Efficient regression priors for reducing image compression artifacts

Conference Paper

Author(s):
Rothe, Rasmus; Timofte, Radu; Van Gool, Luc

Publication date:
2015

Permanent link:
https://doi.org/10.3929/ethz-a-010537014

Rights / license:
In Copyright - Non-Commercial Use Permitted

Originally published in:
https://doi.org/10.1109/ICIP.2015.7351059

EFFICIENT REGRESSION PRIORS FOR REDUCING IMAGE COMPRESSION ARTIFACTS

Rasmus Rothe, Radu Timofte, and Luc Van Gool

Computer Vision Lab, D-ITET, ETH Zürich, Switzerland

ABSTRACT

Lossy image compression allows for large storage savings but at the cost of reduced fidelity of the compressed images. There is a fair amount of literature aiming at restoration by suppressing the compression artifacts. Very recently a learned semi-local Gaussian Processes-based solution (SLGP) has been proposed with impressive results. However, when applied to top compression schemes such as JPEG 2000, the improvement is less significant. In our paper we propose an efficient novel artifact reduction algorithm based on the adjusted anchored neighborhood regression (A+), a method from image super-resolution literature. We double the relative gains in PSNR when compared with the state-of-the-art methods such as SLGP, while being order(s) of magnitude faster.

***Index Terms*—** lossy image compression, artifact removal, regression, image enhancement, super-resolution.

1. INTRODUCTION

For the sake of reducing storage, multimedia content often is stored in compressed form. Furthermore, lossy image compression is preferred to lossless compression because of its significantly higher compression rates. This, however, results in the loss of fidelity to the original content. With the broad adoption of lossy image compression, in particular the compression artifact suppression has become a focus for research. This literature closely connects with important advances that have been made in compression algorithms.

One of the most used coding techniques is block-based discrete cosine transform (BDCT). It is used widely for compression of both images and videos (e.g., JPEG/MPEG). BDCT's main drawback is the presence of discontinuities at block boundaries, also known as block artifacts, especially for low bit rates. JPEG 2000 uses the discrete wavelet transform instead of the BDCT stage from JPEG. In this way, JPEG 2000 exhibits milder artifacts, mostly ringing artifacts. In [1] a scale-space method for ringing estimation is proposed.

There are different research directions for artifact removal seen as an image enhancement task. The main one, followed in this paper, employs the use of prior knowledge. Projection onto convex sets (POCS) models represent prior knowledge under the form of convex constraints (such as smoothness or quantization constraints). POCS models perform well for JPEG [2] and JPEG 2000 [3] image enhancement. Roth and Black [4] propose a field of experts (FoE) with learned clique potentials under Markov Random Field framework for image enhancement. The noise removal is targeted by Laparra *et al.* [5] with a non-parametric suppor vector regression (SVR) method. Tschumperle and Deriche [6] propose a single generic anisotropic diffusion equation as unifying expression for different enhancement applications.

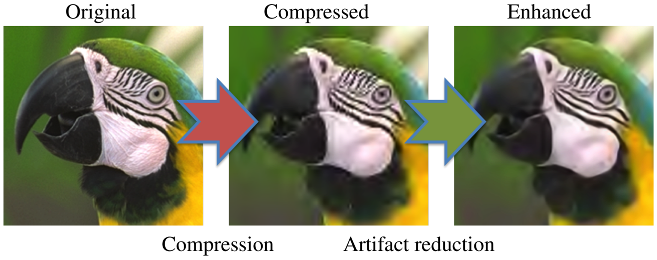

Fig. 1. Image compression artifact reduction result of our method (image 6).

Other works propose specific formulations for the compression artifact removal. Qiu [7] proposes the use of a multilayer perceptron (MLP) model. Foi *et al.* [8] applies a shape-adaptive DCT method (SADCT) pointwisely. Zhai *et al.* [9] uses a block-shift filtering-based algorithm.

Nosratinia [10] observes that the re-application of JPEG reduces the artifacts. He notices the same for JPEG 2000 [11].

Recently, Kwon *et al.* [12] proposed a common solution to image super-resolution and compression artifact removal by using Gaussian Processes (GP) under a semi-local approximation (SLGP). By the approximation scheme the time complexity of large-scale GPs decreases. Since their approach achieves the best results to date for JPEG and JPEG 2000 artifact removal, it is our main comparing method.

