sharp edges however sometimes introduces artifacts in smooth regions when the input noise level is high. By combining those two methods, which is easy with our HQS framework, the result is improved both visually and quantitatively.

We give the derivation of the proposed hybrid method below. Let  $\mathcal{S}(\mathbf{x})$  represents the non-local patch similarity prior. The objective function is:

$$\frac{\lambda}{2} \|\mathbf{b} - \mathbf{Ax}\|_2^2 + \sum_{i=1}^N \phi_i(\mathbf{F}_i \mathbf{x}) + \tau \mathcal{S}(\mathbf{x}) \quad (12)$$

Applying the HQS technique described in Sec. 3, we relax the objective to be:

$$\begin{aligned} \frac{\lambda}{2} \|\mathbf{b} - \mathbf{Ax}\|_2^2 + \frac{\rho}{2} \|\mathbf{z} - \mathbf{x}\|_2^2 + \sum_{i=1}^N \phi_i(\mathbf{F}_i \mathbf{z}) \\ + \frac{\rho_s}{2} \|\mathbf{v} - \mathbf{x}\|_2^2 + \tau \mathcal{S}(\mathbf{v}) \end{aligned} \quad (13)$$

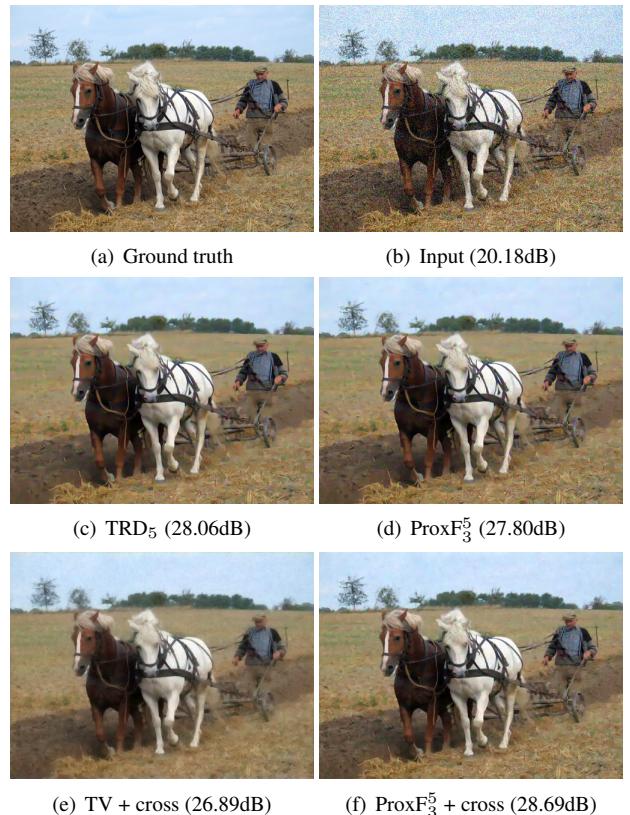


Figure 5. Experiment on incorporating a color prior [14] with our model after being trained. The input noise level  $\sigma = 25$ . (e,f) show the results by combining total variation (TV) denoising with a cross-channel prior, and our method with cross-channel prior, respectively. Please zoom in for better view.

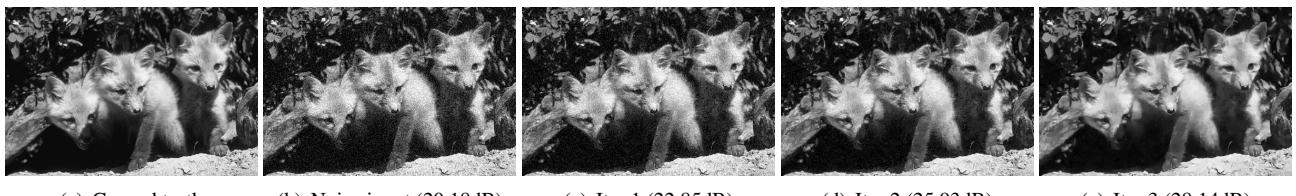
Then we minimize Eq. 13 by alternately solving the following 3 subproblems:

