Image Denoising with Sparse Representation

Pengyu Zhang
pengyu.zhang@ufl.edu
Department of Electrical & Computer & Engineering
University of Florida

Yingwen Huang
yingwenhuang@ufl.edu
Department of Computer & Information Science & Engineering
University of Florida

Abstract

This paper presents our research and our understanding in image denoising, and focuses on the sparse representation to solve problems in image denoising. It also shows our motivation of image denoising and our interested problem statements about this field, the algorithms to solve this problems and the performance on data during our experiment. In addition, it is written to summarise some mathematical formulations and processes, algorithms, experiments to the corresponding solution. It shows our conclusion from the experiments results at the end of this paper.

1 Introduction

1.1 Background

Image processing has been a popular topic in machine learning and computer vision field. It is a method to perform some operations on images to extract useful information, like enhanced images or some important characteristics. In general, there are several steps to perform during the image process: importing the images, manipulating the images, analysing and reporting the results. During these steps, an accurate image plays a mandatory role since it will accelerate the analysing process and improve the result.

The researchers desire to have "pure" images for analysis purpose. But this becomes harder to achieve. First of all, due to the highly developed technologies on the cameras, the amount of images taken every day is exploding. It is a challenge to filter the huge amount of images. And it is difficult to have the accurate information or characteristics from the images because the modern digital cameras can catch more information from the environment, and the images become "noising", which indicates that the images obtain some unnecessary effects.

The "noise" images can be produced by different extrinsic (like the influence of environment) or intrinsic (like the camera sensor) conditions. Most of the times these conditions are not possible to avoid. And these conditions lead to image distortion or loss of image information, where the researchers always lose most details from the noised images. Hence, there are more demands on recovering these image details by removing the unnecessary noise to obtain the clearer images. This process is called image denoising, or image enhancement.

Image denoising is one of the fundamental challenges in the computer vision field, including the image pre-processing and post-processing. And the purpose of image denoising is to obtain the original image content for better performance. Researchers have been studies for decades and developed a

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

wide range of image denoising applications, such as image segmentation, image classification, image restoration, and so on. And many algorithms are proposed for image denoising.

1.2 Motivation

Although people have been studied for a long time and developed many useful algorithms, image denoising still remains an open task, especially when the images are taken under the poor conditions and the noise level of the images is very high. Based on the mathematical perspective, image denoising is an inverse problem, and most of the time there is no solution to solve this problem in practical, where indicates that there are no unique solutions among these problems. In order to solve these insoluble problems, the best approach is to conduct several experiments among different methods and to compare the results to find the one with better performance.

There are some interesting methods of image denoising as following [1]:

- Classical denoising method, including Spatial domain filtering and Variantional denoising (total variation regulation, non-local regularization, sparse representation, and low-rank minimization).
- Transform techniques in image denoising, including transform domain filtering methods (data adaptive and non-data adaptive transform) and BM3D.
- CNN-bases denoising methods, including MLP models and deep learning-based denoising methods.

This project will focus on the sparse representation methods which is very powerful to solve image denoising problems in the linear perspective representation. At the mean time, the paper will explore the sparse representation methods. Some implementations will be conducted to perform the comparisons to show the performance of these method in image enhancement.

2 Problem Statement

2.1 l_0-norm Sparse Representation

Sparse representations can be classified as several categories based on the norm regularization terms, such as l_p-norm ($0 < p < 1$) minimization, l_1-norm minimization, $l_{2,1}$-norm minimization, l_2-norm minimization, and etc. And this paper will focus on the one with l_0-norm minimization.

Let the data matrix $\mathbf{Z} \in \mathcal{R}^{n \times k}, n < k$, where $\mathbf{Z} = [\mathbf{z_1}\ \mathbf{z_2}\ \dots\ \mathbf{z_k}]$, and $\mathbf{z_1}, \mathbf{z_2}, \dots, \mathbf{z_k} \in \mathcal{R}^n$ are the column vectors of $\mathbf{Z}$. Assume the outcome is $\mathbf{y} \in \mathcal{R}^n$, and our problem is to find α such that:

$$\mathbf{y} = \mathbf{Z}\alpha = \alpha_1 \mathbf{z_1} + \alpha_2 \mathbf{z_2} + \dots + \alpha_k \mathbf{z_k} \tag{1}$$

where $\alpha = [\alpha_1\ \alpha_2\ \dots\ \alpha_k]^T$ and $\alpha_1, \alpha_2, \dots, \alpha_k \in \mathcal{R}$. At this point, the main problem is to solve this linear system. However, this linear system is underdetermined since there is not any prior knowledge or constraints on how to find the solution α, and there is no unique solution α.