We propose a novel post-processing method based on the recent adjusted anchored neighborhood regression (A+) [13, 14], a state-of-the-art method in single image super-resolution. In our method, for a certain lossy compression method we learn offline linear regressors from compressed to raw train images, and then apply them to reduce the compression artifacts in test images. Based on these priors extracted from the training material we are able to reduce the artifacts and achieve state-of-the-art performance, doubling the PSNR gain of SLGP [12] while having order of magnitude lower running time.

2. PROPOSED METHOD

2.1. Overview

Our method follows closely the adjusted anchored neighborhood regression (A+) super-resolution method of Timofte *et al.* [13]. The method works with small image patches of fixed size (e.g. 7×7 pixels). The patches are extracted densely over an image grid. The offline training starts with the extraction of pairs of patches in the training compressed image (low resolution, LR) and the corresponding ones in the raw artifact-free image (high resolution, HR). The patches are used to train a sparse dictionary whose atoms/patches are taken as representatives of the compressed image space. These are the anchoring points of our method. For each we offline compute a regressor from the compressed to the artifact-free image patches (from LR to HR). At test time, we extract over a grid patches and regress them to the artifact-free image by picking the stored regressor at its nearest anchoring point. The regressed patches are averaged in the overlapped areas to form the output enhanced image.

2.2. Patches and features

The LR patches are represented by their features which are concatenated responses to 1st and 2nd order gradients applied horizontally and vertically to the LR image. We use the same features as in [13, 15, 16]. Through PCA we reduce the dimensionality of the features, while keeping 99.9% of the energy. The HR patches are represented by the difference between the ground truth HR image and the LR image. The training LR features are normalized by l_2-norm and the corresponding HR patches are scaled accordingly, as in [13]. At test time, the reconstructed image is added to the input LR image for the final output.

2.3. Anchoring points

The relation between the patches from compressed images (LR) and their corresponding artifact-free images (HR) is heavily non-linear. Instead of training a single non-linear regression function to model this complex relation, we partition the LR space around anchoring points and train local linear regressors to the HR space as in [13, 16], which results in a very good approximation.

In order to obtain the anchoring points in LR space, a dictionary $\mathbf{D}_l$, we use the K-SVD [17] method, as in [13, 15].

$$\begin{aligned} \mathbf{D}_l, \{\mathbf{w}^{(k)}\} = \quad & \text{argmin}_{\mathbf{D}_l, \{\mathbf{w}^{(k)}\}} \textstyle\sum_k \|\mathbf{p}_l^{(k)} - \mathbf{D}_l \mathbf{w}^{(k)}\|^2 \\ & \text{s.t. } \|\mathbf{w}^{(k)}\|_0 \leq L \ \forall k, \end{aligned} \tag{1}$$

where $\mathbf{p}_l^{(k)}$ are the training LR features, L is the imposed sparsity, fixed to 3, and $\mathbf{w}^{(k)}$ are the decomposition coefficients over $\mathbf{D}_l$.

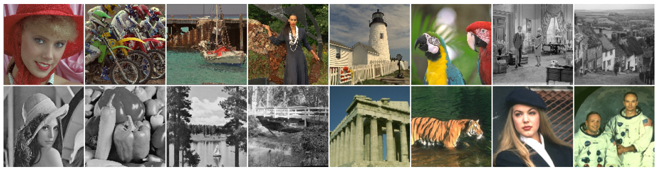

Fig. 2. Evaluation dataset with 16 images *aka* DB1 [12]. The images are numbered 1-8 on 1^{st} row and 9-16 on 2^{nd} row.

2.4. Anchored regressors

We train a linear regressor locally for each anchoring point by solving, as in [13]:

$$\min_{\boldsymbol{\beta}} \{\|\mathbf{y} - \mathbf{S}_{l,\mathbf{d_y}} \boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|^2\}, \tag{2}$$

where $\mathbf{y}$ is a LR patch feature whose nearest anchoring point is $\mathbf{d_y} \in \mathbf{D}_l$, $\mathbf{S}_{l,\mathbf{d_y}}$ are the N nearest neighbors in the training pool for $\mathbf{d_y}$, and λ is the regularization parameter, here fixed to 0.1. The regressor $\mathbf{P}_{\mathbf{d_y}}$ corresponding to the anchoring point $\mathbf{d_y}$ is computed offline:

$$\mathbf{P}_{\mathbf{d_y}} = \mathbf{S}_{h,\mathbf{d_y}} (\mathbf{S}_{l,\mathbf{d_y}}^T \mathbf{S}_{l,\mathbf{d_y}} + \lambda \mathbf{I})^{-1} \mathbf{S}_{l,\mathbf{d_y}}^T, \tag{3}$$

where $\mathbf{S}_{h,\mathbf{d_y}}$ contains the HR patches corresponding to the LR vectors in $\mathbf{S}_{l,\mathbf{d_y}}$.

