Oren Shavit 20597931

Nir Mualem 205467780

236860 – DIP – Digital Image Processing – Fina Project

Deep K-SVD Denoising

Paper by Meyer Scetbon, Michael Elad and Peyman Milanfar

The paper goal is to explore the old and known K-SVD denoising algorithm, the new contribution is to try and use deep learning methods while saving the flow and architecture as the original K-SVD.

The well-known K-SVD is a sparsity-based method which was, at the time (2006), one of the state-of-the-art methods for denoising problems. The Deep K-SVD work is showing that old and theory well-based algorithms as K-SVD still can contest the newer deep learning end-to-end algorithms in their architectures, meaning that the old architecture stay almost the same except entering some modern deep learning methods to improve the results.

By that, this work, in a broader context, is connecting deep-learning solutions for image processing tasks with classical algorithms that have a well-based theory. The results and the future works which will based on that idea might find a better explanation and theories for the "black-magic" of the deep learning while breaking the barrier for improvement of the denoising problem and other image-processing tasks.

In our work we will represent the paper and the K-SVD algorithm innovation, which called LKSVD in the paper, while explaining the architecture and the ideas behind its parts. We will explain our contribution and tries for improvements of the algorithm.

Our work structure:

1. Problem presentation, an explanation about image denoising problems.
2. Introduction to sparse representation.
3. Introduction for the classic K-SVD algorithm.
4. Explanation about the LKSVD, the innovation of K-SVD algorithm while preserving the architecture and using modern deep-learning methods for improvements.
5. The LKSVD training and results.
6. Results Reproduction.
7. Our tries for improvement and ideas.
8. Summary.
9. Previous works

1. <u>Problem Presentation.</u>

The classic image denoising problem, which we were introduced at the course, can be described as:

Let x be an ideal image.

Let v be a white and homogeneous Gaussian noise with a standard deviation σ.

The measured image y is created by adding this kind of noise to an ideal image, meaning, $y = x + v$.

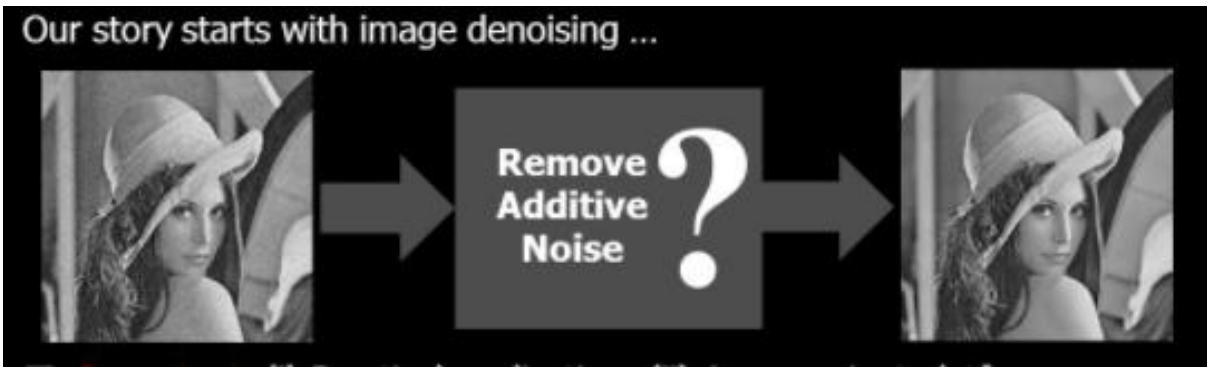
The goal is to recover x from y given the parameter σ.

The problem might see easy to describe but preservation of gentle details in x might be really challenging, as the additive noise interrupts the reconstruction of those fine details.

This problem has a quite similar reparation in each kind of signal, here we are focusing on images. The K-SVD algorithm is one of the sparsity-based method for solving this problem.

K-SVD was one of the state-of-the-art algorithms that demonstrate with this denoising problem when it was published (at 2006).

Now days, there are a few modern algorithms which have an end-to-end deep learning architecture for solving those kinds of problems. Currently, those modern solution are superior at this field regarding to results, although (or maybe because) they have not a well based theory and explanation as we all know deep learning is still a bit of a “black box”.



2. Introduction to sparse representation.

The classic K-SVD algorithm based on sparsity, in this section we will explain briefly what the idea behind sparsity representation for better understanding of the paper.