To solve this task, a proper regularizer constraint is need on solution α. And the sparse representation with l_0-norm minimization can be a appropriate method in this case. The linear system (1) can be converted to the following optimization problem:

$$\begin{gathered} \hat{\alpha} = arg\min \|\alpha\|_0 \\ subject\ to\ \mathbf{y} = \mathbf{Z}\alpha \end{gathered} \tag{2}$$

where $\|\cdot\|_0$ is the l_0-norm referring to the amount of the non-zero elements. Since $k > n$, this optimization problem (2) is also equivalent to the following optimization problem:

$$\begin{gathered} \mathbf{y} = \mathbf{Z}\alpha \\ subject\ to\ \|\alpha\|_0 \leq n \end{gathered} \tag{3}$$

Most of the time, the data is not "pure" and it contains unavoidable noises in practice, especially during the image processing. Hence, the original linear system (1) should be

$$\mathbf{y} = \mathbf{Z}\alpha + \mathbf{s} \tag{4}$$

where $\mathbf{s} \in \mathcal{R}^n$ is noise. Suppose $\mathbf{s}$ is bounded by ϵ, $\|\mathbf{s}\|_2 \leq \epsilon$. In this case, the optimization problem (4) is updated as following:

$$\hat{\alpha} = argmin\|\alpha\|_0 \quad (5)$$
$$subject\ to\ \|\mathbf{y} - \mathbf{Z}\alpha\|_2^2 \leq \epsilon$$

or

$$\hat{\alpha} = argmin\|\mathbf{y} - \mathbf{Z}\alpha\|_2^2 \quad (6)$$
$$subject\ to\ \|\alpha\|_0 \leq \epsilon$$

Assume $\mathbf{\Lambda}$ be the Lagrange multiplier constant, then the equivalent Lagrange problem is following:

$$\hat{\alpha} = \mathcal{L}(\alpha, \mathbf{\Lambda}) = argmin\|\mathbf{y} - \mathbf{Z}\alpha\|_2^2 + \mathbf{\Lambda}\|\alpha\|_0 \quad (7)$$

One popular way to seek for the solution is Greedy algorithm. l_0-norm sparsity representation is considered as an NP-hard problem, and Greedy strategy provides an approximation solution to the l_0 sparsity representation problem, although it cannot directly solve this type of optimization problems. By searching the best local optimal approximation at every iteration, the greedy method only concentrates on finding the most M related components[3], where M represents the value of sparsity.

2.2 Signal Processing Differences between Images and Non-images

The problem statements above is generally used to solve for signal and noise problems from non-image situation. Images are different from regular signals since they own local information in each patch. Due to such properties, it's not wise to take one image as one sample. As results, the question arises: Can we still utilize a global sparse model to denoise images since sparsity representations are indeed limited in small block of images?[6]. To illustrate this problem, the model based on small patches of image and its global extension are going to be discussed.

2.2.1 Local Sparse Model

Assume the image with patches of size $\sqrt{n} \times \sqrt{n}$ pixels, and a dictionary matrix $\mathbf{D} \in \mathcal{R}^{n \times k}$ is defined. Suppose $\mathbf{D}$ is fixed and known, and $\mathbf{x}$ is the original $\mathbf{y}$ without any noise, then (2) is updated as following:

$$\hat{\alpha} = arg\min\|\alpha\|_0 \quad (8)$$
$$subject\ to\ \mathbf{x} \approx \mathbf{D}\alpha$$

Suppose $\mathbf{x}$ belongs to $(\epsilon, L, \mathbf{D})$, which is called Sparse-land signals, and L is defined to indicate the depth of the required sparsity with the form $\|\hat{\mathbf{a}}\|_0 \leq L \ll n$. And $\mathbf{y}$ is contaminated by the noisy version of this Sparse-land signal, then the problem for denoising is to solve

$$\hat{\alpha} = arg\min\|\alpha\|_0 \quad (9)$$
$$subject\ to\ \|\mathbf{D}\alpha - \mathbf{y}\|_2^2 \leq T$$

where T is dictated by ϵ and σ. The required denoised image is given as $\hat{\mathbf{x}} = \mathbf{D}\hat{\alpha}$. Further more, to set the constraint to be a penalty, a proper μ is chosen such that an equivalent optimization problem to (9) is shown as following:

$$\hat{\alpha} = arg\min\|\mathbf{D}\alpha - \mathbf{y}\|_2^2 + \mu\|\alpha\|_0 \quad (10)$$