2.5. Runtime

At test time, we first extract from the input compressed image the patch features densely over a grid. For each input LR feature $\mathbf{y}$ we retrieve the nearest neighboring anchoring point $\mathbf{d_y} \in \mathbf{D}_l$ and obtain the output $\mathbf{x}$ by applying the stored regressor $\mathbf{P}_{l,\mathbf{d_y}}$ at anchoring point $\mathbf{d_y}$:

$$\mathbf{x} = \mathbf{P}_{l,\mathbf{d_y}} \mathbf{y} \tag{4}$$

The regressed $\{\mathbf{x}\}$ patches are averaged in the overlapping areas to obtain the correcting output image. Finally, the input LR image is added to obtain the complete enhanced output HR image.

3. EXPERIMENTS

In this section we evaluate the performance of our proposed method. We show how its performance is influenced by the design parameters and compare to state-of-the-art methods on a standard dataset.

3.1. Benchmark

For a fair comparison with SLGP we use the same images for testing as Kwon *et al.* [12]. The dataset (see Fig. 2) contains 16 images familiar to the community (512×512 or 256×256 pixels). While Kwon *et al.* [12] uses 500 training images from a personal collection, we use the training set of 91 images proposed by Yang *et al.* [18] and extract 5 million patches from them by computing first image scale pyramids with downscaling factor 0.98 and bicubic interpolation.

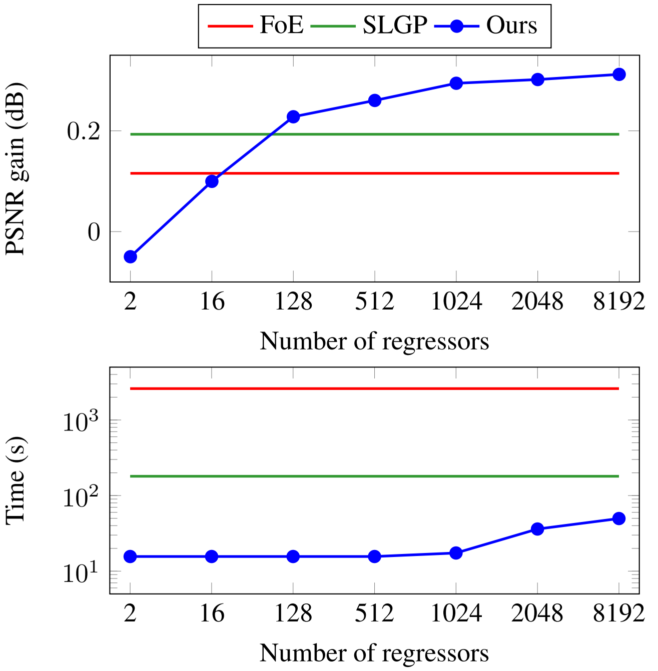

Fig. 3. Number of regressors vs. performance and running time.

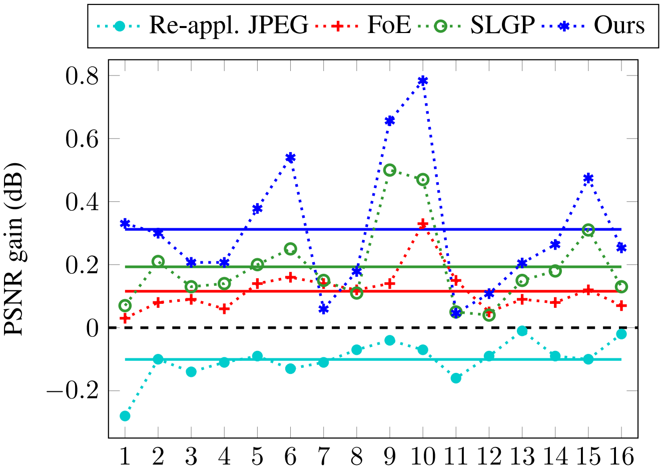

Fig. 4. PSNR gain comparison of the proposed method against re-application of JPEG 2000, FoE, and SLGP image enhancement algorithms. The x axis corresponds to the image index as in Fig. 2. The average PSNR gains across the dataset are marked with solid lines.