Sparse representation is a way to represent the data using linear combination of basic elements called atoms, the combination should be sparse. Therefore, for image x, a sparse representation might be achieved by the equation $x = D\alpha$ where D is a dictionary which is a composition of the atoms.

Finding a sparse representation of image x with a given dictionary D and some threshold

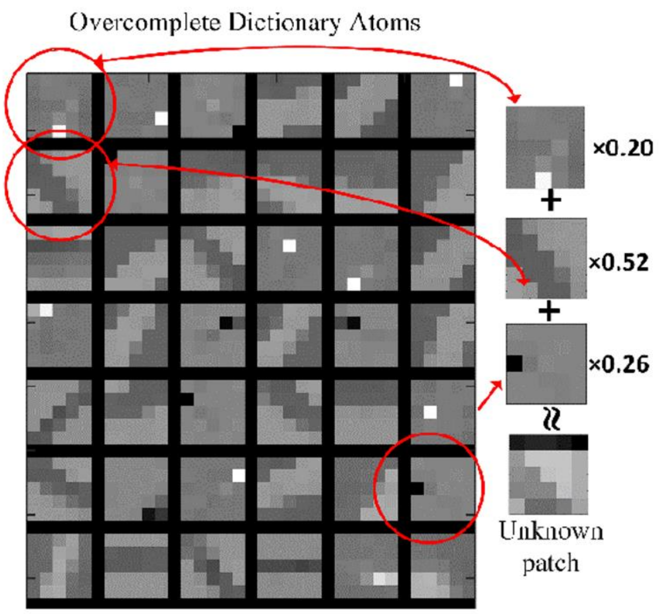
$\epsilon > 0$ can be formulated as the following optimization problem:

$$\underset{\alpha}{\operatorname{argmin}} \left|\left|\alpha\right|\right|_0 \; s.t. \left|\left|D\alpha - y\right|\right|_2^2 \leq \epsilon$$

Of course, the problem is even harder if D is unknown too.

The dictionary atoms are not required to be orthogonal; they can be even an over-complete spanning set. The dictionary might have seemingly redundant atoms which allowing multiple representations of the data and provides an improvement in sparsity and flexibility of the representation.

An overcomplete dictionary allows a sparse representation of signal can be one of the famous transform matrices (as wavelets transform or Fourier transform) or a formulated so that its elements are changed in such a way that it sparsely represents the image in a better way.



An example of a specific patch sparse representation using an overcomplete dictionary.

3. <u>Introduction for the classic K-SVD algorithm.</u>

First, let us describe the classic K-SVD Algorithm as was presented in reference number 1.

It starts by presenting a local prior on patches, rather than the entire image. Let x be such a patch, of size $\sqrt{p} \times \sqrt{p}$ ordered as a column vector of size p.

Like we learned in the course, we will represent x as a sparse combination of dictionary atoms. Let us denote the dictionary as $D \in \mathbb{R}^{p \times m}$ and the s as the number of atoms.

To know which atoms to use from the dictionary, we denote a sparse vector of coefficients $\alpha \in \mathbb{R}^m$, Such that:

$$x = D\alpha$$

Since we have s atoms for this patch, we can denote $||\alpha||_0 = s$.

Now let us add some noise. We will denote the noisy patch as y. We model the noise as additive white gaussian noise with zero mean and σ standard deviation.

As we learned in the course, using the MAP estimator, and aiming for sparsity we get:

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}} ||\alpha||_0 \; s.t. \; ||D\alpha - y||_2^2 \leq p\sigma^2$$

And then the estimated patch is

$$\hat{x} = D\hat{\alpha}$$

Now we will add a Lagrangian multiplier for the constraint:

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \lambda ||\alpha||_0 + \frac{1}{2} ||D\alpha - y||_2^2$$

(A spoiler for later - we will use deep learning to learn λ)

Now we will move to the global prior of the entire image denoted as X and the noisy image as Y using each local patch prior as written above. Both images of size $\sqrt{N} \times \sqrt{N}$ and as a column vector of size N.

We will denote R_k as an operator the extracts the kth patch from the entire image X.