2.2.2 Extension from Local to Global

If the size of image is $\sqrt{N} \times \sqrt{N}$, where $N \gg n$, the most natural idea is to create a much larger dictionary $\mathbf{D}$. However, such a small patches are still powerful since these small patches can simplify the overall image calculation by implying a locality of the resulting algorithms. One possible approach is to complete the calculation on the patches of size $\sqrt{n} \times \sqrt{n}$ and then tile the result, like averaging the results. More discussions will be shown in the algorithm part.

Assume $\mathbf{Y}$ is the measured image, and $\mathbf{X}$ is the denoised version of $\mathbf{Y}$. Suppose every patch in $\mathbf{X}$ belongs to $(\epsilon, L, \mathbf{D})$, then (10) can be replaced by:

$$\{\hat{a}_{ij}, \hat{\mathbf{X}}\} = arg\min_{a_{ij}, \mathbf{X}} \lambda\|\mathbf{X} - \mathbf{Y}\|_2^2 + \sum_{i,j}\mu_{ij}\|\alpha_{ij}\|_0 + \sum_{i,j}\|\mathbf{D}\alpha_{ij} - \mathbf{R}_{ij}\mathbf{X}\|_2^2 \quad (11)$$

where μ_{ij} is location dependent to make sure the constraints in the form of $\|\mathbf{D}\alpha_{\mathbf{ij}} - \mathbf{x}_{ij}\|_2^2 \leq T$, and $\mathbf{R}_{ij} \in \mathcal{R}^{n\times N}$ is the (ij) blocks extracted from the image, and every patch $\mathbf{x}_{ij} = \mathbf{R}_{ij}\mathbf{X}$. In (11), the first term is the log-likelihood global force to demand the proximity between $\mathbf{Y}$ and $\mathbf{X}$. Also, $\|\mathbf{X} - \mathbf{Y}\|_2^2 \leq Const \cdot \sigma^2$ shows the direct relationship between λ and σ. And the second and the third terms are to make sure the constructed image $\mathbf{X}$ and every patch $\mathbf{x}_{ij}$ has a sparse representation bounded by error. Notice that the summation over i, j includes $(\sqrt{N} - \sqrt{n} + 1)^2$ terms, and considers all image patches in $\mathbf{X}$ with overlaps.

3 Algorithm

In this section, the basic idea of the relevant greedy algorithms is demonstrated in the sparse representation, including matching pursuit (MP), orthogonal matching pursuit (OMP) as well as k-means singular value decomposition (K-SVD). The latter algorithm is either an improvement or a more comprehensive model based on the former one, thus to reveal the evolution of greedy strategies on searching the sparse solution. The well-designed K-SVD and one of its extension are further utilized as the theory supports in our image denoising assignment.

3.1 Matching Pursuit (MP)

The MP is the earliest greedy strategy that iteratively searches the most similar atoms from a fixed dictionary to approximate the sparse representation of the original signal. One of the examples of MP that decomposes the original signal with an over-complete dictionary is as follows:

Assuming that the preliminary residual $\mathbf{R}_0 = \mathbf{y}$, the fixed dictionary $\mathbf{D} \in \mathcal{R}^{n\times k}$ and each column of the dictionary are normalized by l_2-norm. The MP first measures the similarity between the signal $\mathbf{y}$ and each column (atom) by using the inner product that satisfies:

$$\begin{aligned} \mathbf{D} &= [\mathbf{d}_0, \mathbf{d}_1, \cdots, \mathbf{d}_k] \\ |\left\langle \mathbf{R}_0, \mathbf{d}_{\lambda_0} \right\rangle| &= \text{sup}|\left\langle \mathbf{R}_0, \mathbf{d}_i \right\rangle| \end{aligned} \tag{12}$$

where λ_0 denotes the index. Therefore, the signal $\mathbf{y}$ can be represented by the sum of new residuals and the sparse representation:

$$\mathbf{y} = \mathbf{R}_1 + \left\langle \mathbf{y}, \mathbf{d}_{\lambda_0} \right\rangle \mathbf{d}_{\lambda_0} \tag{13}$$