We compare our method with the re-application of JPEG 2000 method [11], Field of Experts (FoE) [4] and Semi-local Gaussian Processes (SLGP) [12] (state-of-the-art).

Each image was degraded using the JPEG 2000 encoder from the Kakadu software package [1] at 0.1 bits per pixel (BPP) at test time, a compression at which the artifacts are usually easily noticed. At training time the images are compressed at only 0.3 BPP. At this lower compression rate the regressors can more easily pick up the patterns of the artifacts which leads to an improvement in performance. The performance of the enhancement methods is measured by evaluating the peak signal-to-noise ratio (PSNR) to the uncompressed image. We report the PSNR gain relatively to the degraded image. Note that for our method we use the YCbCr color space for the color images (10 out of 16 images) and perform the enhancement only on the Y channel.

Our choice to work directly with JPEG 2000 is due to the increased difficulty in obtaining significant improvements from most current artifact reduction methods (often less than 0.1dB for 0.1 BPP). With respect to JPEG, JPEG 2000 is a superior compression algorithm, provides better quality for the same BPP. Also JPEG exhibits stronger artifacts, partly due to the BDCT stage (block artifacts), and it is easier to enhance (often over 0.5dB for 0.1 BPP).

3.2. Parameters

The default main parameters of our method are: 5 million training pairs of LR and HR patches, 7 × 7 pixels patch size, 2048 anchoring points / regressors, and 2048 nearest neighbors for the offline computation of each regressor.

For the A+ method [13] applied to super-resolution it was shown that increasing the number of training patches leads to increased in PSNR performance, and indeed our preliminary experiments confirmed the same behaviour for our method on the artifact reduction task.

For the patch size we considered 3×3, 5×5, 7×7, and 9×9 patch sizes. The performance improved up to 7×7, but slightly diminished for 9×9. Therefore, our choice of patch size (7×7) matches the one from the SLGP method [12]. Note that at 3×3 patches and 1024 regressors our method still gains 0.194dB, comparable to SLGP with 7×7 patches.

The number of linear regressors / anchoring points (dictionary size) is evaluated in Fig. 3 with respect to PSNR gain and average running time per image. There is a linear relation between the number of regressors and the running time, since a linear search is involved for picking up the nearest anchoring point and stored regressor for each input patch. The linearity holds above 512 regressors when the searching time dominates. Our method is order(s) of magnitude faster than the compared FoE and SLGP methods. With as few as 16 regressors our method reaches the PSNR gain of the FoE method and with 128 regressors clearly outperforms the SLGP method. Our method peaks at 0.312dB for 8192 regressors, but reaches a plateau at 1024 regressors (0.302dB). We expect that by increasing the size of the training set of images and its variance, as well as potentially the number of training patches, the performance of our method could be even more improved. It might at this point also be noted that in the current setting we only use 91 images, while SLGP uses 500 images. Our method is well behaved: more training data or more regressors usually results in better performance.