Now the MAP estimator will be (where k is an iterator an all patches):

$$\min_{\{\alpha_k\}_k, X} \frac{\mu}{2} ||X - Y||_2^2 + \sum_k \left(\lambda_k ||\alpha_k||_0 + \frac{1}{2} ||D\alpha_k - R_k X||_2^2 \right)$$

The first part is the Lagrangian form of the constraint of $||X - Y||_2^2 \leq N\sigma^2$

Solution

Now we have three unknowns: the dictionary D, the coefficients $\alpha_k's$ and the clean image X.

Let us assume for now D is known and $X = Y$, now our goal will be to find the $\alpha_k's$.

If we look at each patch separately our objective is:

$$\hat{\alpha}_k = \underset{\alpha_k}{\operatorname{argmin}}\, \lambda_k \left|\left|\alpha_k\right|\right|_0 + \frac{1}{2}\left|\left|D\alpha_k - x_k\right|\right|_2^2$$

We will use OMP – Orthonormal Matching Pursuit, gathering the atoms until the error is below $p\sigma^2$.

Going back to our estimated image now that we have $\{\alpha_k\}_k$:

$$\hat{X} = \underset{X}{\operatorname{argmin}} \frac{\mu}{2}\left|\left|X - Y\right|\right|_2^2 + \frac{1}{2}\sum_k \left(\left|\left|D\alpha_k - R_k X\right|\right|_2^2\right)$$

With the closed form solution:

$$\hat{X} = \left(\sum_k R_k^T R_k + \mu \mathbb{I}\right)^{-1} \left(\mu Y + \sum_k R_k^T D \hat{\alpha}_k\right)$$

The matrix that needs inversion is diagonal which makes it a simple problem.

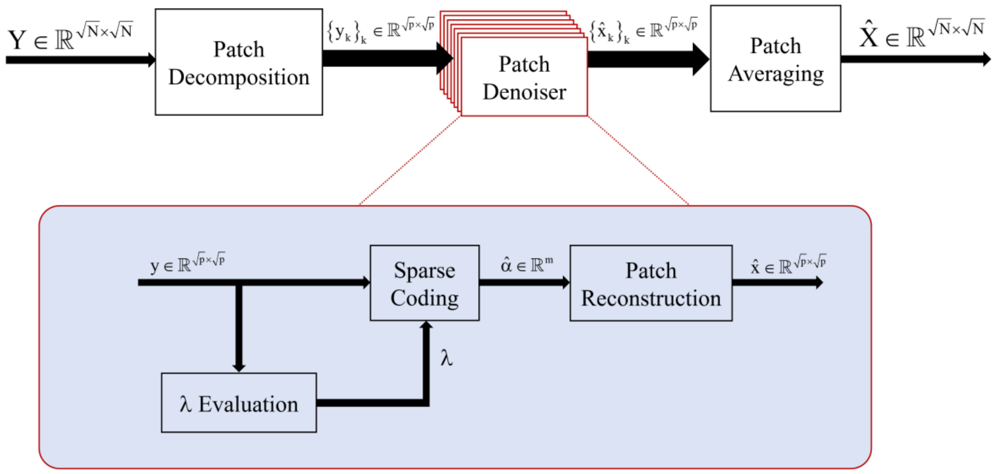
To find D, we initialize it as the DCT matrix and set $X = Y$.

We will iterate between the OMP and update D using the following objective:

$$\min_{\{\alpha_k\}_k, X, D} \frac{\mu}{2}\left|\left|X - Y\right|\right|_2^2 + \sum_k \left(\lambda_k \left|\left|\alpha_k\right|\right|_0 + \frac{1}{2}\left|\left|D\alpha_k - R_k X\right|\right|_2^2\right)$$

4. <u>Explanation about the LKSVD.</u>

The architecture from the paper:



For finding the coefficients $\alpha_k' s$, we can replace the l_0 norm to l_1 norm (which will later make this part differentiable and thus learnable):

$$\min_{\alpha\in\mathbb{R}^m} ||\alpha||_1 \ \ s.t. ||D\alpha - y||_2^2 \leq p\sigma^2$$

In a Lagrangian form:

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \lambda ||\alpha||_1 + \frac{1}{2}||D\alpha - y||_2^2$$

To solve this problem, as we learned in the course, we will use the ISTA algorithm:

- initializing $\alpha_0 = 0$
- c will be the square spectral norm of D
- $S_{\frac{\lambda}{c}}$ will be the component-wise soft thresholding operator:
  $[S_\theta(v)]_i = sign(v_i)(v_i - \theta)_+$

$$\hat{\alpha}_{t+1} = S_{\frac{\lambda}{c}}\left(\hat{\alpha}_t - \frac{1}{c}D^T(D\hat{\alpha}_t - y)\right)$$

Now, we will convert the problem into a learnable one.