Thus, the residual $\mathbf{R}_1$ is orthogonal to the selected vector $\mathbf{d}_{\lambda_0}$. It can be decomposed again until the sparsity holds. For the i^{th} iteration, the most similar atom is $\mathbf{d}_{\lambda_i}$ and the approximation is obtained from the following equation:

$$\mathbf{R}_i = \left\langle \mathbf{R}_i, \mathbf{d}_{\lambda_i} \right\rangle \mathbf{d}_{\lambda_i} + \mathbf{R}_{i+1} \tag{14}$$

If the residual at iteration i is less than the tolerance we set up, the signal $\mathbf{y}$ can be approximately reconstructed by:

$$\mathbf{y} = \sum_{1}^{i-1} \left\langle \mathbf{R}_n, \mathbf{d}_{\lambda_n} \right\rangle \mathbf{d}_{\lambda_n} + \mathbf{R}_i \tag{15}$$

The Algorithm 1 summarizes the MP procedure.

Algorithm 1: Matching Pursuit

Input: Sample $\mathbf{y}$, measurement matrix $\mathbf{X}$, sparse coefficient α
Initialization: $\mathbf{D}_0$=ϕ, t=1, residual $\text{R}_0 = \mathbf{y}$;
$\mathbf{\Lambda}_0 = \phi$, tolerance τ is a small constant

While $||\mathbf{R}_t||_1 > \tau$ **do**

1. Find the largest value of inner product between $\mathbf{R}_{t-1}$ and $\mathbf{x}_j$ ($j \notin \mathbf{\Lambda}_{t-1}$) by computing $\lambda_t = \arg\max_{j\notin\Lambda_{t-1}} |\left\langle \mathbf{R}_{t-1}, \mathbf{x}_j \right\rangle|$. The value is denoted as v_t.
2. Update the coefficient $\hat{\alpha}_{\lambda_t} = v_t$.
3. Update index $\mathbf{\Lambda}_t = \mathbf{\Lambda}_t \bigcup \lambda_t$ and the dictionary $\mathbf{D}_t = [\mathbf{D}_{t-1}, \mathbf{x}_{\lambda_t}]$.
4. Update residual by using $\mathbf{R}_t = \mathbf{y} - \mathbf{D}_t\hat{\alpha}$
5. $t = t + 1$

End Output: $\mathbf{D}$, α

3.2 Orthogonal Matching Pursuit (OMP)

While MP is useful, it has a disadvantage that the residuals can only be guaranteed being orthogonal to the most recently selected atom. This property could lead to a solution that atoms are not independent of each other, which means there still exists redundancy and the algorithm does not converge. Therefore, as one of the improvements of MP, the orthogonal matching pursuit (OMP)[4] is proposed to make the matching pursuit process more efficient by forcing every projection being orthogonal to the space that is constructed by the chosen atoms before the current iteration. With the orthogonality process, atoms are truly independent of each other and the algorithm is guaranteed to converge after limited iterations. The details are shown in Algorithm 2.

Algorithm 2: Orthogonal Matching Pursuit

Input: Sample $\mathbf{y}$, measurement matrix $\mathbf{X}$, sparse coefficient α
Initialization: $\mathbf{D}_0$=ϕ, t=1, residual $\mathrm{R}_0 = \mathbf{y}$,;
$index$;
$\Lambda_0 = \phi$, tolerance τ is a small constant

While $||\mathbf{R}_t||_1 > \tau$ **do**

1. Find the largest value of inner product between $\mathbf{R}_{t-1}$ and $\mathbf{x}_j$ ($j \notin \mathbf{\Lambda}_{t-1}$) by calculating $\lambda_t = \arg\max_{j \notin \Lambda_{t-1}} | \langle \mathbf{R}_{t-1}, \mathbf{x}_j \rangle |$.
2. Update index $\mathbf{\Lambda}_t = \mathbf{\Lambda}_t \bigcup \lambda_t$ and the dictionary $\mathbf{D}_t = [\mathbf{D}_{t-1}, \mathbf{x}_{\lambda_t}]$.
3. Compute the sparse coefficient by computing $\hat{\alpha} = \arg\min ||\mathbf{y} - \mathbf{D}_t\hat{\alpha}||_2^2$
4. Update residual by using $\mathbf{R}_t = \mathbf{y} - \mathbf{D}_t\hat{\alpha}$
5. $t = t + 1$

End Output: $\mathbf{D}$, α

The difference exists in how the sparse coefficients are updated. In MP, the coefficients are directly updated by the largest value of inner product since the atoms are unit vectors. In OMP, instead, the coefficients are obtained by finding the projection of the current atom onto the former space created by all of the former atoms.