$$\begin{aligned} \mathbf{z}^t &= \mathbf{prox}_{\Theta}(\mathbf{x}^{t-1}) \\ \mathbf{v}^t &= \underset{\mathbf{v}}{\operatorname{argmin}} \frac{\rho_s^t}{2} \|\mathbf{v} - \mathbf{x}^{t-1}\|_2^2 + \tau \mathcal{S}(\mathbf{v}) \approx \text{BM3D}(\mathbf{x}^{t-1}, \frac{\tau}{\rho_s^t}) \\ \mathbf{x}^t &= \underset{\mathbf{x}}{\operatorname{argmin}} \lambda \|\mathbf{b} - \mathbf{Ax}\|_2^2 + \rho^t \|\mathbf{z}^t - \mathbf{x}\|_2^2 + \frac{\rho_s^t}{2} \|\mathbf{v}^t - \mathbf{x}\|_2^2, \end{aligned} \quad (14)$$

where  $\mathbf{prox}_{\Theta}$  is from our previous training, and the  $\mathbf{v}^t$  subproblem is *approximated* by running BM3D software on  $\mathbf{x}^{t-1}$  with noise parameter  $\frac{\tau}{\rho_s^t}$  following [14].

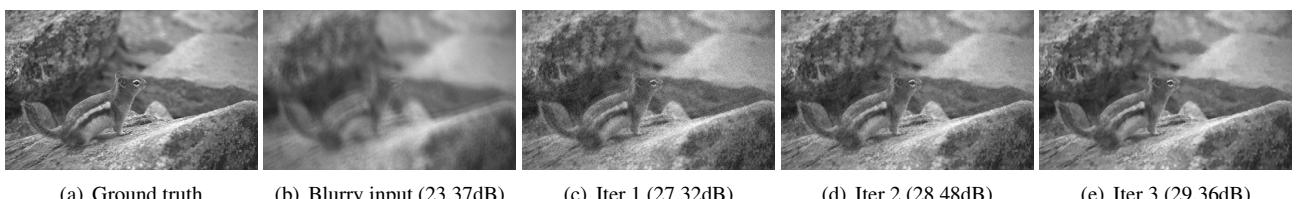
Similarly, our method can incorporate color image priors (e.g., cross-channel edge-concurrence prior [14]) to improve test results on color images, despite our model being trained on gray-scale images. An example is shown in Fig. 5. The hybrid method shares the advantages of our original model that effectively preserves edges and textures and the cross-channel prior that reduces color artifacts.

**Transferability to unseen tasks.** Our method allows for new data-fidelity terms that are not contained in training,



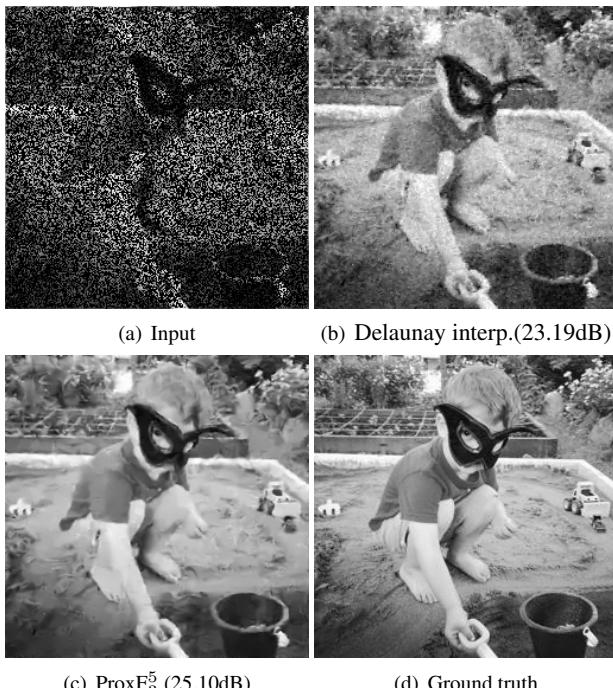
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701

Figure 6. Results at each HQS iteration of our method on image denoising with noise level  $\sigma = 25$ . Inside brackets show the PSNR values. Please zoom in for better view.



