PATCH BASED IMAGE DENOISING THROUGH LOCALLY LINEAR EMBEDDING

ÇAĞATAY KIRMIZIAY

Master's Thesis

Graduate School
Izmir University of Economics
Izmir
2022

PATCH BASED IMAGE DENOISING THROUGH LOCALLY LINEAR EMBEDDING

ÇAĞATAY KIRMIZIAY

A Thesis Submitted to
The Graduate School of Izmir University of Economics
Master's Program in Electrical and Electronics Engineering

Izmir
2022

ABSTRACT

PATCH BASED IMAGE DENOISING THROUGH
LOCALLY LINEAR EMBEDDING

Kırmızıay, Çağatay

Master's Program in Electrical and Electronics Engineering

Advisor: Assoc. Prof. Dr. Mehmet Türkan

July, 2022

In this thesis, image denoising algorithms have been developed by means of Locally Linear Embedding (LLE) which is a dimensionality reduction method in data science. Although denoising has been studied for decades, it is still an active research area because there is not an upper and certain limit yet. By using LLE, new perspectives of image denoising are aimed to establish. Therefore, traditional patch-based approaches and basic dictionary learning algorithms have been developed. The main idea of using a patch-based process is to estimate sparse representations of denoised patches with LLE weights of nearest neighbor patches of each patch. In order to diminish the effect of the noise, various parameters have been analyzed such as patch size, dictionary size, dimension reduction size, number of the nearest neighbor patches, etc. Furthermore, different approaches have been tested such as alpha rooting, hard-thresholding in a transform domain, error based dictionary updating and feature mapping. According to the statistical results and visual assessments, preserving details in images is as much important as removing the noise effect. The experimental results demonstrate that the

developed algorithm based on alpha rooting has very promising results. Moreover, the denoising performance of the proposed method can compete against the well-known denoising algorithms in literature.

Keywords: denoising, patch processing, dictionary learning, linear embedding

ÖZET

YEREL DOĞRUSAL YERLEŞTİRME İLE
GÖRÜNTÜLERDE PARÇA TEMELLİ GÜRÜLTÜ GİDERME

Kırmızıay, Çağatay

Elektrik ve Elektronik Mühendisliği Yüksek Lisans Programı

Tez Danışmanı: Doç. Dr. Mehmet Türkan

Temmuz, 2022

Bu tezde, veri biliminde bir boyutsallık indirgeme yöntemi olan Yerel Olarak Doğrusal Yerleştirme (YDY) aracılığıyla görüntü gürültü giderme algoritmaları geliştirilmiştir. Gürültü giderme onlarca yıldır çalışılsa da henüz bir üst ve kesin sınır olmadığı için halen aktif bir araştırma alanıdır. YDY'yi kullanarak, görüntü gürültü gidermenin yeni bakış açılarının oluşturulması amaçlanmaktadır. Bu nedenle, geleneksel parça tabanlı yaklaşımlar ve temel sözlük öğrenme algoritmaları geliştirilmiştir. Parça tabanlı işlemi kullanmanın ana fikri, her bir parçanın en yakın komşu yamalarının YDY ağırlıkları ile gürültü giderilmiş parçaların seyrek temsillerini tahmin etmektir. Gürültünün etkisini azaltmak için parça boyutu, sözlük boyutu, boyutsallık indirgeme boyutu, en yakın komşu parça sayısı gibi çeşitli parametreler analiz edilmiştir. Dahası, alfa köklendirme, dönüşüm alanında eşikleme, hata tabanlı sözlük güncelleme ve özellik eşleme gibi yaklaşımlar denenmiştir. İstatistiksel sonuçlara ve görsel değerlendirmelere göre, gürültü etkisinin ortadan kaldırılması kadar görüntülerdeki detayların korunması da önemlidir. Deneysel sonuçlar, alfa köklendirmeye dayalı

olarak geliştirilen algoritmanın çok umut verici sonuçlara sahip olduğunu göstermektedir. Ayrıca, önerilen yöntemin gürültü giderme performansı, literatürdeki iyi bilinen gürültü giderme algoritmaları ile rekabet edebilir.

Anahtar Kelimeler: gürültü temizleme, parça işleme, sözlük öğrenimi, doğrusal yerleştirme

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude to my advisor Assoc. Prof. Dr. Mehmet Türkan for supporting me not only in my master's degree but also in everything, for guiding me for my career, and for his patience, effort and helpfulness. I feel fortunate to take a huge chance to work with them in my bachelor's and master's degrees. Thanks to him, I studied in a comfortable and friendly environment.

I also would like to thank the jury members Assoc. Prof. Dr. Pınar OĞUZ EKİM and Prof. Dr. Devrim ÜNAY for their constructive comments.

Furthermore, my sincere thanks to the Scientific and Technological Council of Turkey (TUBITAK) for the award-winning support "2210-A National Scholarship Program for MSc Students".

Finally, I would like to express my deepest and foremost thanks to my beloved Sahranur Tabakoğlu for her regardless support even under the most challenging circumstances. She helped me not only in my thesis but also everything I need to overcome. She is my inspiration source. I feel very lucky she is in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Human beings have tried to infer meaningful and useful information from the environment they lived in for thousands of years. Back in the day, there were not various kinds of data that people can reach. However, today there are many types of data due to the developing technological devices such as mobile phones, satellites, cameras, personal computers, smart-watches, a variety of sensors, etc. One of the efficient collectible data is the images captured through different imaging tools such as cameras, microscopes, telescopes, and medical imaging devices. Although images may be captured for different purposes, they have a common point. This common point is that they need to be processed to extract some critical information which people could not recognize or to give machines visual perception. The quality of the processed outcomes can be affected easily by physical, electronic, and software factors, even by broadcasting. For example, a dust particle on the camera lens can damage the image data or rainy weather conditions can ruin it. Another thing that can disturb the image quality is the noise. Noise can appear in images naturally by changing the intensity values of pixels randomly. As an example, if you want to take a photo in a dimmed light or dark, as it is shown in Figure 1, you will see these damaged pixels at high ISO speeds which is a setting parameter of digital cameras indicating the sensitivity to light. Different types of noise (e.g. salt & pepper, gaussian) can corrupt images at different levels, and the noise effect should be removed or diminished to obtain the meaningful, required data properly for the ultimate purposes. Completely removing the noise effect is the best and desired case. Unfortunately, it is almost impossible to find desired (noise-free) image because of generally unknown noise characteristics. As a consequence, new methods have been developed by researchers to improve the performance of the denoising process for decades, and image denoising is still an attractive topic since it is the fundamental step of many image processing applications. Nowadays, even if deep learning algorithms are commonly employed rather than traditional mathematical approaches, traditional approaches have been providing sources of inspiration for both traditional and learning algorithms.

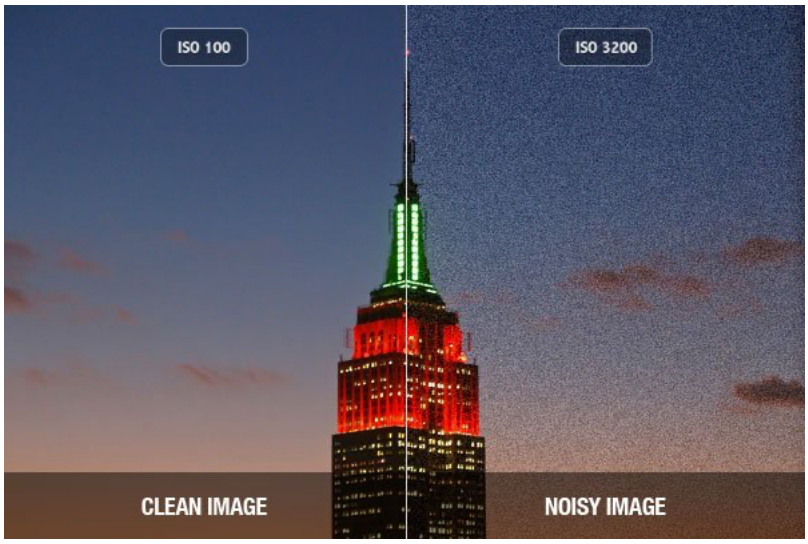


Figure 1. An example of noise at high ISO speeds (Source: Kun, 2012).

Up till today, many approaches were published processing patches of images. One of them is the well-known patch matching. The studies of Buades, Coll and Morel (2005) and Mairal et al. (2009) consider the non-local neighbor patches for processing. Similarly, the study in Zhang and Gunturk (2008) can be mentioned as an example if the patch size is one pixel. One of the famous methods that are exploiting the similarity of patches and their sparsity is Block Matching and 3D collaborative filtering (BM3D) (Dabov et al., 2007b) and its derivatives (Dabov et al., 2007c, 2007a, 2009; Yang and Sun, 2018). K-Singular Value Decomposition (K-SVD) (Elad and Aharon, 2006) is another well-known method that learns and uses the sparse dictionaries from patches of images, also the studies of Li and Liu (2009) and Scetbon, Elad and Milanfar (2021) were derived from K-SVD. As other approaches, multi-scale image denoising methods by using Discrete Cosine Transform (DCT) pyramids (Pierazzo, Morel and Facciolo, 2017), Weighted Nuclear Norm Minimization (WNNM) by applying convex relaxation techniques on non-local patch similarity process (Gu et al., 2014).

In this thesis, the image denoising problem is discussed since it is an active research area and it has high significance for many image processing applications. By using traditional and basic learning approaches, the performance of methods and the effects of parameters are analyzed. The core of all methods is the Locally Linear Embedding (LLE) algorithm which is a dimension reduction method (Roweis and Saul, 2000). Also, the LLE method is employed for super-resolution (Türkan, Thoreau and Guillotel, 2012, 2013) and image restoration applications (Shi, Shen and Chen, 2005;

Fei and Medioni, 2010). Hence, the calculation of weights between neighborhoods and reference data can be exploited for the image denoising problem. Because considering images with their full size is not easy to solve optimization problems mathematically, small blocks (patches) are considered as the benchmark methods in this study. Most of the image denoising algorithms reveal the relationship between patches by representing the weights between them. Eventually, the idea that the combination of the well-known approaches and the LLE may provide us with a new perspective on the image denoising problem is presented in this study.

The brief explanations of the reference studies guiding us are given in Chapter 2. Our methods built in the light of reference studies are described in detail in Chapter 3. Statistical results and sample outputs are presented in Chapter 4. After all, outcomes and future works of the proposed method are presented in Chapter 5 to conclude the thesis.

CHAPTER 2: LITERATURE REVIEW

2.1 Non-Local Means Algorithm

In this study, Buades, Coll and Morel (2005) developed an image denoising algorithm called Non-Local Means (NL-means). The main idea of this method is based on the estimated noise-free image's pixels' values can be interpreted by using the noisy image itself. Since original clear images are unknown, this approach is helpful for the blind denoising processes.

According to the approach, a pixel value can be calculated as a weighted average of pixels in the noisy image. In order to decrease the complexity of the process, similar pixels are found in the related search window including the pixel to be processed. The similarity between pixels is based on the weighted Euclidian distance of fixed-sized neighbors of the related pixels by using the Gaussian kernel.

Although the NL-means algorithm has common points with the neighborhood filtering (Yaroslavsky, 1985), it takes into account the geometrical configuration of related pixels' neighbors differently. It means to determine the similarity comparison of two pixels in grey level is not sufficient, also neighbors should be considered. Moreover, by considering the neighborhood of each pixel, the algorithm eliminates some problems of processing a single pixel such as not robust against noisy values and creating artifacts.

As a result, because of the application's simplicity and effectiveness, and being applicable for color images, NL-means is one of the well-known image denoising algorithms.

2.2 Block-Matching and 3-Dimensional Filtering Algorithm

The research in Dabov et al. (2007b) presents a state-of-art image denoising algorithm. For years, the algorithm with block-matching and 3-Dimensional filtering (BM3D) has been one of the most successful and famous algorithms. BM3D combines the block-matching algorithms used to extract the similar patches from the noisy image and the filtering in the 3D transform domain of the stacked structure of similar patches. Most commonly, the main goal for image denoising algorithms is to diminish the effect

of the noise in an image. In order to do that, noise in the transfer domain can be employed. While the transform-based approaches achieve outstanding performance in terms of statistics, they are not good at conserving the details by using the transformation. As a result, artifacts can appear in the estimated image. As an alternative approach, non-local estimation based methods may come to mind. Since these algorithms consider the average of similar regions' pixels' values to calculate an estimation of the processed region, the output images are oversmooth. The proposed BM3D algorithm combines these approaches to extinguish the bottlenecks of both algorithms.