3.3 K-means Singular Value Decomposition (K-SVD)

The representative K-SVD[6] algorithm generalizes the K-means clustering process, updates the coding book and the sparse coefficients in an alternative manner. Due to its flexibility in matching with many pursuit algorithms(e.g., OMP), it has been widely used as an unsupervised learning approach in image recovery and denoising[5]. The motivation of K-SVD arises from vector quantization problems. To summarize, vector quantization uses K codewords to represent a sequence of signals by using K-means clustering algorithm under l_2-norm.

Instead of only utilizing the closest atom from the dictionary to represent the signal, K-SVD extrapolates the extreme situation (the closest atom) to the linear combination of several atoms and the number of atoms is so-called sparsity [7]. Recall what has been discussed above, the objective function of sparse representation is

$$\hat{\alpha} = argmin\|\mathbf{y} - \mathbf{D}\alpha\|_2^2 \qquad (16)$$
$$subject\ to\ \|\alpha\|_0 \leq \epsilon$$

where the dictionary $\mathbf{D}$ is always assumed fixed. However, not only the sparse coefficients but the dictionary are updated in an alternating manner in K-SVD: First, the coefficients are searched through matching pursuit, then the dictionary is updated based on the old coefficients based on singular value decomposition (SVD) and new coefficients are updated during the dictionary update, too. Such progress makes K-SVD become a more general extension K-means.

Now considering implementing K-SVD on a signal matrix where each signal is stored in column-wise, the first step (sparse coding) can be rewritten as follows:

$$\hat{\alpha} = argmin\|\mathbf{Y} - \mathbf{DA}\|_2^2 \qquad (17)$$
$$subject\ to\ \|\mathbf{A}_i\|_0 \leq \epsilon$$

where the minimization term can be decomposed to k individual problems that have the form:

$$\|\mathbf{Y} - \mathbf{DA}\|_2^2 = \sum_{i=1}^{k} \|\mathbf{y}_i - \mathbf{Da}_i\|_2^2 \tag{18}$$

$$subject\ to\ \|\mathbf{a}_i\|_0 \leq \epsilon, for\ i = 1, 2, \ldots, k.$$

To the knowledge of MP and OMP, the sparse coding process is feasible. The following process is to update the dictionary and none-zero coefficients simultaneously. Suppose that the $\mathbf{D}$ and $\mathbf{A}$ are fixed when the p^{th} column of $\mathbf{D}$, say d_p, and p^{th} row of $\mathbf{A}$, say $\mathbf{A}_p^T$ are going to be updated, the function (18) is thus:

$$\begin{aligned} \|\mathbf{Y} - \mathbf{DA}\|_2^2 &= \|\mathbf{Y} - \sum_{j=1}^{k} \mathbf{d}_j \mathbf{a}_j^T\|_2^2 \\ &= \|(\mathbf{Y} - \sum_{j \neq p} \mathbf{d}_j \mathbf{a}_j^T) - \mathbf{d}_p \mathbf{a}_p^T\|_2^2 \\ &= \|\mathbf{E}_p - \mathbf{d}_p \mathbf{a}_p^T\|_2^2 \end{aligned} \tag{19}$$

where $\mathbf{d}_p$ and $\mathbf{a}_p^T$ can be obtained by decomposing $\mathbf{E}_p$. SVD is proposed here since it has the ability to minimize the error. The Algorithm 3 involves the details.

Algorithm 3: K-means singular value decomposition

Input: Sample $\mathbf{Y}$, dictionary $\mathbf{D}$ with l_2 normalized columns
Initialization: t = 1

Until Convergence:

Sparse coding:

1. Use matching pursuit algorithms to compute the sparse coefficients by solving function (18)

Dictionary update: For each column $p = 1, 2, \ldots, K$ in $\mathbf{D}^{t-1}$

1. Define the family of samples from $\mathbf{Y}$ that use this atom, $w_p = \left\{ i | 1 \leq i \leq N, \mathbf{a}_p^T(i) \neq 0 \right\}$
2. Compute the error matrix $\mathbf{E}_p$ by performing (19)
3. Only choosing certain columns from $\mathbf{E}_p$ according to w_p to avoid updating zero coefficients in $\mathbf{a}_p^T$, and obtaine the filtered $\mathbf{E}_p^F$
4. Decompose $\mathbf{E}_p^F$ by using SVD: $\mathbf{E}_p^F = \mathbf{U\Delta V}^T$. Update the coefficient vector $\mathbf{a}_p^T$ with the multiplication of $\mathbf{\Delta}(1, 1)$ and the first row of $\mathbf{V}^T$. Update the dictionary column $\mathbf{d}_p$ with the first column of $\mathbf{U}$.
5. t=t+1

End Output: Dictionary $\mathbf{D}$, and coefficient matrix $\mathbf{A}$

3.4 Extension of K-SVD on Image Denoising

In this subsection, the notations are consistent with what are discussed in section 2.2. To make K-SVD feasible in image denosing assignments based on small patches, the general objective function is the same as (11):

$$\{\hat{a}_{ij}, \hat{\mathbf{X}}\} = arg \min_{a_{ij}, \mathbf{X}} \lambda \|\mathbf{X} - \mathbf{Y}\|_2^2 + \sum_{ij} \mu_{ij} \|\alpha_{ij}\|_0 + \sum_{ij} \|\mathbf{D}\alpha_{ij} - \mathbf{R}_{ij}\mathbf{X}\|_2^2 \tag{20}$$

It is clear to see that if the dictionary $\mathbf{D}$ is known and fixed, the sparsity measurement $\hat{\alpha}_{ij}$ of each image patch can be easily searched by matching pursuit. Given all $\hat{\alpha}_{ij}$, the reconstruction of image $\mathbf{X}$ is solved by (20) again:

$$\hat{\mathbf{X}} = arg \min_{a_{ij}, \mathbf{X}} \lambda \|\mathbf{X} - \mathbf{Y}\|_2^2 + \sum_{ij} \|\mathbf{D}\alpha_{ij} - \mathbf{R}_{ij}\mathbf{X}\|_2^2 \tag{21}$$

It is nothing but a quadratic problem with respect to variable $\mathbf{X}$. Thus the image after being denoised is computed by:

$$\hat{\mathbf{X}} = (\lambda\mathbf{I} + \sum_{ij}\mathbf{R}_{ij}^T\mathbf{R}_{ij})^{-1}(\lambda\mathbf{Y} + \sum_{ij}\mathbf{R}_{ij}^T\mathbf{D}\hat{\alpha}_{ij}) \tag{22}$$

In a nutshell, the K-SVD is applied to update the local dictionary as well as obtain the sparse representation of each small block of the entire image by scanning through all blocks. When the dictionary is well-trained, the final step is to recover the image $\mathbf{X}$ from contamination. Algorithm 4 displays the combination of K-SVD and quadratic minimization step.

Algorithm 4: Denoising a image

Input: The image $\mathbf{Y}$, $\mathbf{D} =$ overcomplete DCT dictionary with l_2 normalized columns
Initialization: t = 1, n-block size, k-dictionary size, λ-Lagrange multiplier, C-noise gain

Until Convergence:

Sparse coding:

1. Use matching pursuit algorithms to compute the sparse coefficients α_{ij} of each block $\mathbf{R}_{ij}$ by solving function

$$\forall \min_{\alpha_{ij}} \|\alpha_{ij}\| \; s.t. \; \|\mathbf{D^{t-1}}\alpha_{ij} - \mathbf{R}_{ij}\mathbf{X}\|_2^2 \leq (C\sigma)^2$$

Dictionary update: For each column $p = 1, 2, \ldots, K$ in $\mathbf{D}^{t-1}$

1. Define the family of blocks from $\mathbf{Y}$ that use this atom, $w_p = \{(i,j)|\mathbf{a}_{ij}(p) \neq 0\}$
2. For each index $(i,j) \in w_p$, compute the error by

$$\mathbf{E}_{ij}^p = \mathbf{R}_{ij}\mathbf{X} - \sum_{m \neq p}\mathbf{d}_m\alpha_{ij}(m)$$