702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755

Figure 7. Results at each HQS iteration of our method on non-blind deconvolution with a  $25 \times 25$  PSF and noise level  $\sigma = 3$ . Inside brackets show the PSNR values. Please zoom in for better view.



756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
8010  
8011  
8012  
8013  
8014  
8015  
8016  
8017  
8018  
8019  
8020  
8021  
8022  
8023  
8024  
8025  
8026  
8027  
8028  
8029  
8030  
8031  
8032  
8033  
8034  
8035  
8036  
8037  
8038  
8039  
8040  
8041  
8042  
8043  
8044  
8045  
8046  
8047  
8048  
8049  
8050  
8051  
8052  
8053  
8054  
8055  
8056  
8057  
8058  
8059  
8060  
8061  
8062  
8063  
8064  
8065  
8066  
8067  
8068  
8069  
8070  
8071  
8072  
8073  
8074  
8075  
8076  
8077  
8078  
8079  
8080  
8081  
8082  
8083  
8084  
8085  
8086  
8087  
8088  
8089  
8090  
8091  
8092  
8093  
8094  
8095  
8096  
8097  
8098  
8099  
80100  
80101  
80102  
80103  
80104  
80105  
80106  
80107  
80108  
80109  
80110  
80111  
80112  
80113  
80114  
80115  
80116  
80117  
80118  
80119  
80120  
80121  
80122  
80123  
80124  
80125  
80126  
80127  
80128  
80129  
80130  
80131  
80132  
80133  
80134  
80135  
80136  
80137  
80138  
80139  
80140  
80141  
80142  
80143  
80144  
80145  
80146  
80147  
80148  
80149  
80150  
80151  
80152  
80153  
80154  
80155  
80156  
80157  
80158  
80159  
80160  
80161  
80162  
80163  
80164  
80165  
80166  
80167  
80168  
80169  
80170  
80171  
80172  
80173  
80174  
80175  
80176  
80177  
80178  
80179  
80180  
80181  
80182  
80183  
80184  
80185  
80186  
80187  
80188  
80189  
80190  
80191  
80192  
80193  
80194  
80195  
80196  
80197  
80198  
80199  
80200  
80201  
80202  
80203  
80204  
80205  
80206  
80207  
80208  
80209  
80210  
80211  
80212  
80213  
80214  
80215  
80216  
80217  
80218  
80219  
80220  
80221  
80222  
80223  
80224  
80225  
80226  
80227  
80228  
80229  
80230  
80231  
80232  
80233  
80234  
80235  
80236  
80237  
80238  
80239  
80240  
80241  
80242  
80243  
80244  
80245  
80246  
80247  
80248  
80249  
80250  
80251  
80252  
80253  
80254  
80255  
80256  
80257  
80258  
80259  
80260  
80261  
80262  
80263  
80264  
80265  
80266  
80267  
80268  
80269  
80270  
80271  
80272  
80273  
80274  
80275  
80276  
80277  
80278  
80279  
80280  
80281  
80282  
80283  
80284  
80285  
80286  
80287  
80288  
80289  
80290  
80291  
80292  
80293  
80294  
80295  
80296  
80297  
80298  
80299  
80300  
80301  
80302  
80303  
80304  
80305  
80306  
80307  
80308  
80309  
80310  
80311  
80312  
80313  
80314  
80315  
80316  
80317  
80318  
80319  
80320  
80321  
80322  
80323  
80324  
80325  
80326  
80327  
80328  
80329  
80330  
80331  
80332  
80333  
80334  
80335  
80336  
80337  