Since the ISTA is operating on each patch we can use it as convolution and set c and D as the learnable parameters. And we will use a regression task of MLP – Multi Layer Perceptron to learn λ.

Now that we have all the missing parts, we can reconstruct the patches as was shown before. (Patch Reconstruction in the architecture image).

To reconstruct the entire image, we will weight each patch by a weight $w \in \mathbb{R}^{\sqrt{p}\times\sqrt{p}}$ which will also be learned:

$$\hat{X} = \frac{\sum_k R_k^T (w \odot \hat{x}_k)}{\sum_k R_k^T \hat{x}_k}$$

The overall number of parameters for learning λ, c, D, w as was shown in the paper is $32{,}865$.

For the learning task we will take a clean training set of images $\{X_i\}_i$ and synthetically noised images $\{Y_i\}_i$ and then when adding i.i.d noise $\{V\}_i$ as was described before:

$$Y_i = X_i + V_i$$

Now we will denote the K-SVD algorithm with the learning parameters as $\hat{X} = F(Y)$, forward pass on it, and optimize according to the MSE loss function:

$$\mathcal{L} = \sum_i \left|\left|X_i - F(Y_i)\right|\right|_2^2$$

Another interesting point mentioned in the paper is that in the classic K-SVD, each patch's noise is different which makes it a challenge but learning the parameter λ solves this issue.

5. <u>The LKSVD training and results.</u>

The authors of the paper trained for couple of hours on Titan Xp GPU.

Comparing to other classic denoising algorithms, we can see small improvements in **PSNR**:

Dataset	Noise	BM3D	WNNM	$KSVD_1$	$KSVD_2$	LKSVD
BSD 68	15	31.07	31.37	30.91	30.87	**31.48**
	25	28.57	28.83	28.32	28.28	**28.96**
	50	25.62	25.87	25.03	25.01	**25.97**

Where $KSVD_1$ is the image adaptive algorithm and $KSVD_2$ is the one using a universal dictionary.

We can see that **other deep learning methods** are better than the LKSVD:

Dataset	Noise	TNRD	NLNet	DnCNN	NLRNet	$LKSVD_{2,16,1024}$
BSD 68	15	31.42	31.52	**31.73**	**31.88**	31.54
	25	28.92	29.03	**29.23**	**29.41**	29.07
	50	25.97	26.07	**26.23**	**26.47**	26.13
Set 12	15	32.50	-	**32.86**	**33.16**	32.61
	25	30.06	-	**30.44**	**30.80**	30.22
	50	26.81	-	**27.18**	**27.64**	27.04

Table V: LKSVD versus learned methods: Denoising performance (PSNR [dB]) for various noise levels on BSD68 and Set12. Results exceeding LKSVD are marked in bold.

We can see the differences between the dictionaries (learned and classic):

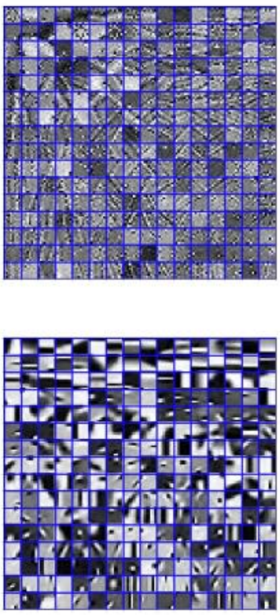


Figure 2: Comparison of the dictionaries (top: LKSVD, bottom: $KSVD_2$) for noise level $\sigma = 25$.

An interesting Note:

Another small experiment is checking the architecture in an unsupervised manner. After training the network and achieving better parameters, taking a new noisy image, and having a few more adaption rounds to the incoming image with its cleaning version from the LKSVD. In that way creating an unsupervised algorithm that is image specific. The results, at least in this experiment, were better than the supervised LKSVD. We will denote the unsupervised version as LKSVD-U.