3. Only choosing certain columns from $\mathbf{E}_p$ according to w_p to avoid updating zero coefficients, and obtaine the filtered $\mathbf{E}_p^F$
4. Decompose $\mathbf{E}_p^F$ by using SVD: $\mathbf{E}_p^F = \mathbf{U}\mathbf{\Delta}\mathbf{V}^T$. Update the coefficients $\{\alpha_{ij}(p)\}_{(i,j)\in w_p}$ with the multiplication of $\mathbf{\Delta}(1,1)$ and the first row of $\mathbf{V}^T$. Update the dictionary column $\mathbf{d}_p$ with the first column of $\mathbf{U}$.
5. t=t+1

End

Image reconstruction: Perform equation (22)
Output: Dictionary $\mathbf{D}$, $\mathbf{X}$

4 Experiments

In this section, K-SVD with OMP algorithm is implemented, and K-SVD with MP algorithm is not designed in this project. As discussed before, OMP is a natural extension of MP with convergence guarantee. The performance results are measured by peak signal-to-noise ratio (PSNR), which is based on the logarithmic decibel scale. PSNR shows the ratio between the maximum possible power of a signal and the power of corrupting noise. During the experiment, the maximum amount of iterations of K-SVD is set as 200. The size of each patch is 8 $\times$ 8. The value constant $Const$ is 1.15. The overcomplete dictionary $\mathbf{D} \in \mathcal{R}^{64\times256}$ is initialized with discrete cosine transform (DCT).

4.1 Performance on Different Gaussian Noise

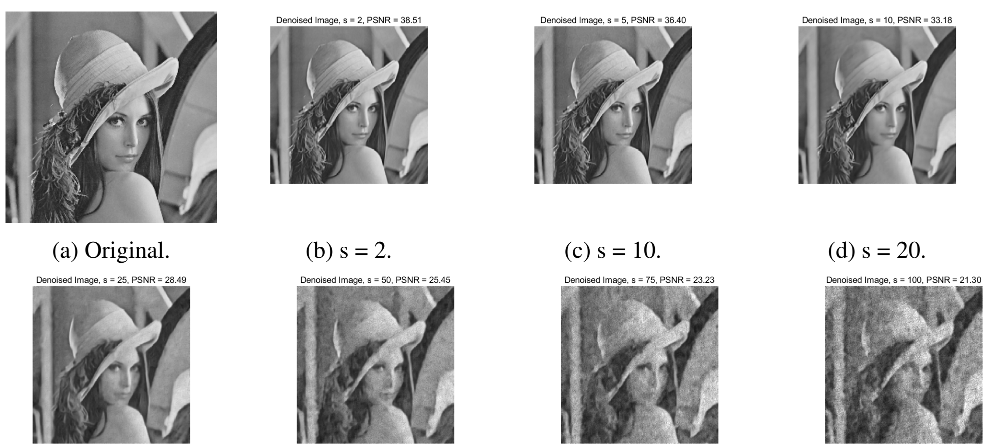
This part is designed to show how well K-SVD with OMP reconstructs the images from different Gaussian noise with different value of $\sigma^2 = s$. The noisy images are shown in Figure 1. It is obvious that the images get more contaminated with the increasing value of s. The denoised images are presented in Figure 2, and it is harder to have a better reconstructed images while s is increasing.

The trade line of PSNR is shown as Figure 3. In the Figure 3(a), it is clear that the values of PSNR of the denoised images are greater than the ones of the noisy images besides the one when $s = 2$. It might because the tolerance caused by the reconstruction is smaller than the added noise with small s. And Figure 3(b) shows that the differences of PSNR between noisy and denoised images is larger when s is larger. However, the speed of increasing the differences is slowing down. It states that more added noises becomes challenging to reconstruct the noisy images.

(a) Original. (b) s = 2. (c) s = 10. (d) s = 20.

(e) s = 25. (f) s = 50. (g) s = 75. (h) s = 100.

Figure 1: Noisy Images with Different Values of s



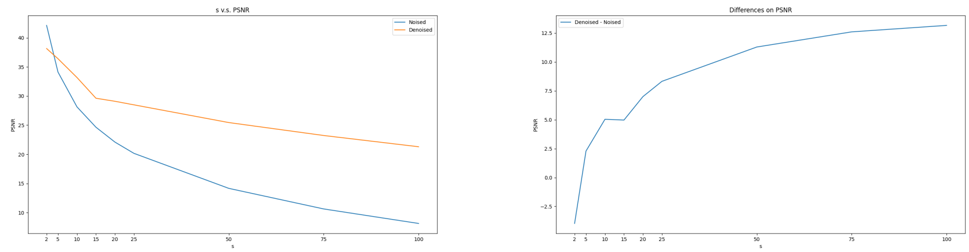
(a) Original. (b) s = 2. (c) s = 10. (d) s = 20.