80338  
80339  
80340  
80341  
80342  
80343  
80344  
80345  
80346  
80347  
80348  
80349  
80350  
80351  
80352  
80353  
80354  
80355  
80356  
80357  
80358  
80359  
80360  
80361  
80362  
80363  
80364  
80365  
80366  
80367  
80368  
80369  
80370  
80371  
80372  
80373  
80374  
80375  
80376  
80377  
80378  
80379  
80380  
80381  
80382  
80383  
80384  
80385  
80386  
80387  
80388  
80389  
80390  
80391  
80392  
80393  
80394  
80395  
80396  
80397  
80398  
80399  
80400  
80401  
80402  
80403  
80404  
80405  
80406  
80407  
80408  
80409  
80410  
80411  
80412  
80413  
80414  
80415  
80416  
80417  
80418  
80419  
80420  
80421  
80422  
80423  
80424  
80425  
80426  
80427  
80428  
80429  
80430  
80431  
80432  
80433  
80434  
80435  
80436  
80437  
80438  
80439  
80440  
80441  
80442  
80443  
80444  
80445  
80446  
80447  
80448  
80449  
80450  
80451  
80452  
80453  
80454  
80455  
80456  
80457  
80458  
80459  
80460  
80461  
80462  
80463  
80464  
80465  
80466  
80467  
80468  
80469  
80470  
80471  
80472  
80473  
80474  
80475  
80476  
80477  
80478  
80479  
80480  
80481  
80482  
80483  
80484  
80485  
80486  
80487  
80488  
80489  
80490  
80491  
80492  
80493  
80494  
80495  
80496  
80497  
80498  
80499  
80500  
80501  
80502  
80503  
80504  
80505  
80506  
80507  
80508  
80509  
80510  
80511  
80512  
80513  
80514  
80515  
80516  
80517  
80518  
80519  
80520  
80521  
80522  
80523  
80524  
80525  
80526  
80527  
80528  
80529  
80530  
80531  
80532  
80533  
80534  
80535  
80536  
80537  
80538  
80539  
80540  
80541  
80542  
80543  
80544  
80545  
80546  
80547  
80548  
80549  
80550  
80551  
80552  
80553  
80554  
80555  
80556  
80557  
80558  
80559  
80560  
80561  
80562  
80563  
80564  
80565  
80566  
80567  
80568  
80569  
80570  
80571  
80572  
80573  
80574  
80575  
80576  
80577  
80578  
80579  
80580  
80581  
80582  
80583  
80584  
80585  
80586  
80587  
80588  
80589  
80590  
80591  
80592  
80593  
80594  
80595  
80596  
80597  
80598  
80599  
80600  
80601  
80602  
80603  
80604  
80605  
80606  
80607  
80608  
80609  
80610  
80611  
80612  
80613  
80614  
80615  
80616  
80617  
80618  
80619  
80620  
80621  
80622  
80623  
80624  
80625  
80626  
80627  
80628  
80629  
80630  
80631  
80632  
80633  
80634  
80635  
80636  
80637  
80638  
80639  
80640  
80641  
80642  
80643  
80644  
80645  
80646  
80647  
80648  
80649  
80650  
80651  
80652  
80653  
80654  
80655  
80656  
80657  
80658  
80659  
80660  
80661  
80662  
80663  
80664  
80665  
80666  
80667  
80668  
80669  
80670  
80671  
80672  
80673  
80674  
80675  
80676  
80677  
80678  
80679  
80680  
80681  
80682  
80683  
80684  
80685  
80686  
80687  
80688  
80689  
80690  
80691  
80692  
80693  
80694  
80695  