The whole BM3D algorithm is summarized in Figure 2. This algorithm can be divided into two parts. In the first part, similar patches are found for each sliding patch of the noisy image. To determine the similarity between the patches Euclidian distances of thresholded values of the (2-dimensional) 2D transformed (e.g. DCT, DFT, etc.) patches are used. The advantage of this similarity calculation is filtering the noisy patches roughly in the 2D transfer domain to find similar patches more accurate. After collecting the reference patch and noisy similar patches in a (3-dimensioanl) 3D matrix, which is essential to increase the sparsity, 3D filtering is performed in the 3D transfer domain of the stack. By applying the 3D transform and filtering on the stacked patches, transform coefficients are thresholded to diminish the noise. The estimated noise-free version of the reference patch can be obtained by applying an inverse-3D transform. The whole estimated noise-free image is generated by considering the weighted average of estimated patches. In the second part of this algorithm, almost the same procedure is repeated with small differences. One of the differences is that the estimated image of the first part is utilized to find similar patches. Another one, the L2-norm of difference of patches is employed for block-matching. Also, Weiner filtering is applied on the 3D transform-domain instead of thresholding. At the end of the second part, by taking the weighted averages of the estimated patches, the final estimated version of the clean (noise-free) image is obtained.

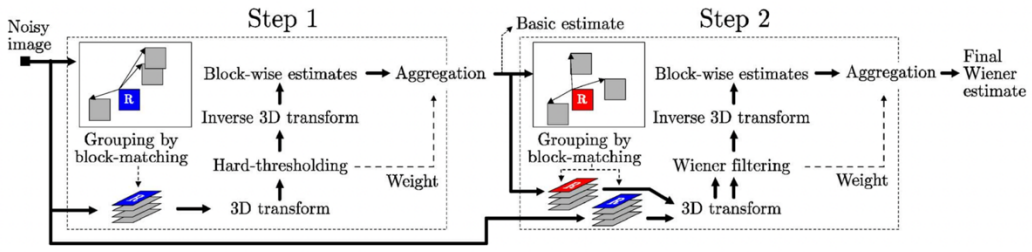


Figure 2. Scheme of the BM3D Algorithm (Source: Dabov et al., 2007b).

According to the statistical results, the BM3D method has an outstanding performance in terms of not only objective criteria such as Peak Signal-to-Noise Ratio (PSNR) but also visual assessment. In addition to the success of the algorithm, it is applicable to color images (Dabov et al., 2007a) and is promising to improve the algorithm by considering the complicated bases such as Principle Component Analysis (PCA) (Dabov et al., 2009).

The BM3D algorithm first was applied to greyscale images. It is also useful for color image denoising. As mentioned in (Dabov et al., 2007a), the whole procedure can be used to process each channel of luminance-chrominance color space such as YCbCr. However, since the Y channel, the luminance channel, contains the important image structure data and has high Signal-to-Noise Ratio (SNR) value than Cb and Cr channels (chrominance channels), it is sufficient to apply the BM3D algorithm on just the luminance channel. For the process on chrominance channels, patches sharing the same locations as the patches used for the process for the luminance channel are used. Even if the Cr chrominance channel has a low SNR value because of the iso-luminant image parts and sharp transition in the Cr channel, these are not factors that can affect the performance of the algorithm. As a result, statistical and visual results are still satisfactory.

In order to improve the performance of the above developed BM3D algorithm, some elaborative moves are done at some points. The main goal of these improvements is to strengthen the sparsity in the 3D transfer domain. For the step of finding similar patches, adaptive-shape neighborhoods and an enclosing square block of each reference pixel are considered (Dabov et al., 2009). Similar blocks are found based on the distance between the enclosing square block, then an adaptive-shape neighborhood is extracted from each similar block. According to the ratio of the number of similar

neighborhoods to the number of pixels in the reference adaptive-shape neighborhood, the 2D transformation method is determined. If the ratio is under the pre-defined threshold value which means there is not sufficient similar neighborhood data to train the algorithm, the fixed-sized shape adaptive DCT (SA-DCT) should be used. However, even supposing that the adaptive-shape neighborhood approach facilitates the local adaptivity to image features, SA-DCT is not useful for the local adaptivity. Instead of the SA-DCT, PCA which depends on the eigenvalue decomposition should be employed. The eigenvectors which have eigenvalues greater than the threshold are selected. Following that, a transformation in the third dimension and shrinkage in the 3D transfer domain should be applied respectively. In order to reconstruct the denoised image, the same procedure as the BM3D is used. Although the statistical results in terms of PSNR do not have an increasing attitude for all test scenarios compared to the BM3D, it is worth studying for further improvements.

Since the image denoising methods provide smooth results, a sharpening process should be considered as another way to improve the performance of the BM3D. According to the study (Dabov et al., 2007c), in order to contain the details in the denoised image, the thresholded coefficients in the 3D transfer domain can be augmented. This augmentation is named alpha-rooting and Eq. (1) is given as,

$$t_{sh}(i) = \begin{cases} sign[t(i)]|t(0)| \left|\frac{t(i)}{t(0)}\right|^{\frac{1}{\alpha}}, & if\ t(0) \neq 0 \\ t(i), & otherwise \end{cases} \quad (1)$$

where t is a transform spectrum of a signal, $t(0)$ is the DC coefficient, and t_{sh} is sharpened transform spectrum. The following steps are the same as the classical BM3D method, except for the weight definition for aggregation. The weight of the estimated patch is calculated by considering the total variance of the thresholded and sharpened block-matched group.

The alpha-rooting method can be applied to both 2D and 3D spectrum of the group. When the results of the two approaches are analyzed subjectively, we can say that details of the images can be preserved based on the alpha value. Therefore this application is promising to apply to color images and is open to improvement.

All improvements on the classical BM3D are not done by using only traditional

mathematical approaches, also the classical BM3D is tried to improve with some deep learning approaches. One of them is the BM3D-Net (Yang and Sun, 2018). Although most recently learning-based algorithms have been used widely, the researchers want to emphasize that combinations of traditional and deep learning approaches may broaden the horizon of the new applications. In this manner, the processing steps of the BM3D are adapted to the Convolutional Neural Network (CNN) structure. The proposed BM3D-Net algorithm contains five layers which are extraction, convolution, nonlinear transform, convolution, and aggregation. The extraction layer takes the place of the block-matching process of the BM3D. The convolution layer, nonlinear transform layer, and the other convolution layer stand for the wavelet transform, shrinkage of the wavelet transform coefficients, and inverse transform respectively. The last layer, the aggregation layer, is for the reconstruction process to reach the estimated noise-free image. Besides these five layers representing all the steps of the classical BM3D, this proposed algorithm has an additional loss layer to train the network. According to the objective results, BM3D-Net can provide satisfactory outcomes as much as the BM3D provides, and has the potential to be developed by using deeper neural network architectures.

2.3 Image Denoising Via Sparse And Redundant Representations Over Learned Dictionaries

Another well-known and effective algorithm is presented in (Elad and Aharon, 2006). The K-Singular Value Decomposition (K-SVD) method is an iterative image denoising method that follows a sparse representation of the given signal by using the current dictionary and dictionary learning steps respectively. Indeed, fixed, off-the-shelf dictionaries also can be employed in the same optimization approach. However, the main contribution of the K-SVD is to learn the dictionary by updating the atoms (columns) of the dictionary based on examples that use the corresponding atom.

As it is mentioned before, the K-SVD algorithm has two main steps. The first one is sparse coding. According to the sparse coding, the estimation of the true (noise-free) image can be calculated as a linear combination of the dictionary atoms by minimizing the number of non-zero coefficients in the sparse representation of the given data. In order to model this process mathematically, a local “sparseland model” is created with

an error boundary, a number of atoms to be used from the dictionary, and a redundant dictionary. To find the weights, although other pursuit algorithms can be employed, the K-SVD employs the Orthogonal Matching Pursuit (OMP) algorithm. After sparse coding is done, the next step is updating the dictionary by considering the calculated sparse signal in the previous step. In order to do that, each atom of the current dictionary should be updated individually by dealing with the corresponding sparse signals (examples). The update step is based on the minimization of the residual between the examples and generated examples with the dictionary atom and its corresponding sparse coefficients by using the Singular Value Decomposition (SVD).

In summary, the K-SVD is one of the famous and benchmark methods in image denoising, especially among the methods that are using learned dictionaries. Moreover, due to the novelty of this study, many other traditional and deep learning based studies that use a similar approach are derived from K-SVD (Li and Liu, 2009; Scetbon, Elad and Milanfar, 2021).

2.4 Weighted Nuclear Norm Minimization with Application to Image Processing

The image denoising problem has been extensively studied in the last two decades, especially after the utilization of similar patches of noisy images to estimate noise-free images. Grouping the non-local similar patches and processing the groups has become an inspiration source for many other studies. This approach is also attractive for the low rank matrix factorization (LRMF) problem, hence a group of similar noisy patches provides a low rank matrix. With the provided low rank matrix coming from the noisy patches, the main aim is to find the denoised image patches by exploiting the nuclear norm minimization (NNM) as a convex relation method of the LRMF. Since the classic norm minimization adjusts the singular values with equal weights to preserve the convexity of the objective function, this method is not effective and flexible for practical problems usually. The proposed method in (Gu et al., 2014), called Weighted Nuclear Norm Minimization (WNNM), removes this main disadvantage of the NNM by applying weighted soft-threshold to singular values to keep the prior and significant data contained by higher singular values. Moreover, the proposed method mathematically analyzes the weighting process by considering different conditions and properties of the weights. To evaluate the performance of the algorithm, PSNR

statistics can be compared with the statistics of the BM3D which is one of the benchmark algorithms. As a result, WNNM has better performance, even in preserving small details in images.

CHAPTER 3: METHODOLOGY

In this chapter, different approaches have been proposed for the image denoising problem. According to Eq. (2), the main goal is to find an estimation of the image X,

$$Y(i) = X(i) + N(i) \tag{2}$$

where Y stands for the noisy image, X represents the clean noise-free image, N is the noise, and i denotes the 2D spatial coordinate of each pixel. As mentioned, to find X at once is not an easy optimization problem because of the data size. Therefore, patches of the clean image (x_i) should be estimated by using patches of noisy image (y_i) to reconstruct the estimated clean image ($\hat{X}$), where ($\hat{x}_i$) denotes the estimated clean patches. By using controlled data set of clean and noisy images, the performance of the algorithm can be defined as the equation Eq. (3),

$$Error = min\left|X - \hat{X}\right|_2^2 \tag{3}$$

where the above equation is an error minimization problem. These notations and the error minimization model will be used throughout the thesis. More specifically, the required notation and the models are explained in the remaining part of this chapter. The whole study can be divided into two subtitles: (i) traditional approaches and (ii) learning approaches.

3.1 Traditional Approaches

3.1.1 Version 1

In this version, pixel based processing is employed by considering the patch size of *n*-by-*n* where *n* is an odd number. It means for each pixel of a noisy image, $Y(i)$, there is a corresponding reference patch (y_i) which is containing the pixel at the center. By finding the similar patches in a search window size of *t.n*-by-*t.n* where *t* is an odd number. Due to the adjustments to the size of patches and search windows, it is ensured that the reference patch (y_i) is located at the center of the related search window (s_i). To explain the process on one pixel, similar patches of the reference patch ($\tilde{y}_{i,k}$) are extracted from the corresponding search window to create a dictionary (D_i) peculiar to the reference pixel as it is shown in Figure 3. The weights ($\omega_{i,LLE}$) of the atoms (columns) of the D_i are calculated by using the LLE approach by considering the constraints on $w_{i,LLE}$ given as $\forall\omega_{i,LLE}(j) \in [0,1]$ and $\sum_j \omega_{i,LLE}(j) = 1$ where $j = \{1,2,3,\dots,n^2\}$. By using the LLE weights and the D_i, the $\hat{x}_i$ is calculated.

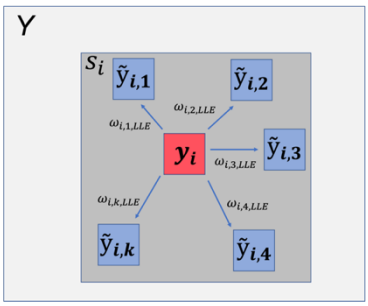


Figure 3. Selecting similar patches to the reference patch from the corresponding search window.

The $\hat{X}(i)$ value, located at the center of the $\hat{x}_i$, should be placed back into the $\hat{X}$. In the first trial of this version, all patches inside the s_i are used. In the second trial, to select closer patches in terms of similarity distance, an error boundary is defined according to the percentage of the maximum distance. Moreover, in the third trial, in order to determine the similar patches, a PCA based dimension reduction method is employed on D_i where each atom of D_i has n^2 elements. By applying dimension reduction, not only computational cost is decreased but also the noisy coefficients are eliminated by keeping a higher PCA coefficient. Because the PCA is based on the eigenvectors, higher coefficients contain the significant components of the data (Wold, Esbensen and Geladi, 1987).

