

Dictionary Based Filtering

Aatman Dholakia, 1401013, Parth Satodiya, 1401056, Anuj Shah, 1401084, Vishal Raiyani, 1401094

Abstract—Main objective is to denoise image efficiently using previous $N \times N$ patches of noisy images and it's filtered images and find nearest patch which compares to the original image from this dictionary and then compare it's complexity and efficiency with classical convolution based filtering. Also we compare efficiency between different image patch sizes.

Index Terms—Classical convolution, low pass filter, dictionary learning, $N \times N$ patches

I. INTRODUCTION

First we take some training images of dimensions $M \times N$ and filtered image is obtained by classical convolution. Now we divide this image into patches each of dimensions $P \times Q$ where $P, Q \leq M, N$ and store it in the dictionary. In the dictionary the key is noisy part of the image and the value is filtered part of the image. Now test image is taken and divided into patches and corresponding value is obtained by iterating the keys. If key is not found then add this new key - value pair in dictionary database.

II. FLOW CHART

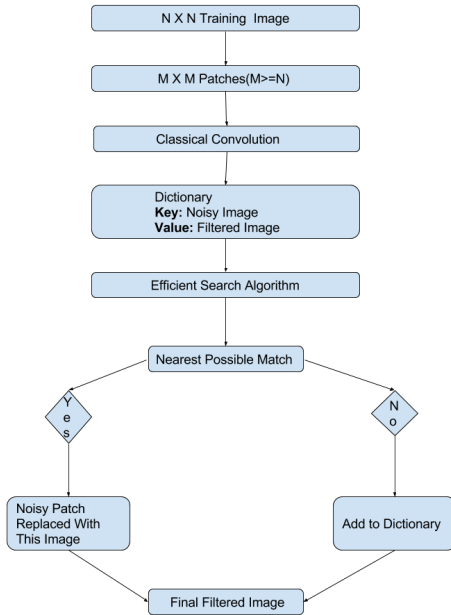


Fig. 1: Flow chart

III. METHODOLOGY

1) Low pass filtering

The most basic of filtering operations is called "low-pass". A low-pass filter, also called a "blurring" or

"smoothing" filter, averages out rapid changes in intensity. The simplest low-pass filter just calculates the average of a pixel and all of its eight immediate neighbors. The result replaces the original value of the pixel. The process is repeated for every pixel in the image.

Kernel used for low pass filter:

$$K = \begin{bmatrix} 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \end{bmatrix}$$

2) Classical convolution

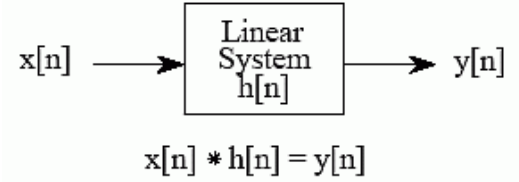


Fig. 2: Linear convolution

3) Salt and pepper noise reduction

Salt-and-pepper noise is a form of noise sometimes seen on images. It presents itself as sparsely occurring white and black pixels.

For reducing either salt noise or pepper noise, but not both, a contraharmonic mean filter can be effective.

It is done through low pass filtering.

4) K - SVD Decomposition Dictionary Learning

Most existing dictionary learning methods consider an over-complete dictionary and formulate the learning process as a minimization problem. K-SVD method learns an over-complete dictionary from an input image via solving the following minimization model:

$$\min_{D, \{\alpha_i\}} \sum_i \|g_i - D\alpha_i\|_2^2 + \lambda_i \|\alpha_i\|_0,$$

Fig. 3: K - SVD

where $g_i \in R_n$ is the collection of image patches after vectorization. $D = [d_1, \dots, d_k] \in R^{n \times k}$ with $k > n$ is the unknown over-complete dictionary.

5) Learned Dictionary

The concept of the adaptivity has been exploited to design the dictionary specifically optimized for the target image, the so-called dictionary learning. Starting from the set of overlapping image patches collected from the input image, our algorithm for dictionary

learning iterates between two sub-problems: sparse coding and dictionary updating. Both works on the principle of heuristic greedy method. The sparse coding is done via orthogonal matching pursuit (OMP).

6) Image denoising

Algorithm can be directly applied on de-noising by taking the noisy image as the input image for training. For computational efficiency the patches for trained image are uniformly selected from the image at random. The patches for denoising are the patches uniformly selected with overlaps.

7) Algorithm for denoising of image

- Set of training images are selected. ($\approx 10 - 20$)
- Each training image is divided into 1024 patches of 8×8 pixel each and stored in the dictionary using key - value pair.
- Salt and pepper noise is applied for which threshold is 0.5.
- Low pass filter is applied after above process.
- Dictionary is created for both low pass filtered image and salt and pepper image.
- Now a noisy test image is selected and it is compared with the low pass and salt and pepper dictionary using mse and a filtered image without noise will be obtained.

8) Methods of Comparing

• MSE

The mean squared error (MSE) of an estimator measures the average of the squares of the errors or deviations—that is, the difference between the estimator and what is estimated.

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2$$

One problem with mean-squared error is that it depends strongly on the image intensity scaling. Peak Signal-to-Noise Ratio (PSNR) avoids this problem by scaling the MSE according to the image range:

$$PSNR = -10 \log_{10} \left[\frac{MSE}{S^2} \right]$$

Where, S = Max. pixel value.

• SSIM

The structural similarity (SSIM) index is a method for predicting the perceived quality of digital television and cinematic pictures, as well as other kinds of digital images and videos.