80696  
80697  
80698  
80699  
80700  
80701  
80702  
80703  
80704  
80705  
80706  
80707  
80708  
80709  
80710  
80711  
80712  
80713  
80714  
80715  
80716  
80717  
80718  
80719  
80720  
80721  
80722  
80723  
80724  
80725  
80726  
80727  
80728  
80729  
80730  
80731  
80732  
80733  
80734  
80735  
80736  
80737  
80738  
80739  
80740  
80741  
80742  
80743  
80744  
80745  
80746  
80747  
80748  
80749  
80750  
80751  
80752  
80753  
80754  
80755  
80756  
80757  
80758  
80759  
80760  
80761  
80762  
80763  
80764  
80765  
80766  
80767  
80768  
80769  
80770  
80771  
80772  
80773  
80774  
80775  
80776  
80777  
80778  
80779  
80780  
80781  
80782  
80783  
80784  
80785  
80786  
80787  
80788  
80789  
80790  
80791  
80792  
80793  
80794  
80795  
80796  
80797  
80798  
80799  
80800  
80801  
80802  
80803  
80804  
80805  
80806  
80807  
80808  
80809  
80810  
80811  
80812  
80813  
80814  
80815  
80816  
80817  
80818  
80819  
80820  
80821  
80822  
80823  
80824  
80825  
80826  
80827  
80828  
80829  
80830  
80831  
80832  
80833  
80834  
80835  
80836  
80837  
80838  
80839  
80840  
80841  
80842  
80843  
80844  
80845  
80846  
80847  
80848  
80849  
80850  
80851  
80852  
80853  
80854  
80855  
80856  
80857  
80858  
80859  
80860  
80861  
80862  
80863  
80864  
80865  
80866  
80867  
80868  
80869  
80870  
80871  
80872  
80873  
80874  
80875  
80876  
80877  
80878  
80879  
80880  
80881  
80882  
80883  
80884  
80885  
80886  
80887  
80888  
80889  
80890  
80891  
80892  
80893  
80894  
80895  
80896  
80897  
80898  
80899  
80900  
80901  
80902  
80903  
80904  
80905  
80906  
80907  
80908  
80909  
80910  
80911  
80912  
80913  
80914  
80915  
80916  
80917  
80918  
80919  
80920  
80921  
80922  
80923  
80924  
80925  
80926  
80927  
80928  
80929  
80930  
80931  
80932  
80933  
80934  
80935  
80936  
80937  
80938  
80939  
80940  
80941  
80942  
80943  
80944  
80945  
80946  
80947  
80948  
80949  
80950  
80951  
80952  
80953  
80954  
80955  
80956  
80957  
80958  
80959  
80960  
80961  
80962  
80963  
80964  
80965  
80966  
80967  
80968  
80969  
80970  
80971  
80972  
80973  
80974  
80975  
80976  
80977  
80978  
80979  
80980  
80981  
80982  
80983  
80984  
80985  
80986  
80987  
80988  
80989  
80990  
80991  
80992  
80993  
80994  
80995  
80996  
80997  
80998  
80999  
80100  
80101  
80102  
80103  
80104  
80105  
80106  
80107  
80108  
80109  
80110  
80111  
80112  
80113  
80114  
80115  
80116  
80117  
80118  
80119  
80120  
80121  
80122  
80123  
80124  
80125  
80126  
80127  
80128  
80129  
80130  
80131  
80132  
80133  
80134  
80135  
80136  
80137  
80138  
80139  
80140  
80141  
80142  
80143  
80144  
80145  
80146  
80147  
80148  
80149  
80150  
80151  
80152  
80153  