3.1.2 Version 2

The main difference between Version 1 and Version 2 is that estimated values are considered to reconstruct the image. More clearly, the estimated values of the reference pixels are relocated in the Version 1, but in the Version 2 whole estimated patches, $\hat{x}_i$, are relocated. According to that, the core idea of our approaches are represented with Eq (4).

$$arg \min_{\omega_{i,k,LLE}} \left\| y_i - \sum_k \omega_{i,k,LLE} . \tilde{y}_{i,k} \right\|_2^2 \; subject\ to$$
$$\sum_k \omega_{i,k,LLE} = 1 \ \wedge \ 0 < \omega_{i,k,LLE} < 1, \forall i,k \tag{4}$$

Since patches are overlapping, the equally weighted average of the overlapping values of each pixel should be relocated. According to the Version 1 results, the same procedure as Version 1 Trial 3 procedure is employed except the reconstruction of $\hat{X}$.

3.1.3 Version 3

Although all atoms of the D_i are used to calculate the $\hat{x}_i$ in the Version 2, all atoms may not be required for the calculation. By using iterative approach to select the atoms one by one, the calculation of $\hat{x}_i$ continue until the sum of squared error (SSE) is less than a predefined error boundary ε which is defined in Eq. (5),

$$\varepsilon = (c\sigma n)^2 \tag{5}$$

where c is a constant number, σ is the standard deviation of the noise appearing in the noisy image Y. The usage of the iterative approach by considering the ε is originated from Dabov et al. (2007b). The selection of atoms is done based on maximum correlation values for each iteration. In the first iteration, the correlation between the normalized D_i and the normalized y_i is considered. For the further steps of the iterative calculation, the correlation between the normalized D_i and normalized version of error vector defined as $e = y_i - \hat{x}_i$. If the $\sum_j e^2(j)$, where $j = \{1,2,3, \dots, n^2\}$, is less than the ε, the iterative calculation should be stopped.

In the first trial, same approach of searching similar patches as Version 2 is performed by adding an iterative calculation of $\hat{x}_i$. Since finding similar patches is done in a corresponding search window, similar patches can be repeated in the D_i. The reuse of atoms can cause calculation of $\hat{x}_i$ repeats themselves using the identical atoms without decreasing the error e. Thus, in Trial 2, the step of keeping unique atoms is adopted. The iterative calculation of $\hat{x}_i$ is modified for Trial 3 by including a residual dictionary update stage. In each iteration, $D_i^{new} = D_i - e$ updating is done before selecting the atom that has a higher projection value onto D_i. Up to this point, the reconstruction of $\hat{X}$ is based on an equally weighted average of overlapping $\hat{x}_i$ for each pixel. However, the error of each calculation is not equal, so for reconstruction of $\hat{X}$, overlapping $\hat{x}_i$ patches should be weighted based on difference between $\varepsilon - \sum_j e^2(j), j = \{1,2,3, \dots, n^2\}$. For Trials 5 and 6, the difference from Trial 4 is the number of atoms and the step of finding unique atoms. In Trial 5, without finding the unique atoms, at least eight atoms are considered. In Trial 6, a fixed number of atoms is considered.

3.1.4 Version 4

In our approach and in studies we consider as reference ones (Elad and Aharon, 2006; Dabov et al., 2007b), the main aim is to find the coefficients of each atom that can

sparsely represent the noise-free patch. From this perspective, the L1-norm minimization approach can be employed to find the optimum coefficient values by considering the linear programming equation as in Eq. (6) for the first trial,

$$\min_{\omega_i} f^T \omega_i \ \ such\ that \begin{cases} A_{eq}\omega_i = b_{eq} \\ l_b \leq \omega_i \end{cases} \tag{6}$$

where A_{eq} is defined as the concateanation of the dictionary $A_{eq} = [D_i, -D_i]$, and f is defined as $f = [1, \dots, 1]_{1-by-n^2}$. A_{eq}. l_b is a lower boundary given as a column vector of all zeros. The dictionary D_i consists of all unique neighbor patches extracted from the corresponding s_i except the reference y_i. According to solution of the problem, $\hat{X}$ can be calculated as eqaually weighted average of overlapping estimated denoised patches, $\hat{x}_i$, defined as $\hat{x}_i = A_{eq}\omega_i$. In the second trial, almost the same procedure as Trial 1 is employed except the definition of f. It is defined as a row array of all ones.

3.1.5 Version 5

Version 5 is derived from the residual dictionary updating calculation of Version 3. To extract the dictionary, all possible patches of downsampled ($Y_{\downarrow}$) of the Y are used without corresponding search window. The main idea of resizing is to change the noise behavior from coarse to fine (Zhang and Gunturk, 2008). In the first trial, the patches extracted from the original size of Y are omitted to create the dictionary. The number of atoms used for calculation is fixed. For the reconstruction part, an equally weighted averaging of the overlapping patches is employed. The main difference between Trials 1 and 2 is the original size of the Y is used to create a dictionary. In Trial 3, Trial 2 is exploited by considering the weighted reconstruction of the overlapping patches. In addition to Trial 3, only the original size is used for Trial 4. Up to this point, to resize the Y, bicubic filters are used. However, to obtain $Y_{\downarrow}$ of the Y, Gaussian pyramids also can be employed. This approach has the advantage of using a Gaussian filter for resizing by changing the behavior of the noise at the same time. While the original level (starting) level is used for Trial 5, the original level is not used for Trial 6.

3.1.6 Version 6

The main aim of using similar patches is to exploit the sparse representation of the noise-free patch. To increase the sparsity, a shrinkage process (e.g. hard-thresholding) can be performed on 2D or 3D transform spectrum of similar patches. This shrinkage

approach is named as "*collaborative filtering*" in the study Dabov et al. (2007b). By applying collaborative filtering on the Discrete Cosine Transform (DCT) of each n-by-n patch, the coefficients that are representing the high frequency components, such as noise, will be eliminated. Therefore, the 2D collaborative filtering is utilized on the selected atoms from the dictonary. The iterative calculation by using residual dictionary updating is employed with a fixed number of similar patches. Thus, Version 6 is derived from Version 5 Trial 3 by adding the shrinkage on the 2D DCT spectrum.

3.2 Transfotmation Matrix Learning

The main purpose of the approach explained here is to reveal the relationship between $\omega_{i,LLE}$ of noisy patches and clean pathces by learning the transformation matrix (T_σ) utilizing a set of clean images (θ_m) and a set of corresponding noisy images (β_m) at a certain level of noise specified by standard deviation of the noise σ.

3.2.1 Learning Dictionary With Original Sized Images

In this version, only original sized images are used to learn the transformation matrix. Using a dataset (Set12), N samples noisy patches (b_N) are selected randomly from each noisy image (β_m) and the locations of these samples are kept. For each sample, a fixed number of similar (*k-Nearest Neighbor,k-NN*) patches are found in the dictionary containing all possible patches from β_m. Between each b_N and its *k-NN* patches, $\omega_{N,LLE,\beta}$ are calculated and stored by a concatenatenation step. As a next step, $\omega_{N,LLE,\theta}$ between N samples of clean patches (a_N), which have the same locations with b_N patches, and *k-NN* patches of b_N patches is calculated by concatenation. When the process is completed as defined above for each image in both data set, T_σ is calculated according to Eq. (7),

$$T_\sigma = \omega_{LLE,\theta}.\omega_{LLE,\beta}^T.(\omega_{LLE,\beta}.\omega_{LLE,\beta}^T)^{-1} \quad (7)$$

where $\omega_{LLE,\theta}$ and $\omega_{LLE,\beta}$ are the concatenated weights coming from each θ_m and β_m, respectively. After T_σ is found, a test image Y is used. For each y_i of Y, the LLE weights are calculated between y_i and its *k-NN* patches represented as $y_{i,kNN}$. The calculated weights are transformed to ω_{i,LLE,T_σ} by using the T_σ. To obtain $\hat{x}_i$ patches, Eq. (8) is applied. $\hat{X}$ is reconstructed via equally weighted average of overlapping $\hat{x}_i$ patches.

$$\hat{x}_i = y_{i,kNN} . \omega_{i,LLE,T_\sigma} \quad (8)$$

The whole procedure is the definition of the first level of this approach. For further levels, at the beginning of the process, while keeping set of α images is same, set of β images should be updated with $\hat{X}$ images of the previous level. The algorithm of the first level is given in Algorithm 1.

Algorithm 1. The first level of transormation matrix learning by using original sized images

Learning

Input: Set of Noisy Images (β), Set of Clean Images (θ)

Output: Transformation Matrix (T_σ)

for each β_m and θ_m **do**

  Select N noisy patches randomly ($b_{N,i}$) (i: location)

  for each $b_{N,i}$ **do**

    Find *k-NN* patches

    Calculate $\omega_{N,LLE,\beta}$ and store in $\omega_{LLE,\beta}$

    Extract $a_{N,i}$

    Calculate $\omega_{N,LLE,\theta}$ and store in $\omega_{LLE,\theta}$

  end

end

Calculate $T_\sigma \leftarrow \omega_{LLE,\beta}$, $\omega_{LLE,\theta}$

Testing

Input: Noisy Test Image (Y), T_σ

Output: Estimated Noise-Free Image ($\hat{X}$)

Extract all patches of Y (y)

for each y_i **do**

  Find *k-NN* patches

  Calculate ω_{LLE}

  Transform weights: $\omega_{LLE,T_\sigma} = T_\sigma . \omega_{LLE}$

  Calculate $\hat{x}_i = k\text{-}NN . \omega_{LLE,T_\sigma}$

end

Reconstruct $\hat{X} \leftarrow \hat{x}$

3.2.2 Learning Dictionary With Resized Images

In Section 3.2.1, the structure of the learning algorithm to find the T_σ is defined. In this section, the same structure is used with a difference about the set of images. In the

previous setup, sample patches are extracted from original sized θ_m and β_m images. In addition to that approach, resized versions of images ($\theta_{m,\downarrow}, \beta_{m,\downarrow}$) are used to extract samples by excluding the original size. By adjusting the resolution, the behavior of the noise can be changed as it is mentioned in Section 3.1.5. The rest of the procedure applied identically and the structure of the algorithm is given in Algorithm 2.

Algorithm 2. The first level of transormation matrix learning by using downsampled images

---

Learning

Input: Set of Noisy Images (β), Set of Clean Images (θ)

Output: Transformation Matrix (T_σ)

for each β_m and θ_m **do**

  Down sample β_m and $\theta_m \rightarrow \beta_{m,\downarrow}$, $\theta_{m,\downarrow}$

  for each $\beta_{m,\downarrow}$, $\theta_{m,\downarrow}$ **do**

    Select N noisy patches randomly ($b_{N,i}$) (i: location)

    for each $b_{N,i}$ **do**

      Find *k-NN* patches

      Calculate $\omega_{N,LLE,\beta}$ and store in $\omega_{LLE,\beta}$

      Extract $a_{N,i}$

      Calculate $\omega_{N,LLE,\theta}$ and store in $\omega_{LLE,\theta}$

    end

  end

end

Calculate $T_\sigma \leftarrow \omega_{LLE,\beta}$, $\omega_{LLE,\theta}$

Testing

Input: Noisy Test Image (Y), T_σ

Output: Estimated Noise-Free Image ($\hat{X}$)

Extract all patches of Y (y)

for each y_i **do**

  Find *k-NN* patches

  Calculate ω_{LLE}

  Transform weights: $\omega_{LLE,T_\sigma} = T_\sigma . \omega_{LLE}$

  Calculate $\hat{x}_i = k\text{-}NN . \omega_{LLE,T_\sigma}$

end

Reconstruct $\hat{X} \leftarrow \hat{x}$

3.3 Preserving High Frequency Components

Image denoising algorithms try to find the best estimation of the noise-free image $\hat{X}$. To obtain $\hat{X}$, the effect of the noise on pixel intensity values should be diminished. However, by decreasing the effect of the noise, image denoising algorithms can waste the details and features of images, because features and noise are represented by high frequency components on the frequency spectrum of images. Therefore, the outcomes of any image denoising algorithm are smoothed images. According to the visual results of our approach given in Chapter 4, our approach can denoise image, but it is not successful to preserve the details and the features. For this reason, our approach needs to be developed on this subject of preserving features. There are three ways to be considered: 1) Feature mapping, 2) alpha rooting (Dabov et al., 2007c), 3) High Frequency Component Learning.

3.3.1 Feature Mapping

There are several methods to obtain the weights of images representing the significant and informative image content. According to the studies of Mertens, Kautz and van Reeth (2009) and Karakaya, Ulucan and Turkan (2022), PCA and Well-Exposedness methods can be employed.

3.3.1.1 Version 1

In this section, PCA mapping is utilized to estimate $\hat{x}_i$ patches. It is expected to be useful to preserve details, since it is Singular Value Decomposition (SVD) based method (Wold, Esbensen and Geladi, 1987). At the beginning, *k-NN* patches (D_i) of each y_i are extracted from all possible patches coming from Y. D_i is transformed by using the PCA coefficients of D_i. To ensure that all weights belong to atoms of transformed D_i are $\omega_i \in [0,1]$, transformed D_i should be normalized column-wise and row-wise respectively. All atoms of multiplication of D_i and normalized, transformed D_i (weights) are summed up to obtain $\hat{x}_i$. $\hat{X}$ is reconstructed by using equally weighted average of overlapping $\hat{x}_i$ patches.