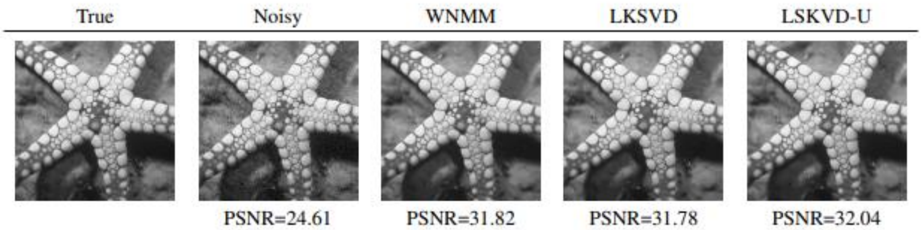
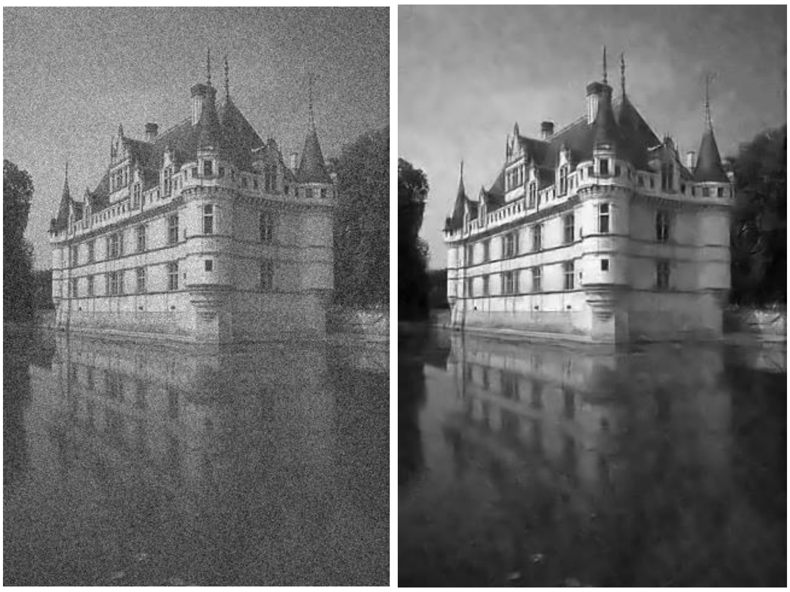


Figure 10: Self-adaptation: Denoising results for the image Starfish with noise level $\sigma = 15$. As can be seen, the additional adaptation leads to 0.26dB boost in performance, surpassing the WNNM method.

6. <u>Results Reproduction</u>

We were able to reproduce the paper results running the code. For example:

With PSNR of the noise one is 20.1542 and of the restored one is 27.3806.



With PSNR of the noise one is 20.1868 and of the restored one is 27.8358.

7. Our tries for improvement and ideas.

First, let us introduce some critical thinking on the paper:

- In the paper they used 32,865 parameters in the network. Perhaps less parameters could improve the run time and reduce complexity – maybe even better optimization and better results.
- For the loss function they used MSE. Perhaps a different loss function would have given better results. For example, L1, SSIM, MS-SSIM or a mix of them, as was checked in reference number 2 and 3.

We also added one linear layer to the network thinking increasing the complexity will make the model stronger, but the results actually did not improve.

8. Summary

The paper shows that K-SVD denoising algorithm can improve using deep learning – the Learned K-SVD, and even getting closer to modern deep-learning end-to-end based denoisers. This was achieved simply by setting the parameters of the classic K-SVD in another method (supervised learning), while preserving its original flow.

Our work tried to improve it even more with extra layer for preforming better learning with higher complexity, but could not be due to the lack of computation power and time for learning as we had to use Google Cloud GPU that has a free user limitation. Still, we noticed the great achievement of turning an old classic algorithm to a better one only by a better parameter choice (with deep learning).

As for the paper reveals, their goal goes beyond the KSVD denoising and its improvement, towards more fundamental questions related to the role and maybe even a theory base of deep learning in modern image processing methods.

9. Previous works

1. M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans. on Image Processing, 15(12):3736–3745, 2006.
2. Hang Zhao , Orazio Gallo , Iuri Frosio , and Jan Kautz. Loss Functions for Image Restoration with Neural Networks. IEEE TRANSACTIONS ON COMPUTATIONAL IMAGING, VOL. 3, NO. 1, MARCH 2017
3. Thomas Oberlin, François Malgouyres, Jin-Yi Wu. Loss functions for denoising compressed images: a comparative study. EUSIPCO, Sep 2019, Coruna, Spain. ffhal-02952604v2f