80154  
80155  
80156  
80157  
80158  
80159  
80160  
80161  
80162  
80163  
80164  
80165  
80166  
80167  
80168  
80169  
80170  
80171  
80172  
80173  
80174  
80175  
80176  
80177  
80178  
80179  
80180  
80181  
80182  
80183  
80184  
80185  
80186  
80187  
80188  
80189  
80190  
80191  
80192  
80193  
80194  
80195  
80196  
80197  
80198  
80199  
80200  
80201  
80202  
80203  
80204  
80205  
80206  
80207  
80208  
80209  
80210  
80211  
80212  
80213  
80214  
80215  
80216  
80217  
80218  
80219  
80220  
80221  
80222  
80223  
80224  
80225  
80226  
80227  
80228  
80229  
80230  
80231  
80232  
80233  
80234  
80235  
80236  
80237  
80238  
80239  
80240  
80241  
80242  
80243  
80244  
80245  
80246  
80247  
80248  
80249  
80250  
80251  
80252  
80253  
80254  
80255  
80256  
80257  
80258  
80259  
80260  
80261  
80262  
80263  
80264  
80265  
80266  
80267  
80268  
80269  
80270  
80271  
80272  
80273  
80274  
80275  
80276  
80277  
80278  
80279  
80280  
80281  
80282  
80283  
80284  
80285  
80286  
80287  
80288  
80289  
80290  
80291  
80292  
80293  
80294  
80295  
80296  
80297  
80298  
80299  
80299  
80300  
80301  
80302  
80303  
80304  
80305  
80306  
80307  
80308  
80309  
80310  
80311  
80312  
80313  
80314  
80315  
80316  
80317  
80318  
80319  
80320  
80321  
80322  
80323  
80324  
80325  
80326  
80327  
80328  
80329  
80330  
80331  
80332  
80333  
80334  
80335  
80336  
80337  
80338  
80339  
80340  
80341  
80342  
80343  
80344  
80345  
80346  
80347  
80348  
80349  
80350  
80351  
80352  
80353  
80354  
80355  
80356  
80357  
80358  
80359  
80360  
80361  
80362  
80363  
80364  
80365  
80366  
80367  
80368  
80369  
80370  
80371  
80372  
80373  
80374  
80375  
80376  
80377  
80378  
80379  
80380  
80381  
80382  
80383  
80384  
80385  
80386  
80387  
80388  
80389  
80390  
80391  
80392  
80393  
80394  
80395  
80396  
80397  
80398  
80399  
80399  
80400  
80401  
80402  
80403  
80404  
80405  
80406  
80407  
80408  
80409  
80410  
80411  
80412  
80413  
80414  
80415  
80416  
80417  
80418  
80419  
80420  
80421  
80422  
80423  
80424  
80425  
80426  
80427  
80428  
80429  
80430  
80431  
80432  
80433  
80434  
80435  
80436  
80437  
80438  
80439  
80440  
80441  
80442  
80443  
80444  
80445  
80446  
80447  
80448  
80449  
80450  
80451  
80452  
80453  
80454  
80455  
80456  
80457  
80458  
80459  
80460  
80461  
80462  
80463  
80464  
80465  
80466  
80467  
80468  
80469  
80470  
80471  
80472  
80473  
80474  
80475  
80476  
80477  
80478  
80479  
80479  
80480  
80481  
80482  
80483  
80484  
80485  
80486  
80487  
80488  
80489  
80490  
80491  
80492  
80493  
80494  
80495  
80496  
80497  
80498  
80499  
80499  
80500  
80501  
80502  
80503  
80504  
80505  
80506  
80507