3.3.1.2 Version 2

The main difference between Version 1 and 2 is that Well-Exposedness mapping is employed instead of PCA mapping. This mapping is indicating the intensity values or

brightness of pixels. Therefore, it could be beneficial to distinguish the details. For the second trial of Version 2, weights of Well-Exposedness reversed by extracting from one. Rest of the structure of Version 2 is preserved.

3.3.1.3 Version 3

In this section, both mapping methods are used. In the first trial, to find the transformed D_i, weights of the PCA and Well-Exposedness methods are multiplied. In the second trial, just reversed Well-Exposedness weights differs from the first trial. Moreover, two level process is applied for Trial 2. As the third trial, by considering the Level 2 results of Trial 2, iterative residual based dictionary updating approach is added and employed to find LLE weights for the reconstruction step.

3.3.2 Aplha Rooting

In this section, the core method is the alpha rooting approach to preserve the significant features in images by filtering out the noise (Dabov et al., 2007c). In the first trial, *k-NN* patches (D_i) of each noisy patch y_i are found from all possible patches of Y. Each 2D *k-NN* pathches with y_i are transformed by using DCT. To eliminate the DCT coefficients that represent the noisy particles, hard-thresholding is applied by considering the thresholding limit, and this hard-thresholding process is notated as Γ_λ where λ is the limit. As a next step, the alpha rooting method is applied to augment the thresholded non-zero DCT coefficients by considering the DC coefficient as it is given in Eq. (1). Hard-thresholded and sharpened via alpha rooting DCT coefficients are applied inverse DCT. $\omega_{i,LLE}$ is calculated by using the processed y_i and the processed D_i to find $\hat{x}_i$. To reconstruct $\hat{X}$, equally weighted average of overlapping $\hat{x}_i$ patches is used. In the second trial, a residual based iterative approach is employed to keep useful and important coefficients rather than hard-thresholding. When the *l2-norm* based residual between the transformed data including y_i and D_i is less than the stopping criteria defined as $n^2\sigma^2$, keeping the DCT coefficients should be stopped. $\omega_{i,LLE}$ calculation and the reconstruction process are applied same as in the first trial. In the third trial, only the stopping criteria differs from the Trial 2. It is defined as $n^2\sigma^2c^2$, where the c is a constant number (Dabov et al., 2007b). For the fourth trial, instead of an equally weighted average of overlapping $\hat{x}_i$ patches, the group variance based weighting is employed for the reconstruction step where the group contains y_i and

corresponding D_i. The other steps are used the same as Trial 2.

3.3.3 High Frequency Component Learning

According to experimental results of the methods reported in Sections 3.1 and 3.2, the pale side of the developed methods is preserving the details and features. In Sections 3.3.1 and 3.3.2, approaches preserving details are nested in the whole step. It means that steps of denoising and keeping features have been done at the same time. However, in this section, conservation of details and features is considered as a further and disjoint step of denoising . By using a basic learning algorithm similar to the algorithms defined in Sections 3.2.1 and 3.2.2, the transformation matrix providing possible details of images by processing noisy patches is determined to be found.

In the first trial, the learning algorithm explained in Section 3.2.1 is used to find a transformation matrix. However, as the set of noisy images (β), the results of Trial 2 of the algorithm mentioned in section 3.3.1.3 are considered (β_{proc}). To reveal the transformation matrix of high frequency components, a_{HF} pathces and b_{proc} pathces are extracted from the images containing details (high frequency components=$\theta_{i,HF}$) and pre-found outcomes (β_{proc}) respectively. Since, the processed images are denoised sufficiently successful but they are smoothed, the images containing details can be obtained as the difference between original images and processed (denoised) images such that $\theta_{i,HF} = \theta_i - \beta_{i,proc}$. The LLE weights calculation of each $\hat{x}_{i,HF}$ and the reconstruction of $\hat{X}_{HF}$ steps are used as they are defined in Section 3.2.1. $\hat{x}_{i,HF}$ and $\hat{X}_{HF}$ stand for patches and image containing details. At the end, final estimated noise-free image can be calculated as $\hat{X}_i = \beta_{i,proc} + \hat{X}_{HF}$.

In Trial 2, the main difference from the first trial is that 2D gradient of the processed images $\beta_{proc,grad}$ are employed. Therefore, a_{HF} patches are extracted from the details (θ_{HF}), b_{proc} patches are found from $\beta_{proc,grad}$ images. According to the results of Trial 1, the noisy images is not an efficient way to find the high frequency components. Therefore, to obtain the estimated details, the learning algorithm should be initialized by considering the details which are presented via gradient.

In the third trial, there is a method combination of the alpha rooting approach mentioned in Section 3.3.2 and the transformation matrix learning approach by using the multi-resolution images approach mentioned in Section 3.2.2. To summarize the procedure one more time to emphasize the differences; there are two sets of images: 1) clean (original) images (θ), 2) noisy images (β). N samples (b_N patches) are selected randomly from each downsampled noisy image $\beta_{i,\downarrow}$ by excluding the original size of the images. For each b_i, *k-NN* patches are extracted from all possible patches of $\beta_{i,\downarrow}$. Before finding $\omega_{N,LLE,\beta_{i,\downarrow}}$, the hard-thresholding and alpha rooting procedures are applied on $y_{i,\downarrow}$ and its *k-NN* patches. The calculated $\omega_{N,LLE,\beta_{i,\downarrow}}$ and the locations of each sample's *k-NN* patches are stored. N samples from the clean image (a_N patches), $\theta_{i,\downarrow}$ are selected by according to the stored locations of the noisy samples. For each a_N, *k-NN* pathces of corresponding b_N are used, and hard-thresholding approach and alpha rooting approach are employed on only the *k-NN* patches, because a_N patches are noise-free. All hard-thresholding and alpha rooting are done according to the algorithm explained in Section 3.3.2 Trial 2. By using $\omega_{N,LLE,\beta_{i,\downarrow}}$ and $\omega_{N,LLE,\theta_{i,\downarrow}}$, T_σ is calculated to apply for LLE weights of thresholded and alpha rooting applied patches and related *k-NN* patches of noisy test images in order to find estimated noise free patches. For the reconstruction step, the weighted average of overlapping estimated noise-free patches is employed.

CHAPTER 4: EXPERIMENTAL RESULTS & DISCUSSION

In this chapter, experimental results of each methodological approach are presented visually and statistically to evaluate the effects of the parameters and the performance of the algorithms. Moreover, visual and statistical results of several benchmark methods are shown for comparison such as BM3D (Dabov et al., 2007b), K-SVD (Elad and Aharon, 2006), WNNM (Gu et al., 2014), and Expected Patch Log-Likelihood (EPLL) (Zoran and Weiss, 2011). The quantitative results contain four statistics: 1) *Peak-Signal-to-Noise-Ratio* (PSNR), 2) *Structural Similarity Index Measurement (SSIM)*, 3) *Visual Information Fidelity* (VIF), and 4) *Information Fidelity Criterion* (IFC).

Our approach is developed on MATLAB R2022a. "Set12" dataset is used as original image set. These images are widely used greyscale images in image processing applications. All images are shown in Figure 4 with their names and sizes of images either *256x256* or *512x512.* The noisy images are derived from the Set12 images artifically to make controlled experiments. The standard deviation of the additive Gaussian white noise is defined $\sigma = 25$ with mean $\mu = 0$ for all noisy images.

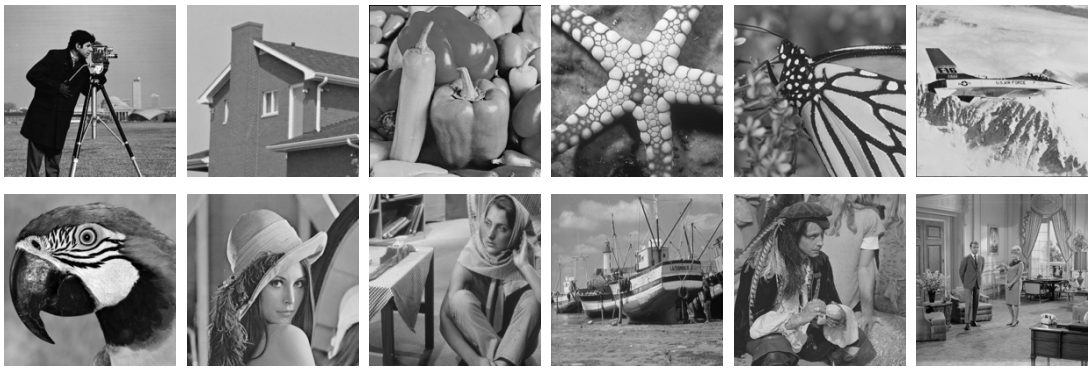


Figure 4. Images of Set12 Dataset.

4.1 Results of Traditional Approaches

In this section, visual and objective results of the methods described in Section 3.1 are presented. Also, the evaluation of parameters and algorithms can be observed by analyzing the experimental results.

4.1.1 Version 1

In order to make a feasebility study, a pixel-based approach is employed to reconstruct $\hat{X}$ by using calculated ω_{LLE} between y_i patches and corresponding *k-NN* patches. To determine the optimum parameter values such as n and t, several tests have been done by considering $n \in [5,7,9,13,15]$ and $t \in [3,5]$. According to the test results of Version 1 Trial 1, although the best results are reached with $n = 15$ and $t = 5$, these are not applicable values for parameters of patch size (n-by-n) and window size (tn-by-tn) respectively because of the computational cost and blur occurrence in denoised images. Therefore, the useful cases should consider $n \in [5,7,9]$ and $t \in [3]$. PSNR comparisons and visual comparisons of patch sizes are shown in Figure 5 and Figure 6 by considering the outcomes of Version 1 Trial 1.

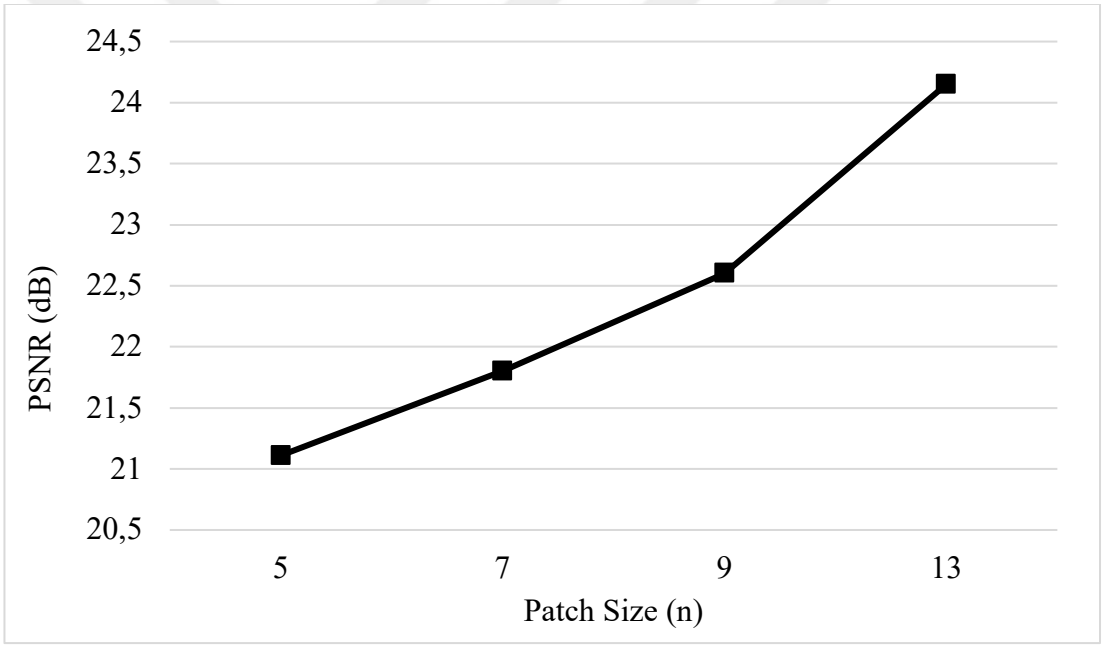


Figure 5. Patch size effect on PSNR by considering Version 1 Trial 1 in terms of overall PSNR.

a) $n = 5$ b) $n = 7$ c) $n = 9$ d) $n = 13$ e) $n = 15$

Figure 6. Patch size effect on blur by considering Version 1 Trial 1.

Moreover, the statistics of Version 1 Trial 1 are given in Table 1 for the parameters $n = 5$ and $k = 3$, which means window size is *15*-by-*15*.

Table 1. Statistical results of Version 1 Trial 1 including PSNR, SSIM, VIF and IFC of each image of Set12 (N-O: Noisy Vs. Original, E-O: Estimated Vs. Original).