The SSIM value can vary between -1 and 1, where 1 indicates perfect similarity.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

IV. SIMULATION RESULTS

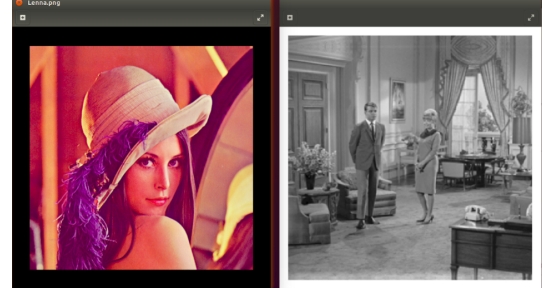


Fig. 4: Original image

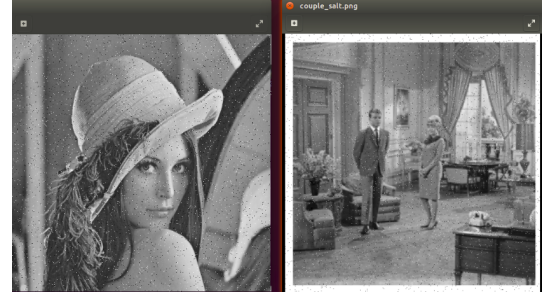


Fig. 5: Salt and pepper noise

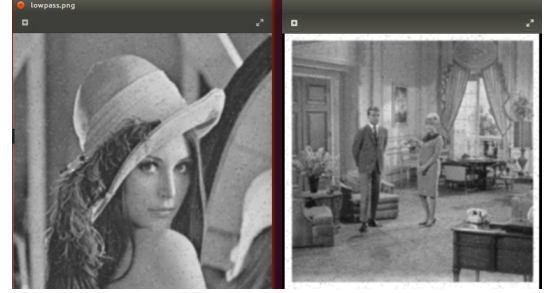


Fig. 6: Lowpass filter

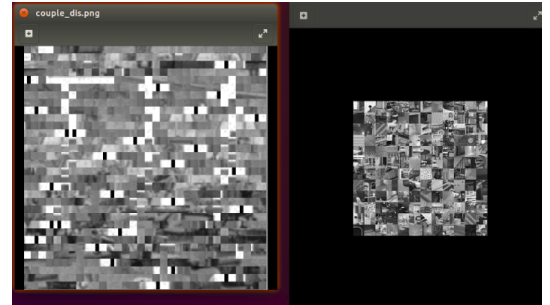


Fig. 7: Recovered image using mse - disoriented form

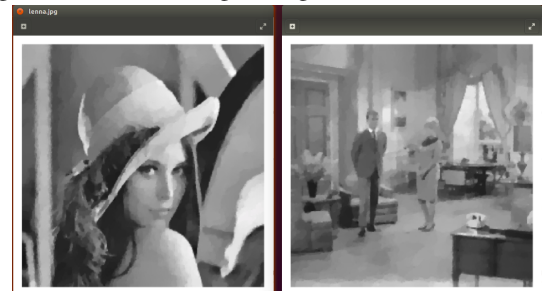


Fig. 8: Recovered image using mse

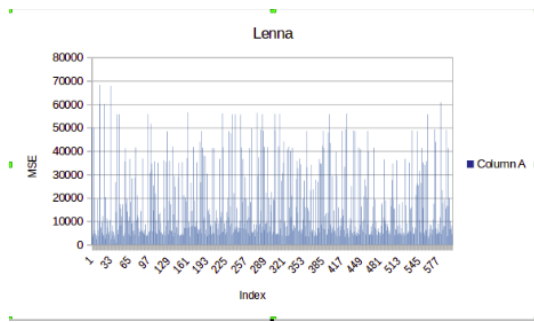


Fig. 9: MSE error of lenna test image

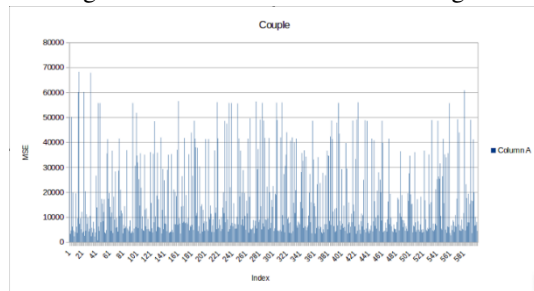


Fig. 10: MSE error of couple test image

V. CONCLUSION

- The MSE increases, it is to be implied that the images are less similar. Assume that original image has MSE = 0.
- K-SVD is better in performance than matching algorithm which compares for each and every image. But the output of two algorithms is almost the same.
- There is trade-off between performance and complexity. More number of patches implies better performance and more time complexity and vice versa.

VI. FUTURE WORK

- Improve the performance of the system which is currently $O(n^2)$
- At present we are taking only 10 - 20 images as training images. But we can include a large database to improve the accuracy of the system.

REFERENCES

- [1] Digital Image Processing, JAYARAMAN
- [2] "Median filter", En.wikipedia.org, 2017. [Online]. Available:
- [3] https://en.wikipedia.org/wiki/Median_filter. [Accessed: 03- Mar- 2017].
- [4] https://lear.inrialpes.fr/people/mairal/tutorial_iccv09/tuto_part2.pdf, 2017. Web. 6 Mar. 2017.
- [5] <https://pdfs.semanticscholar.org/45b0/676a6b8236cefaaeac0faabea159369fdb65.pdf>, 2017. Web. 7 Mar. 2017
- [6] <http://www.dspguide.com/ch6/2.htm>
- [7] https://en.wikipedia.org/wiki/Salt-and-pepper_noise
- [8] http://www.bogotobogo.com/Matlab/Matlab_Tutorial_Digital_Image_Processing_6_Filter_Smoothing_Low_Pass_special_filter2.php
- [9] <http://www.pyimagesearch.com/2014/09/15/python-compare-two-images/>