[1] http://www.kakadusoftware.com

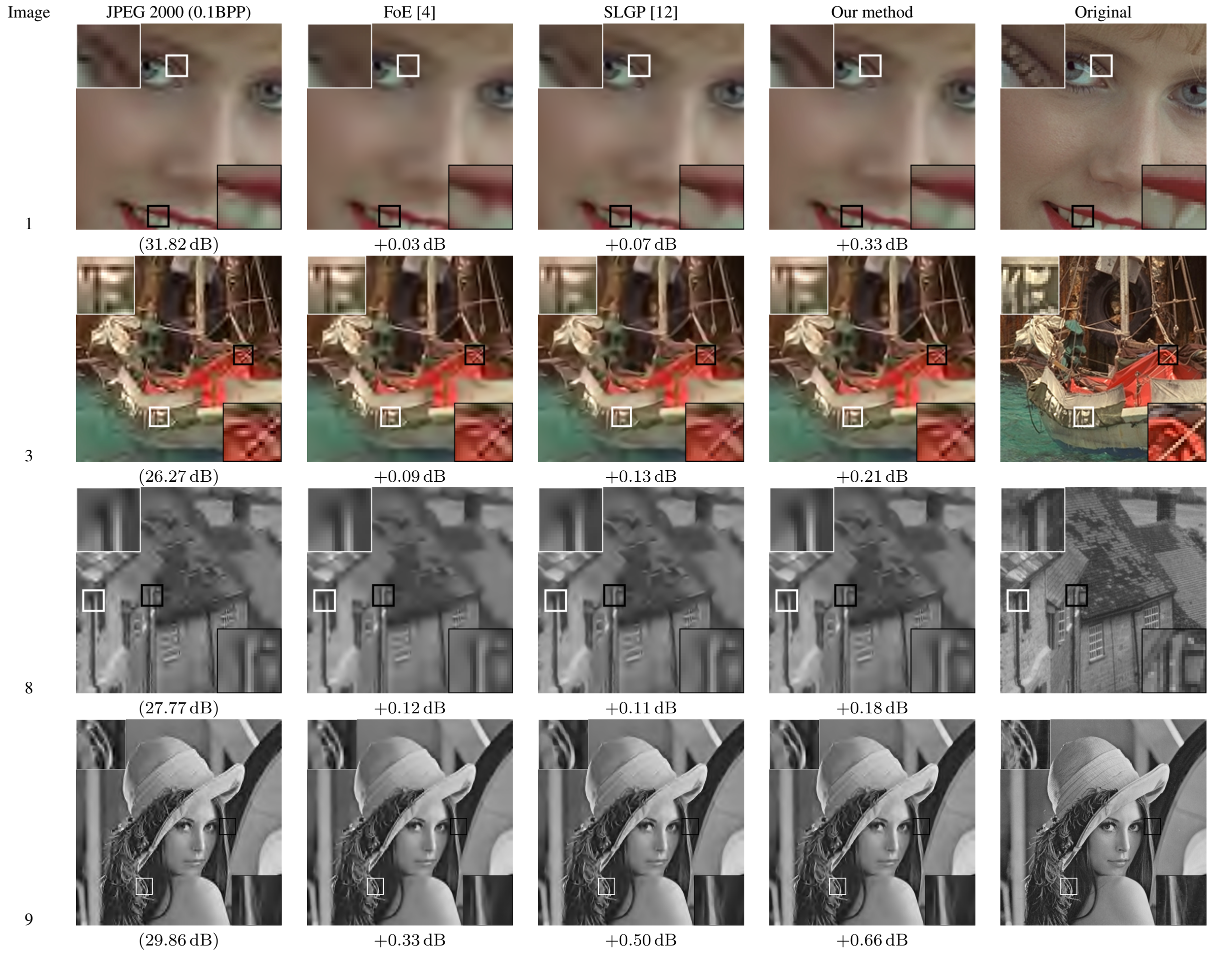

Fig. 5. Qualitative results for image 1, 3, 8, and 9 from the testing dataset (see Fig. 2). Best seen on screen.

3.3. Performance

In order to assess the performance of our method we build up our benchmark following the settings from Kwon *et al.* [12] as used for their SLGP state-of-the-art method. In Fig. 4 we compare the proposed method against re-application of JPEG 2000 method, FoE, and SLGP in terms of PSNR gain. We keep the same image indices from [12], as depicted in Fig. 2 and report also the average performance. Our method improves over SLGP for all the images, except image 7. Also in average performance we achieve a strong 0.312dB, significantly better than SLGP with 0.192dB and FoE with 0.115dB. The re-application of JPEG 2000 leads to negative gains.

The running time of our method compares favorable with the other top methods such as SLGP or FoE (see Fig. 3). If SLGP requires 180s and FoE $\sim$ 2600s per 512×512 pixels images, our method needs only 15s with 1024 regressors (Matlab). Our codes are publicly available at:
`http://www.vision.ee.ethz.ch/~timofter/`

For the qualitative performance assessment we compare enhancement results for 4 images in Fig. 5. We notice a clear improvement in quality between the JPEG 2000 input image and the result of SLGP or of our method. FoE tends to oversmooth the edges, while our method produces relatively sharp edges and remains closer to the uncompressed original image.