Image	**PSNR N-O**	**PSNR E-O**	**SSIM N-O**	**SSIM E-O**	**VIF N-O**	**VIF E-O**	**IFC N-O**	**IFC E-O**
Cameraman	20,54	21,30	0,35	0,37	0,19	0,20	2,02	2,07
House	20,19	20,91	0,28	0,30	0,14	0,15	1,42	1,47
Peppers	20,28	21,12	0,36	0,38	0,18	0,20	1,92	1,99
Starfish	20,38	21,24	0,47	0,50	0,24	0,25	2,59	2,66
Butterfly	20,22	21,21	0,44	0,48	0,23	0,25	2,64	2,72
Jetplane	20,33	21,14	0,38	0,40	0,20	0,21	2,13	2,17
Parrot	20,59	21,42	0,39	0,42	0,22	0,23	2,36	2,45
Lena	20,24	20,93	0,60	0,62	0,13	0,13	1,36	1,41
Barbara	20,30	21,09	0,70	0,72	0,16	0,17	1,80	1,87
Boats	20,28	21,01	0,66	0,68	0,17	0,18	1,95	1,99
Man	20,25	20,94	0,67	0,69	0,17	0,18	1,86	1,91
Couple	20,27	20,99	0,69	0,71	0,19	0,20	2,10	2,14
Average	**20,32**	**21,11**	**0,50**	**0,52**	**0,18**	**0,20**	**2,01**	**2,07**

According to Table 1, although there is not significant improvement in statistics, the method introduced as Version 1 Trial 1 is important to determine suitable values of parameters. Also it is important to distunguish the parts of the algorithm that need to be improved. As the further trials are mentioned in Section 3.1.1, Trial 2 includes exploiting the similar patches, also Trail 3 has a PCA based dimension reduction approach while finding the similar pathces. The mentioned approach in Version 1 Trial 3 not only beneficial to decrease the computational cost but also useful to extract the similar patches that are filtered by keeping higher PCA coefficients. However, the question of how many coefficients are required needs to be answered to determine the optimal parameters's values for further versions of the algorithm. According to Figure 7, the effect of the dimension reduction size does not have a significant effect on the statistical results, especially for *n*=5 and *n*=7. Therefore, the dimension reduction size

may be selected as equals to *n* to keep computational cost low. Also, statistical results of Version 1 Trial 3 are shown in Table 2 by considering the most suitable values of parameters rather than the values providing the highest results.

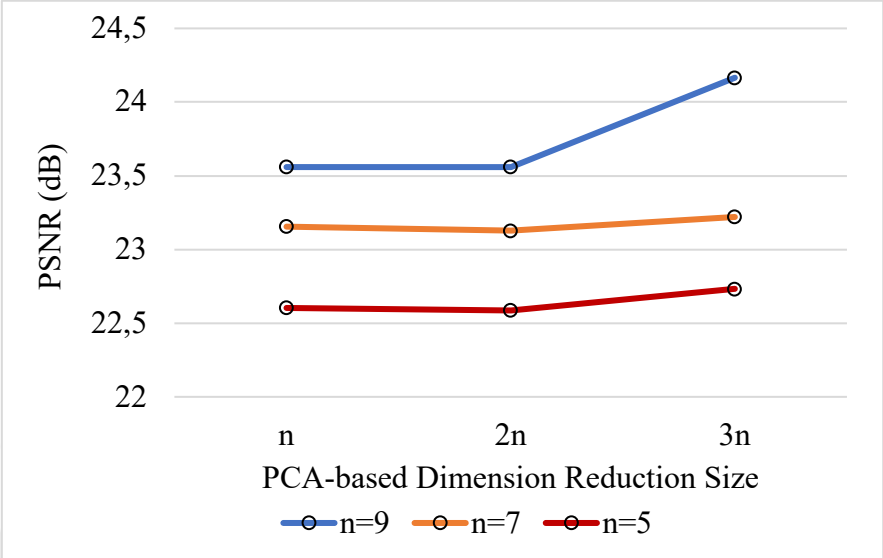


Figure 7. The effect of the number of the PCA coefficient for dimension reduction in terms of overall PSNR.

Table 2. Statistical results of Version 1 Trial 3 including PSNR, SSIM, VIF and IFC of each image of Set12 (*n*=5, dimension reduction size =*n*).

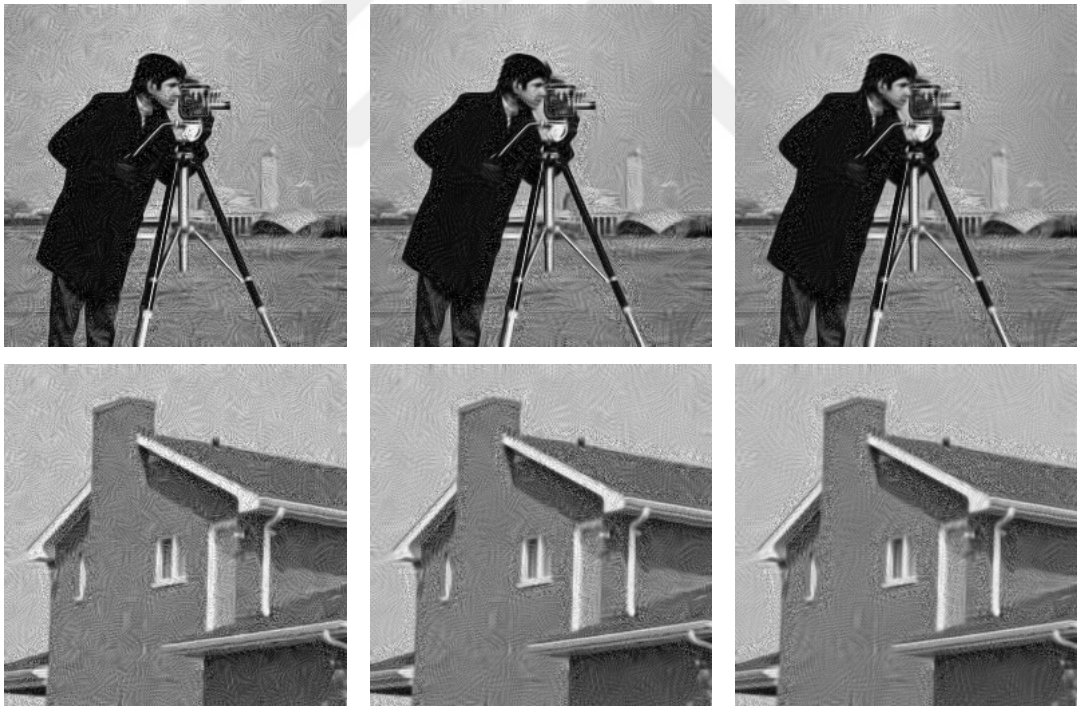
Image	**PSNR N-O**	**PSNR E-O**	**SSIM N-O**	**SSIM E-O**	**VIF N-O**	**VIF E-O**	**IFC N-O**	**IFC E-O**
Cameraman	20,54	22,56	0,35	0,40	0,19	0,21	2,02	1,92
House	20,19	22,91	0,28	0,36	0,14	0,17	1,42	1,47
Peppers	20,28	22,42	0,36	0,42	0,18	0,21	1,92	1,95
Starfish	20,38	22,45	0,47	0,53	0,24	0,25	2,59	2,49
Butterfly	20,22	22,15	0,44	0,51	0,23	0,25	2,64	2,60
Jetplane	20,33	22,07	0,38	0,42	0,20	0,20	2,13	1,91
Parrot	20,59	22,63	0,39	0,45	0,22	0,24	2,36	2,26
Lena	20,24	22,94	0,60	0,69	0,13	0,14	1,36	1,34
Barbara	20,30	22,87	0,70	0,77	0,16	0,19	1,80	1,84
Boats	20,28	22,78	0,66	0,73	0,17	0,19	1,95	1,82
Man	20,25	22,69	0,67	0,73	0,17	0,18	1,86	1,73
Couple	20,27	22,78	0,69	0,75	0,19	0,20	2,10	1,95
Average	**20,32**	**22,60**	**0,50**	**0,56**	**0,18**	**0,20**	**2,01**	**1,94**

4.1.2 Version 2

In this section, the results of the method explained in Section 3.1.2 are presented. The main development is that overlapping estimated pathces are used rather than estimated pixels for the reconstruction step. Moreover, the parameter about similarity of pathces is examined. This parameter can affect the dictionary size, since there is no a fixed size of dictionary. According to the optimal parameters, overall statistics of Set12 and some sample outcomes are given in Table 3 and Figure 8 respectively.

Table 3. Overall statistics of Version 2.

Patch Size	**PSNR N-O**	**PSNR E-O**	**SSIM N-O**	**SSIM E-O**	**VIF N-O**	**VIF E-O**	**IFC N-O**	**IFC E-O**
5	20,32	25,31	0,50	0,68	0,18	0,27	2,01	2,30
7	20,32	25,91	0,50	0,71	0,18	0,29	2,01	2,35
9	20,32	**26,20**	0,50	**0,72**	0,18	**0,30**	2,01	**2,37**



a) n=5 b) n=7 c)n=9

Figure 8. Samples outcomes (*Cameraman*, *House*) of Version 2 with different patch sizes.

As it is understood from the table and figure, although higher patch size value provides better statistics, blurring and computational cost causes problem. Therefore, the main

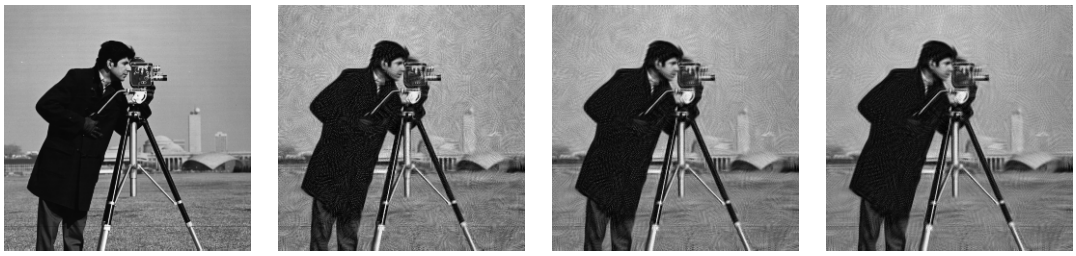
aim should be to diminish artifacts with lower patch size values.

4.1.3 Version 3

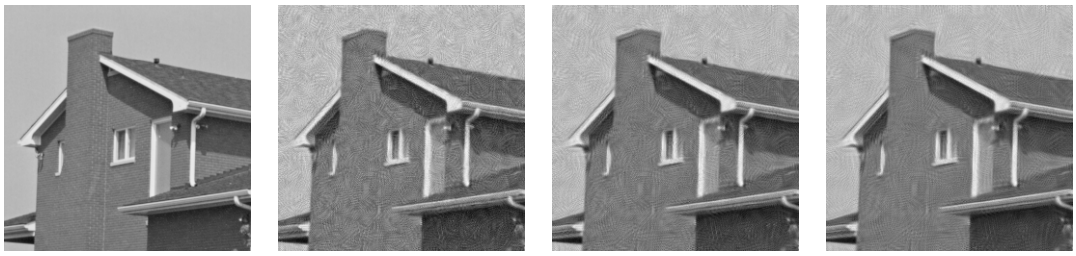
According to the methods that are explained in Section 3.1.3, the most innovator step is using iterative approach since it increases the sparsity of the representation of estimated noise-free patches ($\hat{x}_i$). The question should be answered that how many iterations are required to obtain the error below the pre-defined error boundary. As using this iterative approach, utilizing the atoms of the dictionary is also analyzed in terms of similarity of the pathces, the number of the atoms in the dictionary and uniqueness of the atoms. For these terms, several parameters are analyzed and tested. The performance of the algorithm is improved a little bit in terms of the statistics. However, with these parameters, all images can not be processed because of the lack of the sufficient number of atoms in the dictionary. Therefore, some of these concepts are eliminated such as uniqueness. The second important contribution in Version 3 is using the residual-based dictionary updating. Since it has successful effect on the statistics, it has combined with the methods that are using "at least" and "fixed" number of atoms cases by excluding the uniqueness of atoms. To reconstruct $\hat{X}$ weighted average of $\hat{x}_i$ patches are employed. According to the all tests, the most successful, applicable to all images, and promising approach is Version 3 Trial 6. The statistical results and some samples of the method are presented in Table 4 and Figure 9 respectively.

Table 4. Overall performance of Version 3 Trial 6.

Patch Size	**Window Size**	**Dimension Reduction Size**	**# atoms**	**# iterations**	**PSNR E-O**	**SSIM E-O**	**VIF E-O**	**IFC E-O**
5-by-5	15-by-15	5	8	8	26,7455	0,7213	0,2934	2,3682
5-by-5	15-by-15	5	16	16	26,7559	0,7225	0,2941	2,3554
7-by-7	21-by-21	7	8	8	27,4170	0,7635	0,3234	2,4414
7-by-7	21-by-21	7	16	16	27,5290	0,7665	0,3286	2,4479
9-by-9	27-by-27	9	8	8	27,6202	0,7855	0,3388	2,4646
9-by-9	27-by-27	9	16	16	**27,7949**	**0,7907**	**0,3478**	**2,4820**



a) Original b) n=5 c) n=7 d) n=9



e) Original f) n=5 g) n=7 h) n=9

Figure 9. Sample outcomes of Version 3 Trial 6 by using *Cameraman* and *House* images by considering patch sizes (n) are 5, 7 and 9, dimension reduction size equals to patch size (n), and the number of atoms and iterations are 16.