(e) s = 25. (f) s = 50. (g) s = 75. (h) s = 100.

Figure 2: Denoised Images with Different Values of s

4.2 Performance on Different Sparsity

This section is designed to show the influence of sparsity, which is a parameter of the amount of non-zero elements in α. And the results obtained by setting 20 as the value of s is shown in Figure 4. PSNR remains in the range of $[28.80, 29.20]$ with the increasing sparsity, and the trading line is close to horizontal line. This performance result indicates the sparsity does not have impact on the performance of reconstructing the denoised images.



(a) PSNR.

(b) PSNR Differences.

Figure 3: s v.s. PSNR

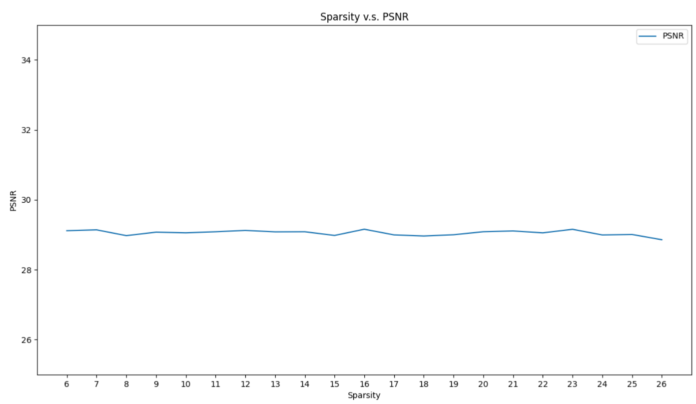


Figure 4: Sparity v.s PSNR

5 Conclusion

This work has presented some basic idea of image denosing field and some interesting problems in this fields. And it also has discussed several methods to solve the problems involving the image denoising with sparse representation. Image denoising is very essential either in research and in practical, and people have been studying for decades for the efficient algorithms. K-SVD with OMP implemented within this project to help us to understand the concepts in this specific filed. In the proposed method, the content of the dictionary has shown its importance during the denoising processing. Based on the experiment results, K-SVD with OMP works well but its performance is greatly effected by the levels of noise. And this work shows that the performance of image denoising with sparse representation is not dramatically improved by increasing the value of sparsity. Hence, smaller sparsity can be chosen to simplify the calculation and such property also demonstrates the elegance of K-SVD by keeping significant information while filtering noises.

During the experiment, it is clear that these algorithms are efficient with low scale images, and the runtime can be increased if the size of the image increases. Hence, it remains as a question whether any techniques will solve these problems as effectively as the small problems. There are several research directions to be considered to improve the performance on larger scale images, such as optimizing the parameters, using multiples dictionaries, replace MP or OMP by a faster pursuit technique, and etc.

References

The following are some papers we will read and use for our project. [1][2] summarize the state-of-art problems and methods in image denoising. [3] provides a comprehensive introduction of sparse representation and related algorithms, which will be helpful for us to digest and apply.[4][5][7]

describe the algorithms and thoughts that we will concentrate on. [6] provides the specific application in image denosing via sparse representation.

[1] Fan, L., Zhang, F., Fan, H. et al. Brief review of image denoising techniques. Vis. Comput. Ind. Biomed. Art 2, 7 (2019).

[2] Goyal B., Dogra A., Agrawal S., Sohi B.S., Sharma A.Image denoising review: From classical to state-of-the-art approaches Information Fusion, 55 (2020), pp. 220-244

[3] Z. Zhang, Y. Xu, J. Yang, X. Li and D. Zhang, "A Survey of Sparse Representation: Algorithms and Applications," in IEEE Access, vol. 3, pp. 490-530, 2015, doi: 10.1109/ACCESS.2015.2430359.

[4] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” in Proc. 27th Asilomar Conf. Signals, Syst., Comput., Nov. 1993, pp. 40–44.

[5] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on signal processing, 54(11):4311, 2006.

[6] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” IEEE Trans. Image Process., vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[7] Aharon, Michal Elad, Michael Bruckstein, Alfred. (2006). K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. Signal Processing, IEEE Transactions on. 54. 4311 - 4322. 10.1109/TSP.2006.881199.