Our method improves $\sim$ 0.12dB over the SLGP method. While this might be a small improvement in absolute terms, it is a very solid result given the difficult scenario (JPEG 2000 @ 0.1BPP) we dealt with. In fact, looking at relative terms, our method with 0.31dB gain almost doubles the performance of SLGP (0.19dB), and triples FoE (0.11dB). Moreover, our method is orders of magnitude faster.

4. CONCLUSION

In this paper, we propose a novel and efficient artifact reduction algorithm based on A+. We embed prior information from the training images and the compressed outputs into a set of learned linear regressors. At test, after applying these we improve the compressed images by reducing the artifacts. The experiments show large improvements doubling the PSNR gain when compared to state-of-the-art methods such as SLGP [12], while being an order of magnitude faster.

Acknowledgements. This work was supported by the ERC Advanced Grant VarCity (#273940), Climate KIC (#2-71298-15), and KTI-SUPSI (#2-69650-14) projects.

5. REFERENCES

[1] Andrey V. Nasonov and Andrey S. Krylov, “Scale-space method of image ringing estimation,” in *ICIP*, 2009.

[2] Yongyi Yang, N.P. Galatsanos, and A.K. Katsaggelos, “Projection-based spatially adaptive reconstruction of block-transform compressed images,” *IP*, vol. 4, no. 7, pp. 896–908, 1995.

[3] Xin Li, “Improved wavelet decoding via set theoretic estimation,” *CSVT*, vol. 15, no. 1, pp. 108–112, 2005.

[4] Stefan Roth and Michael J. Black, “Fields of experts,” *IJCV*, vol. 82, no. 2, pp. 205–229.

[5] Valero Laparra, Juan Gutiérrez, Gustavo Camps-Valls, and Jesús Malo, “Image denoising with kernels based on natural image relations,” *JMLR*, vol. 11, pp. 873–903, 2010.

[6] D. Tschumperle and R. Deriche, “Vector-valued image regularization with pdes: a common framework for different applications,” *PAMI*, vol. 27, no. 4, pp. 506–517, 2005.

[7] Guoping Qiu, “Mlp for adaptive postprocessing block-coded images,” *CSVT*, vol. 10, no. 8, pp. 1450–1454, 2000.

[8] A. Foi, V. Katkovnik, and K. Egiazarian, “Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images,” *IP*, vol. 16, no. 5, pp. 1395–1411, 2007.

[9] Guangtao Zhai, Weisi Lin, Jianfei Cai, Xiaokang Yang, and Wenjun Zhang, “Short communication: Efficient quadtree based block-shift filtering for deblocking and deringing,” *J. Vis. Comun. Image Represent.*, vol. 20, no. 8, pp. 595–607, 2009.

[10] Aria Nosratinia, “Enhancement of jpeg-compressed images by re-application of jpeg,” *VLSISP*, vol. 27, no. 1-2, pp. 69–79, 2001.

[11] A. Nosratinia, “Postprocessing of jpeg-2000 images to remove compression artifacts,” *SPL*, vol. 10, no. 10, pp. 296–299, 2003.

[12] Younghee Kwon, Kwang In Kim, James Tompkin, Jin Hyung Kim, and Christian Theobalt, “Efficient learning of image super-resolution and compression artifact removal with semi-local gaussian processes,” *PAMI*, 2014.

[13] Radu Timofte, Vincent De Smet, and Luc Van Gool, “A+: Adjusted anchored neighborhood regression for fast super-resolution,” in *ACCV*, 2014.

[14] Jiqing Wu, Radu Timofte, and Luc Van Gool, “Efficient regression priors for post-processing demosaiced images,” in *ICIP*, 2015.

[15] Roman Zeyde, Michael Elad, and Matan Protter, “On single image scale-up using sparse-representations,” in *Curves and Surfaces*, pp. 711–730. Springer, 2012.

[16] Radu Timofte, Vincent De Smet, and Luc Van Gool, “Anchored neighborhood regression for fast example-based super-resolution,” in *ICCV*, 2013.

[17] Michal Aharon, Michael Elad, and Alfred Bruckstein, “K-svd: An algorithm for designing overcomplete dictionaries for sparse representation,” *SP*, vol. 54, no. 11, pp. 4311–4322, 2006.

[18] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma, “Image super-resolution as sparse representation of raw image patches,” in *CVPR*. IEEE, 2008, pp. 1–8.