According to Table 4 and Figure 9, the performance of the propesed approach is improved significantly. Since the statistical results are very close to each other, computational cost should be considered primarily. However, the effect of the patch size on blurring can be observed by analyzing Figure 9. The bigger patch size prevents the block artifacts by blurring, but it fails to preserve the details although smoother images make feel more successful.

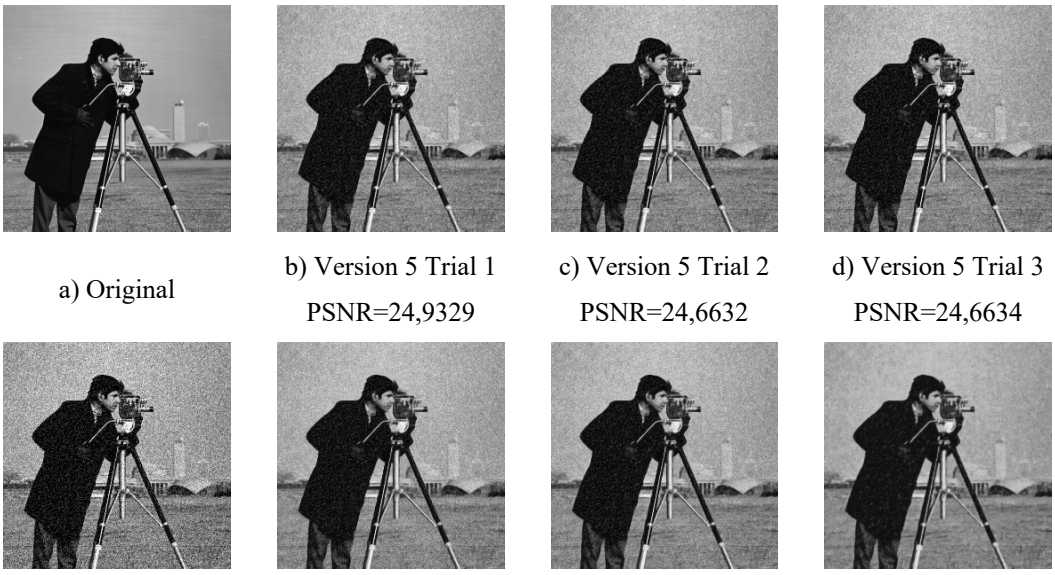
4.1.4 Version 4

Since the general idea is to optimize the weights of atoms to obtain estimated patches, L1-norm based approach is employed as it is explained in Section 3.1.4. By linear programming, two basic optimization approaches have been tested. However, none of them have promising results.

4.1.5 Version 5

This method is developed based on Version 3. According to the experience gained in Version 3, number of atoms in the dictionary is fixed. The first major difference is that patches are not extracted from corresponding search window, extracted from all image. The second major difference is using multi-resolution versions of the image to extract patches and their LLE weights. The noise behavior can change from coarse to fine due

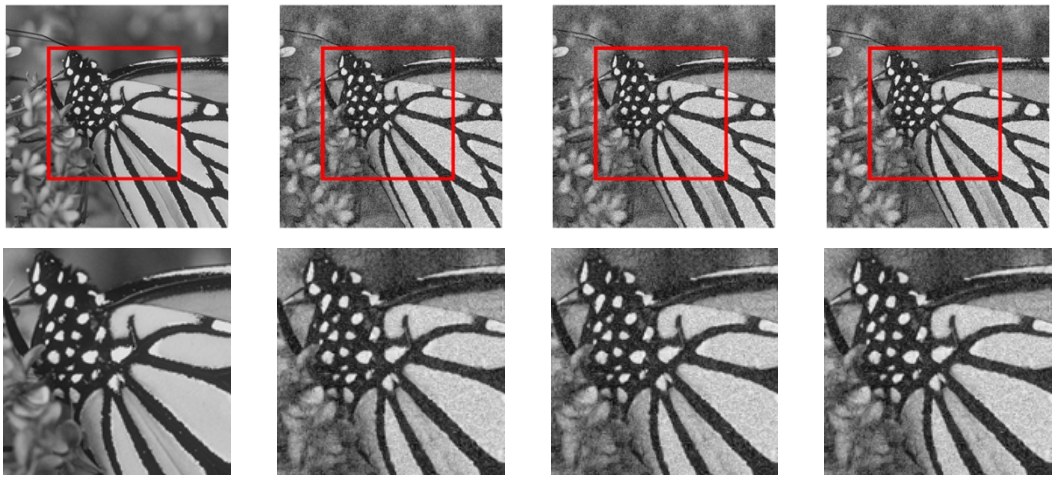
to downsampling. Additionally, it could have great benefit to increase sparsity because it seem like a filtering process before finding similar patches, thus it is expected to increase the success of finding similar patches. To understand how it performs, resizing is applied by including and excluding the original size. The reconstruction is done with and without weights to interpret the effect on blocking artifacts. Moreover, Gaussian pyramid is employed by including and excluding the original sizes of images to obtain multi-resolution images. It is expected to be more successful than the other resizing process, because there is a Gaussian filtering that can diminish the effect of the noise between each level of Gaussian pyramid. However, since each level of the pyramid has half size of the previous level, it can be disadvantageous to create sufficiently large dictionary. Also this kind of resizing can break the relevance between the patches extracted from original size images and resized images. To compare the performances of the trials, *Cameraman* images are given in Figure 10 with fixed eight atoms and n=5. As it is seen, visually satisfying results could not be obtained. For example, although the outcome of Version 5 Trial 6 has highest PSNR (26,2801 dB), details of the image are not preserved very successfully. To see Version 5 noise removing and detail preserving performance from a closer perspective, Figure 11 is given by condisering the *Butterfly* image.



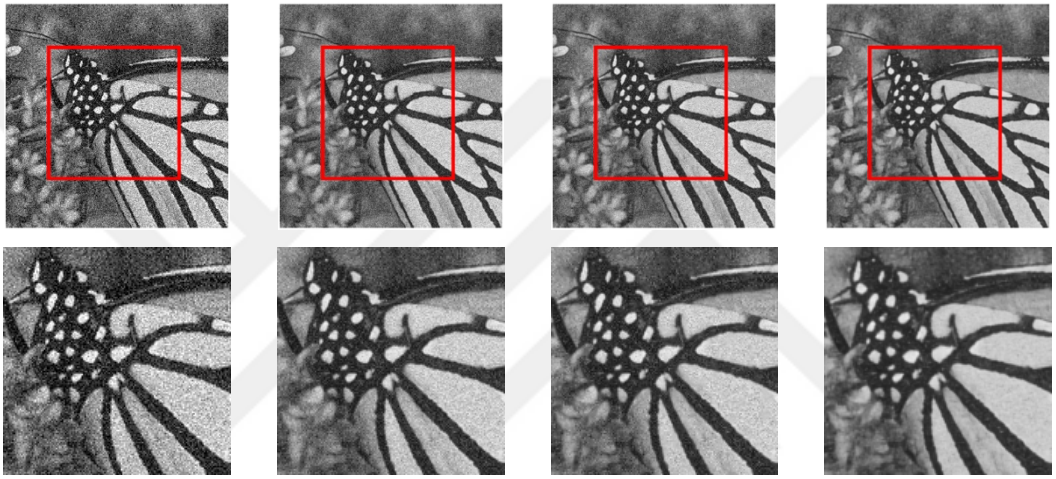
a) Original | b) Version 5 Trial 1 PSNR=24,9329 | c) Version 5 Trial 2 PSNR=24,6632 | d) Version 5 Trial 3 PSNR=24,6634

e) Noisy (σ=25) PSNR=20,5396 | f) Version 5 Trial 4 PSNR=23,9891 | g) Version 5 Trial 5 PSNR=24,6217 | h) Version 5 Trial 6 PSNR=26,2801

Figure 10. Sample outputs of *Cameraman* processed by using Version 5 trials.



a) Original b) Trial 1 c) Trial 2 d) Trial 3



e) Noisy (σ = 25) f) Trial 4 g) Trial 5 h) Trial 6

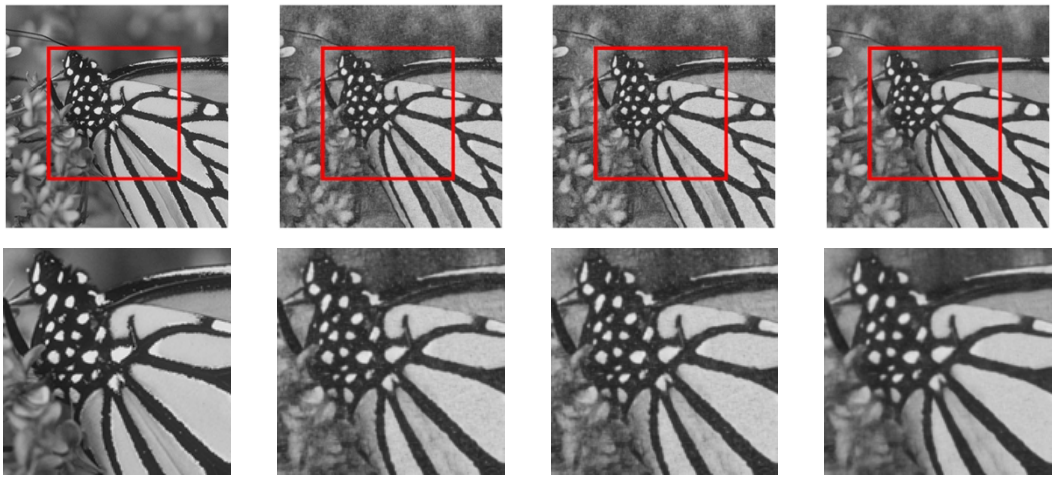
Figure 11. Sample outputs of *Butterfly* processed by using Version 5 trials.

In addition, overall statistics of each trial of Version 5 are given in Table 5 by considering patch size as 5 and number of the atoms as 8. When Table 5 is considered, it is obvious that Trial 6 which is employing Gaussian pyramid by including the original image has the brightest results. Besides the pyramids, rest of the trial results are close to each other. In the first trial, the coefficients of resizing are [0.90, 0.80, 0.75, 0.50, 0.25] which means the original size is excluded, and number of atoms (number of selected k-NN patches) is fixed as 8 and 16. In Trial 2, the coefficients of resizing are [1.00, 0.90, 0.80, 0.75, 0.50, 0.25] which means the original size is used, and for the reconstruction step, equally weighted average of overlapping estimated patches. The main difference between Trial 2 and Trial 3 is that Trial 3 employs weighted recontruction step. In Trial 4, only the original size is used by considering weighted reconstruction step. Between all these trials, there is not a significant improvement. Nevertheless, to see the effect of these approaches on details of images,

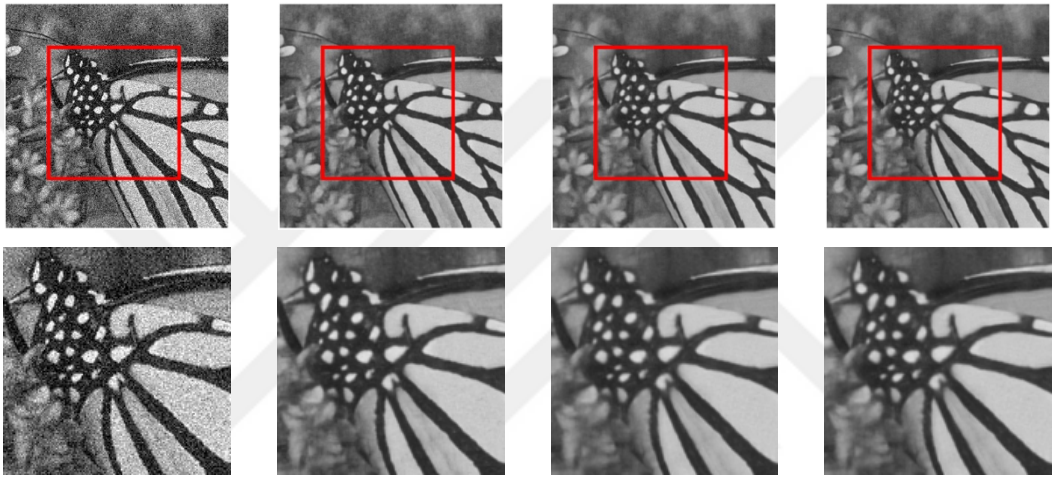
sample outcomes are presented in Figure 12.