756

## 5. Conclusion

757

In this paper, we proposed the trainable proximal fields model, a generalization of traditional proximal operators. By combining advanced proximal optimization algorithms and discriminative learning techniques, a single training pass leads to a transferable model useful for a variety of image restoration tasks and problem conditions. Furthermore, our method is flexible and can be combined with existing priors and likelihood terms after being trained, allowing to improve image quality on a task at hand. In spite of this generality, our method achieves comparable run-time efficiency as previous discriminative approaches, making it suitable for high-resolution image restoration and mobile vision applications.

771

We believe that in future work, our framework incorporating advanced optimization with discriminative learning techniques can be extended to deep learning, for training more compact and shareable models, and to solve high-level vision problems. Another plan is to train our models for ensemble tasks with larger datasets and use advanced learning optimization techniques, which can potentially further improve results.

772

## References

780

- [1] B. C. A. Buades and J. M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling and Simulation*, 4(2):490–530, 2005. 2
- [2] A. Barbu. Training an active random field for real-time image denoising. *IEEE Transactions on Image Processing*, 18(11):2451–2462, 2009. 4
- [3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011. 1, 2, 3
- [4] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with BM3D? In *CVPR 2012*. 2
- [5] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011. 3
- [6] Y. Chen, T. Pock, R. Ranftl, and H. Bischof. Revisiting loss-specific training of filter-based mrfs for image restoration. In *German Conference on Pattern Recognition 2013*. 4
- [7] Y. Chen, W. Yu, and T. Pock. On learning optimized reaction diffusion processes for effective image restoration. In *CVPR 2015*. 1, 2, 3, 4, 5
- [8] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image restoration by sparse 3d transform-domain collaborative filtering. In *Electronic Imaging 2008*. 6
- [9] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. 1, 2, 4
- [10] W. Dong, L. Zhang, G. Shi, and X. Li. Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing*, 22(4):1620–1630, 2013. 1, 2
- [11] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006. 1, 2, 4
- [12] D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE Transactions on Image Processing*, 4(7):932–946, 1995. 1, 2, 3
- [13] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *CVPR 2014*. 1, 2, 4
- [14] F. Heide, M. Steinberger, Y.-T. Tsai, M. Rouf, D. Pajak, D. Reddy, O. Gallo, J. Liu, W. Heidrich, K. Egiazarian, et al. Flexisp: a flexible camera image processing framework. *ACM Transactions on Graphics (TOG)*, 33(6):231, 2014. 6
- [15] V. Jain and H. Seung. Natural image denoising with convolutional networks. *NIPS 2009*. 2
- [16] J. Jancsary, S. Nowozin, T. Sharp, and C. Rother. Regression tree fields - an efficient, non-parametric approach to image labeling problems. In *CVPR 2012*. 2
- [17] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In *NIPS 2009*. 1, 2, 3
- [18] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *CVPR 2011*. 2
- [19] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM transactions on graphics (TOG)*, 26(3):70, 2007. 2, 5
- [20] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *CVPR 2011*. 5, 7
- [21] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *ICCV 2010*. 2
- [22] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *ICCV 2009*. 4
- [23] P. Milanfar. A tour of modern image filtering: New insights and methods, both practical and theoretical. *IEEE Signal Processing Magazine*, 30(1):106–128, 2013. 2
- [24] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2013. 1, 2
- [25] J. Ragan-Kelley, C. Barnes, A. Adams, S. Paris, F. Durand, and S. Amarasinghe. Halide: a language and compiler for optimizing parallelism, locality, and recomputation in image processing pipelines. *ACM SIGPLAN Notices*, 48(6):519–530, 2013. 5
- [26] D. Rosenbaum and Y. Weiss. The return of the gating network: Combining generative models and discriminative training in natural image priors. In *NIPS 2015*. 2
- [27] S. Roth and M. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, 2009. 2
- [28] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *CVPR 2005*. 1, 2, 4, 5

- 864 [29] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992. 1, 2 918  
865 919  
866 920  
867 921  
868 922  
869 923  
870 924  
871 925  
872 926  
873 927  
874 928  
875 929  
876 930  
877 931  
878 932  
879 933  
880 934  
881 935  
882 936  
883 937  
884 938  
885 939  
886 940  
887 941  
888 942  
889 943  
890 944  
891 945  
892 946  
893 947  
894 948  
895 949  
896 950  
897 951  
898 952  
899 953  
900 954  
901 955  
902 956  
903 957  
904 958  
905 959  
906 960  
907 961  
908 962  
909 963  
910 964  
911 965  
912 966  
913 967  
914 968  
915 969  
916 970  
917 971