**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. Previous works related to this paper.
4. Introduction for the classic K-SVD algorithm.
5. Explanation about the LKSVD, the innovation of K-SVD algorithm while preserving the architecture and using modern deep-learning methods for improvements.
6. The LKSVD training and results.
7. Results Reproduction
8. Our tries for improvement and ideas.
9. Summary.

1. Problem Presentation.

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 .

The measured image y is created by adding this kind of noise to an ideal image, meaning, .

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

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”.

Graphical user interface, application

Description automatically generated

1. 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 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

can be formulated as the following optimization problem:

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](https://en.wikipedia.org/wiki/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.

Diagram

Description automatically generated

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

1. Introduction for the classic K-SVD algorithm.

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 be such a patch, of size ordered as a column vector of size .

Like we learned in the course, we will represent as a sparse combination of dictionary atoms. Let us denote the dictionary as and the as the number of atoms.

To know which atoms to use from the dictionary, we denote a sparse vector of coefficients , Such that:

Since we have atoms for this patch, we can denote .

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

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

And then the estimated patch is

Now we will add a Lagrangian multiplier for the constraint:

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

Now we will move to the global prior of the entire image denoted as and the noisy image as using each local patch prior as written above. Both images of size and as a column vector of size .

We will denote as an operator the extracts the th patch from the entire image .

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

The first part is the Lagrangian form of the constraint of

Solution

Now we have three unknowns: the dictionary , the coefficients and the clean image .

Let us assume for now is known and , now our goal will be to find the .

If we look at each patch separately our objective is:

We will use OMP – Orthonormal Matching Pursuit, gathering the atoms until the error is below .

Going back to our estimated image now that we have :

With the closed form solution:

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 .

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

1. Explanation about the LKSVD.

The architecture from the paper:

Diagram

Description automatically generated

For finding the coefficients we can replace the norm to norm (which will later make this part differentiable and thus learnable):

In a Lagrangian form:

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

* initializing
* will be the square spectral norm of D
* will be the component-wise soft thresholding operator:

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 and as the learnable parameters. And, we will use a regression task of MLP – Multi Layer Perceptron to learn .

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 which will also be learned:

The overall number of parameters for learning as was shown in the paper is .

For the learning task we will take a clean training set of images and synthetically noised images and then when adding i.i.d noise as was described before:

Now we will denote the K-SVD algorithm with the learning parameters as , forward pass on it, and optimize according to the MSE loss function:

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 solves this issue.

1. The LKSVD training and results.

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**:

Table

Description automatically generated

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

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

Table

Description automatically generated

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

A picture containing text

Description automatically generated

An interesting Note:

An unsupervised version of this architecture, the authors had a small experiment in the end of the paper for checking the architecture in an unsupervised manner. After training the net and parameters achievement, they took a noisy new image and had a few more adaption round to the incoming image with its cleaning version from the LKSVD. In that way they created an unsupervised algorithm that is specific image adopt. The results, at least in this experiment was a bit better the LKSVD.

Calendar

Description automatically generated

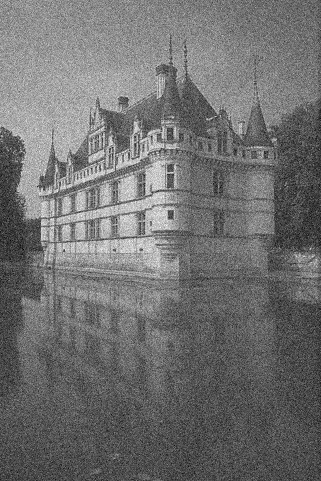
1. Results Reproduction

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

A person holding a stick

Description automatically generated with low confidence

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

 A picture containing outdoor, white, black, old

Description automatically generated

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

1. 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 3 and 4.

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.

Another, more immediate direction to answer the above challenge, is self-adaptation along the lines described in [38]. The core idea is to run the trained universal network to create an initial denoised result, followed by an adaptation round to the incoming image by few epochs on the image itself and its initially cleaned version. Figure 10 presents an example result of this idea on the image Starfish, showing a boost of 0.26dB in its denoising. This adaptation is obtained via the exact same training procedure as the one used to train the LKSVD network, where the target image is replaced by the restored image obtained from the universally trained LKSVD network and using only few minutes of training.

1. Summary

The paper shows that K-SVD denoising algorithm can become with much better performing, and even getting closer to modern deep-learning end-to-end based denoisers. This experiment was achieved simply by setting the parameters in other method (supervised), while preserving its original flow.

Our work tried to improve it even more with extra layer for preforming better learning with higher complexity, but sadly without the same computation power and time for learning as we had to use google cloud GPU that have 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.

Related 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