Table 5. Overall statistics of Version 5 (number of atoms =8, n = 5).

Trial	n	# atoms	PSNR N-O	PSNR E-O	SSIM N-O	SSIM E-O	VIF N-O	VIF E-O	IFC N-O	IFC E-O
Trial 1	5	8	20,3237	24,8597	0,4986	0,6576	0,1845	0,2402	2,0129	2,2279
Trial 2	5	8	20,3237	24,5691	0,4986	0,6466	0,1845	0,2381	2,0129	2,2249
Trial 3	5	8	20,3237	24,5686	0,4986	0,6466	0,1845	0,2381	2,0129	2,2248
Trial 4	5	8	20,3237	23,8395	0,4986	0,6237	0,1845	0,2431	2,0129	2,2357
Trial 5	5	8	20,3237	24,5890	0,4986	0,6515	0,1845	0,2512	2,0129	2,2761
Trial 6	5	8	20,3237	**26,6965**	0,4986	**0,7614**	0,1845	**0,2825**	2,0129	**2,2953**



a) Original b) Test 1 c) Test 2 d) Test 3



e) Noisy (σ = 25) f) Test 4 g) Test 5 h) Test 6

Figure 12. *Butterfly* image outcome samples of Version 5 Trial 6. Blurring effect (loosing details) increases with the increasing patch size.

The description of the parameters of the trials mentioned in Figure 12 is; the patch size (n) is 5, and the number of atoms (number of k-NN) is fixed as 8 for the first two tests. The difference between Test 1 and Test 2 is the number of levels of Gaussian pyramid. By excluding the Level 0 (original size), Level 1 and 2 are used for Test 1, and Level 1, 2 and 3 are used for Test 2. The same pattern is followed for other patch size values which are 7 and 9. All test results are reported in Table 6. According to the table, the patch size does not affect considerably. Although the only and major difference between Trial 5 and Trial 6 is whether exploiting Level 0 or not, Trial 5 does not have a good performance as much as Trial 6 has.

Table 6. Test performances of Version 5 Trial 6.

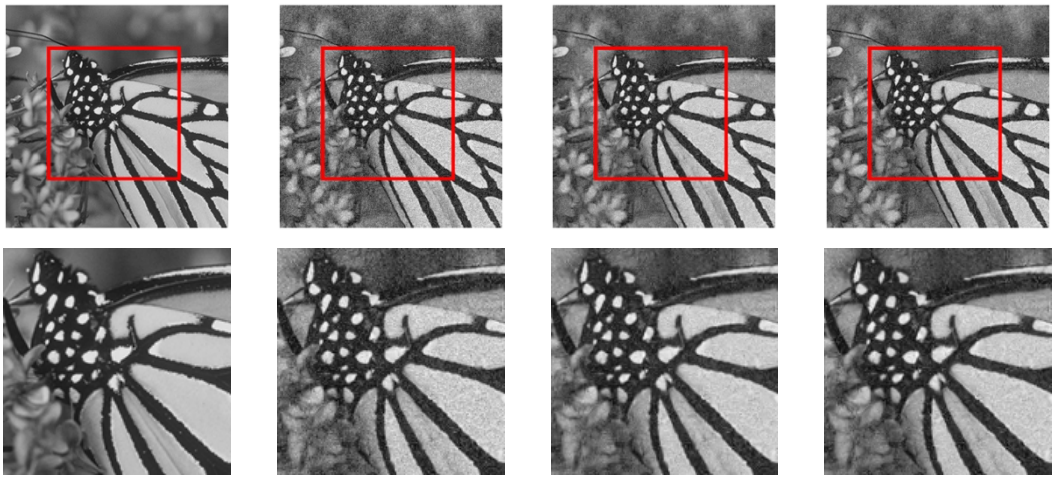
# Test	n	Levels	PSNR		SSIM		VIF		IFC	
			N-O	E-O	N-O	E-O	N-O	E-O	N-O	E-O
1	5	1,2	20,32	26,70	0,50	0,76	0,18	0,28	2,01	2,30
2	5	1,2,3	20,32	26,68	0,50	0,76	0,18	0,28	2,01	2,29
3	7	1,2	20,32	**27,22**	0,50	**0,81**	0,18	**0,33**	2,01	**2,32**
4	7	1,2,3	20,32	**27,22**	0,50	**0,81**	0,18	**0,33**	2,01	**2,32**
5	9	1,2	20,32	26,78	0,50	0,82	0,18	0,36	2,01	2,22
6	9	1,2,3	20,32	26,80	0,50	0,82	0,18	0,36	2,01	2,22

4.1.6 Version 6

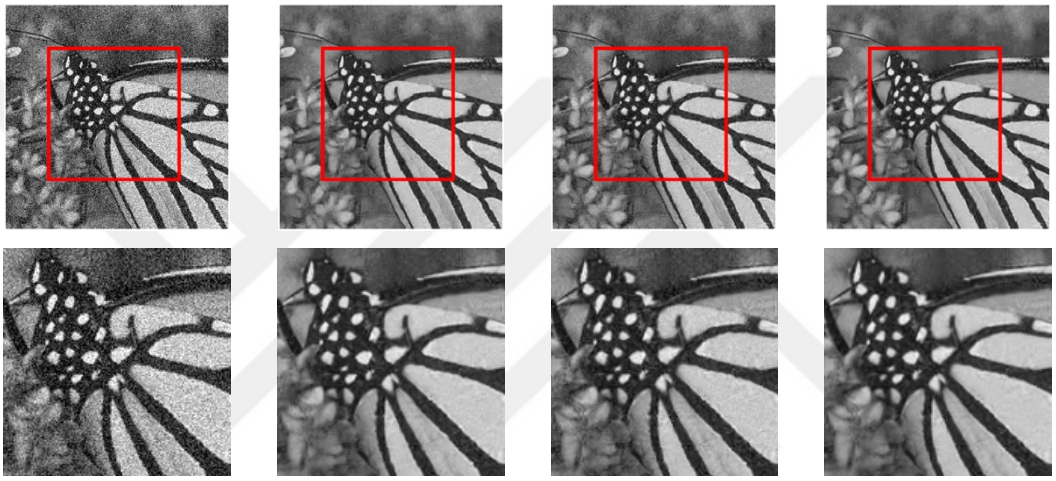
As it is explained in Section 3.1.6, a shrinkage method (hard-thresholding) is applied on DCT transform of each *k-NN* patches where λ is the hard-threshold limit. Also, by including the original size, the multi-resolution approach is used with the resizing coefficients [1.00, 0.90, 0.80, 0.75, 0.50, 0.25]. The thresholding limit as determined as 2 (Dabov et al., 2007b) and σ. The number of *k-NN* patches is fixed as 8, and patch sizes (n) are 5, 7 and 9. According to Table 7 and Figure 13, λ has significant effect on the performance of the algorithm. With the optimum value of λ, performance can be maximized. By considering the results, there is no sufficient and satisfying improvement on statistics and also visual quality of outcomes. Increasing the patch size is not a suitable solution to filter out noise by keeping details as it can be seen in Figure 13. It is obvious especially by focusing on the are intersecting dark and light parts of the wing.

Table 7. Statistical performance of Version 6.

n	λ	PSNR		SSIM		VIF		IFC	
		N-O	E-O	N-O	E-O	N-O	E-O	N-O	E-O
5	2	20,32	24,57	0,50	0,65	0,18	0,24	2,01	2,22
5	σ	20,32	26,29	0,50	0,73	0,18	0,27	2,01	2,32
7	2	20,32	26,21	0,50	0,72	0,18	0,26	2,01	2,27
7	σ	20,32	27,64	0,50	0,80	0,18	0,31	2,01	2,35
9	2	20,32	26,92	0,50	0,75	0,18	0,28	2,01	2,27
9	σ	20,32	**28,05**	0,50	**0,83**	0,18	**0,33**	2,01	**2,33**



a) Original b) n=5, λ=2 c) n=5, λ= σ d) n=7, λ=2



e) Noisy (σ = 25) f) n=7, λ= σ g) n=9, λ=2 h) n=, λ=σ

Figure 13. Outcome samples of Version 6.

4.2 Results of Transformation Matrix Learning Approach

According to all the given statistics and visual outcome images in Section 4.1, the methods that are employed by condsidering the reference studies, desired and expected results could not be achieved. Thus, to use LLE-based basic learning algorithm can be more reasonable and effective rather than the traditional methods. As a consequence, in this section, performance of the algorithms explained in Sections 3.2.1 and 3.2.2 are expressed to interpret the behavior of the algorithm for further improvements.

4.2.1 Results of Learning Dictionary With Original Sized Images

The details of the learning algorithm is explained in Section 3.2.1. According to the procedure, there are several parameters critical to analyze the performance of the algorithm. These parameters are n, N and k for k-NN patches. Two n values are

accepted which are 7 and 9. The value of k representing the number of the selected nearest neighbor patches, and it is fixed to k=8, 16, 24. The parameter that is expected to make difference for learning algorithm is N where it is the number of sample patches selected randomly. All patches of images are not used for purpose of learning to avoid overfitting problem. Since LLE weights of each sample patch are stacked to calculate T, N can affect the results directly. To understand the impact of N, several values of N are used in a large scale. Thus, N is determined as [250,1000,5000,10000]. After obtaining the T, it is applied for all patches of the noisy images for testing. Moreover, the algorithm is repeated itself to increase the level of the process. Applying the same procedure on processed images should increase success of the algorithm. By using these values of three parameters the performance of the algorithm that exploits the original sized images, statistics of each combination of parameters are given in Table 8, 9, 10, and 11

Table 8. Overall PSNR performance of the transformation matrix learning approach using only original sized images.

n	***k***	***N***	**PSNR N-O**	**PSNR E-O**				
				Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**
7	**8**	**250**	20,3237	26,6502	28,0104	28,1525	28,0748	27,9924
7	**8**	**1000**	20,3237	26,6518	28,0071	28,1198	28,0428	27,9496
7	**8**	**5000**	20,3237	26,6511	28,0180	28,1319	28,0603	27,972
7	**8**	**10000**	20,3237	26,6500	28,0113	28,1298	28,0519	27,9582
7	**16**	**250**	20,3237	26,7859	27,8038	27,8431	27,7360	27,6125
7	**16**	**1000**	20,3237	26,7866	27,8159	27,8416	27,7378	27,6277
7	**16**	**5000**	20,3237	26,7867	27,8196	27,8544	27,7583	27,6491
7	**16**	**10000**	20,3237	26,7859	27,8111	27,8462	27,7490	27,6379
7	**24**	**250**	20,3237	26,8667	27,5255	27,4073	27,2274	27,0558
7	**24**	**1000**	20,3237	26,86039	27,5349	27,3914	27,1825	26,9995
7	**24**	**5000**	20,3237	26,8639	27,5270	27,3882	27,2034	27,0431
7	**24**	**10000**	20,3237	26,8642	27,5176	27,3820	27,1841	27,0039
9	**8**	**250**	20,3237	27,2805	28,1025	28,1010	27,9952	27,8908
9	**8**	**1000**	20,3237	27,2792	28,0992	28,1291	28,0242	27,9146
9	**8**	**5000**	20,3237	27,2790	28,1057	28,1210	28,0099	27,8995
9	**8**	**10000**	20,3237	27,2788	28,1027	28,1232	28,0106	27,9033

Table 8 (continued). Overall PSNR performance of the transformation matrix learning approach using only original sized images.

9	**16**	**250**	20,3237	27,2502	28,0389	28,1443	28,0629	27,9967
9	**16**	**1000**	20,3237	27,2398	28,0277	28,1417	28,0410	27,9190
9	**16**	**5000**	20,3237	27,2459	28,0343	28,1417	28,0768	28,0140
9	**16**	**10000**	20,3237	**27,2465**	**28,0312**	**28,1446**	**28,0793**	**28,0186**
9	**24**	**250**	20,3237	27,1732	27,8320	27,8642	27,7645	27,6850
9	**24**	**1000**	20,3237	27,1710	27,8358	27,8975	27,7933	27,7317
9	**24**	**5000**	20,3237	27,1626	27,8151	27,8679	27,7305	27,6042
9	**24**	**10000**	20,3237	27,1679	27,8159	27,8630	27,7307	27,6054

Table 9. Overall SSIM performance of the transformation matrix learning approach using only original sized images.

n	***k***	***N***	**SSIM N-O**	**SSIM E-O**				
				Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**
7	**8**	**250**	0,4986	0,7180	0,8172	0,8427	0,8469	0,8466
7	**8**	**1000**	0,4986	0,7180	0,8172	0,8425	0,8464	0,8459
7	**8**	**5000**	0,4986	0,7180	0,8172	0,8426	0,8467	0,8463
7	**8**	**10000**	0,4986	0,7179	0,8172	0,8426	0,8466	0,8460
7	**16**	**250**	0,4986	0,7299	0,8209	0,8404	0,8423	0,8406
7	**16**	**1000**	0,4986	0,7300	0,8210	0,8404	0,8423	0,8408
7	**16**	**5000**	0,4986	0,7302	0,8209	0,8405	0,8426	0,8412
7	**16**	**10000**	0,4986	0,7300	0,8209	0,8404	0,8425	0,8410
7	**24**	**250**	0,4986	0,7405	0,8214	0,8342	0,8332	0,8297
7	**24**	**1000**	0,4986	0,7403	0,8213	0,8341	0,8324	0,8286
7	**24**	**5000**	0,4986	0,7406	0,8213	0,8340	0,8327	0,8295
7	**24**	**10000**	0,4986	0,7407	0,8213	0,8339	0,8324	0,8286
9	**8**	**250**	0,4986	0,7551	0,8348	0,8453	0,8447	0,8433
9	**8**	**1000**	0,4986	0,7552	0,8346	0,8457	0,8452	0,8437
9	**8**	**5000**	0,4986	0,7552	0,8347	0,845527	0,8449	0,8433
9	**8**	**10000**	0,4986	0,7551	0,8346	0,8455	0,8449	0,8434
9	**16**	**250**	0,4986	0,7613	0,8365	0,8470	0,8467	0,8460
9	**16**	**1000**	0,4986	0,7615	0,8363	0,8470	0,8464	0,8445
9	**16**	**5000**	0,4986	0,7615	0,8363	0,8469	0,8469	0,8463
9	**16**	**10000**	0,4986	**0,7614**	**0,8363**	**0,8470**	**0,8470**	**0,8464**

Table 9 (continued). Overall SSIM performance of the transformation matrix learning approach using only original sized images.

9	**24**	**250**	0,4986	0,7644	0,8347	0,8429	0,8419	0,8406
9	**24**	**1000**	0,4986	0,7647	0,8346	0,8434	0,8424	0,8416
9	**24**	**5000**	0,4986	0,7647	0,8344	0,8430	0,8414	0,8392
9	**24**	**10000**	0,4986	0,7647	0,8344	0,8428	0,8413	0,8391

Table 10. Overall VIF performance of the transformation matrix learning approach using only original sized images.

n	***k***	***N***	**VIF N-O**	**VIF E-O**				
				Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**
7	**8**	**250**	0,1845	0,3077	0,3848	0,4211	0,4352	0,4387
7	**8**	**1000**	0,1845	0,3078	0,3848	0,4214	0,4353	0,4391
7	**8**	**5000**	0,1845	0,3077	0,3848	0,4215	0,4354	0,4392
7	**8**	**10000**	0,1845	0,3077	0,3849	0,4215	0,4356	0,4394
7	**16**	**250**	0,1845	0,3162	0,3932	0,4266	0,4381	0,4404
7	**16**	**1000**	0,1845	0,3163	0,3933	0,4270	0,4383	0,4405
7	**16**	**5000**	0,1845	0,3164	0,3932	0,4270	0,4384	0,4408
7	**16**	**10000**	0,1845	0,3163	0,3932	0,4269	0,4382	0,4406
7	**24**	**250**	0,1845	0,3237	0,3976	0,4262	0,4337	0,4333
7	**24**	**1000**	0,1845	0,3236	0,3973	0,4265	0,4342	0,4336
7	**24**	**5000**	0,1845	0,3238	0,3976	0,4264	0,4341	0,4337
7	**24**	**10000**	0,1845	0,3238	0,3976	0,4265	0,4339	0,4333
9	**8**	**250**	0,1845	0,3389	0,4237	0,4498	0,4513	0,4461
9	**8**	**1000**	0,1845	0,3389	0,4235	0,4493	0,4508	0,4455
9	**8**	**5000**	0,1845	0,3389	0,4236	0,4496	0,4516	0,4468
9	**8**	**10000**	0,1845	0,3389	0,4236	0,4495	0,4514	0,4462
9	**16**	**250**	0,1845	0,3458	0,4322	0,4555	0,4562	0,4512
9	**16**	**1000**	0,1845	0,3460	0,4320	0,4557	0,4570	0,4536
9	**16**	**5000**	0,1845	0,3460	0,4320	0,4558	0,4565	0,4516
9	**16**	**10000**	0,1845	**0,3460**	**0,4320**	**0,4557**	**0,4564**	**0,4515**
9	**24**	**250**	0,1845	0,3496	0,4351	0,4571	0,4568	0,4517
9	**24**	**1000**	0,1845	0,3499	0,4349	0,4569	0,4566	0,4510
9	**24**	**5000**	0,1845	0,3498	0,4349	0,4570	0,4571	0,4527
9	**24**	**10000**	0,1845	0,3498	0,4350	0,4571	0,4570	0,4525

Table 11. Overall IFC performance of the transformation matrix learning approach using only original sized images.

n	*k*	*N*	IFC N-O	IFC E-O				
				Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**
7	**8**	**250**	2,0129	2,4163	2,4419	2,3526	2,2436	2,1720
7	**8**	**1000**	2,0129	2,4165	2,4411	2,3438	2,2356	2,1640
7	**8**	**5000**	2,0129	2,4166	2,4436	2,3466	2,2392	2,1688
7	**8**	**10000**	2,0129	2,4166	2,4420	2,3463	2,2377	2,1654
7	**16**	**250**	2,0129	2,4093	2,4062	2,2996	2,1835	2,1050
7	**16**	**1000**	2,0129	2,4088	2,4090	2,2985	2,1829	2,1070
7	**16**	**5000**	2,0129	2,4083	2,4104	2,3025	2,1895	2,1145
7	**16**	**10000**	2,0129	2,4086	2,4082	2,3005	2,1868	2,1105
7	**24**	**250**	2,0129	2,3963	2,3522	2,2129	2,0816	1,9904
7	**24**	**1000**	2,0129	2,3954	2,3559	2,2121	2,0751	1,9831
7	**24**	**5000**	2,0129	2,3954	2,3545	2,2100	2,0778	1,9905
7	**24**	**10000**	2,0129	2,3952	2,3516	2,2095	2,0736	1,9815
9	**8**	**250**	2,0129	2,4571	2,4295	2,2980	2,2032	2,1441
9	**8**	**1000**	2,0129	2,4565	2,4301	2,3037	2,2085	2,1470
9	**8**	**5000**	2,0129	2,4565	2,4321	2,3031	2,2062	2,1448
9	**8**	**10000**	2,0129	2,4565	2,4314	2,3034	2,2059	2,1449
9	**16**	**250**	2,0129	2,4468	2,4270	2,3008	2,2073	2,1533
9	**16**	**1000**	2,0129	2,4438	2,4276	2,3033	2,2052	2,1432
9	**16**	**5000**	2,0129	2,4453	2,4289	2,3037	2,2117	2,1580
9	**16**	**10000**	2,0129	**2,4455**	**2,4282**	**2,3044**	**2,2126**	**2,1602**
9	**24**	**250**	2,0129	2,4292	2,3985	2,2553	2,1552	2,0974
9	**24**	**1000**	2,0129	2,4273	2,3998	2,2615	2,1594	2,1049
9	**24**	**5000**	2,0129	2,4256	2,3978	2,2594	2,1513	2,0856
9	**24**	**10000**	2,0129	2,4270	2,3977	2,2570	2,1493	2,0837

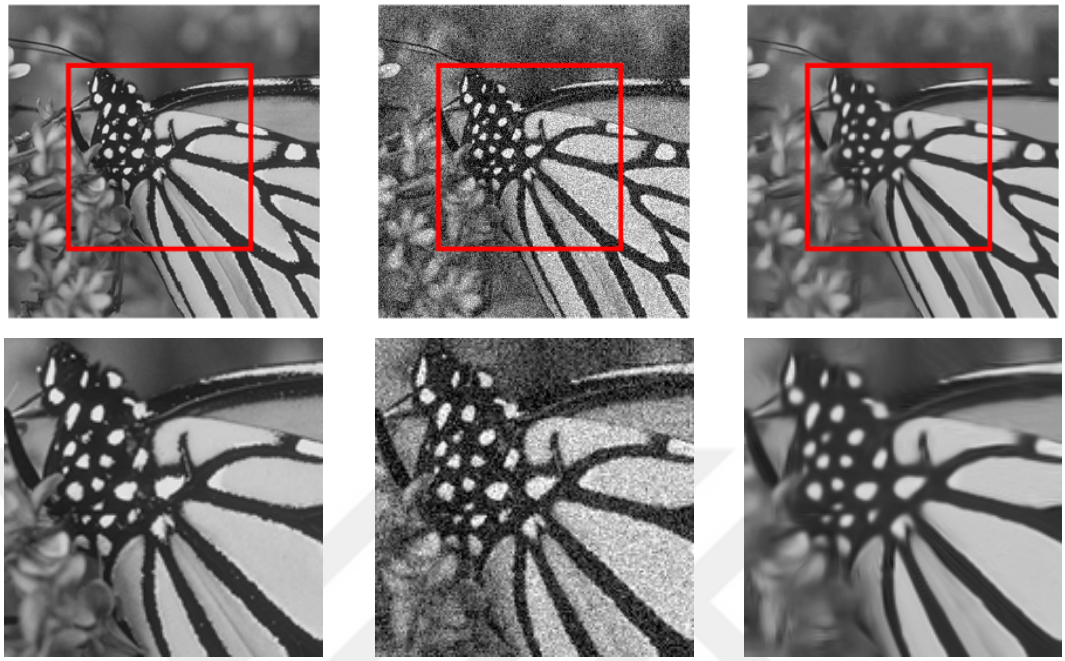
In addition to the statistical results, to interpret on the effects of the level of progress, number of *k-NN* patches, and patch size, visual outcomes and explainations are presented in Figure 14, 15, and 16. . Moreover, to focus on the details, sample outcomes are presented as zoomed in, since detail preserving is important as much as removing the noise.

Figure 14. Visual outcomes of the approach described as transformation matrix learning by using only original sized images with the parameters n=9, k=16, and N=10000. a) Original Image (X) b) Noisy Image (Y) (σ=25) c) Estimated Noise-Free Image ($\hat{X}$) of Level 1 d) Level 2 $\hat{X}$ e) Level 3 $\hat{X}$ f) Level 4 $\hat{X}$ g) Level 5 $\hat{X}$

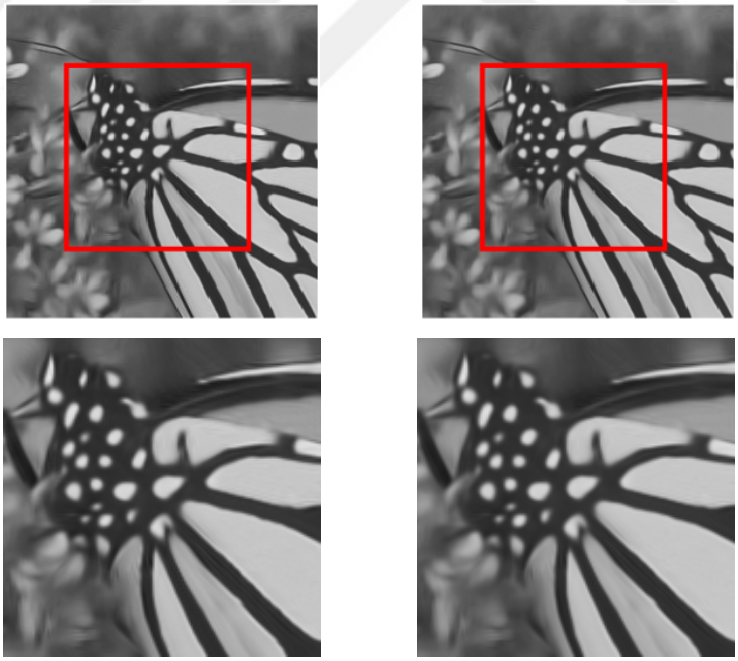
Figure 14 (continued). Visual outcomes of the approach described as transformation matrix learning by using only original sized images with the parameters n=9, k=16, and N=10000. a) Original Image (X) b) Noisy Image (Y) (σ=25) c) Estimated Noise-Free Image ($\hat{X}$) of Level 1 d) Level 2 $\hat{X}$ e) Level 3 $\hat{X}$ f) Level 4 $\hat{X}$ g) Level 5 $\hat{X}$

According to the methodology of an iterative procedure with levels, the processed image does not perform as it is supposed to work at some levels. The statistics are inclined to settle down, even to be worse. Therefore, to compare the effect of other parameters, Level 3 outcomes can be considered according to the given statistics. To investigate the visual quality of outcomes of all levels and differences between levels, example images are in given in Figure 14 for the parameters n=9, k=16, and N=10000. As it is seen from Figure 15 and the statistics, k does not have significant importance in terms of removing noise and keeping details. In addition to that, Figure 16 is given to interpret about the effect of the patch size. According to this figure, although noise removing performance of bigger patch size is better statistically, it fails to keep details and introduces blur. Also the parameter N can be important for the performance of the

algorithm. Therefore, it is worth to study on N especially for the ramdomly selection process.

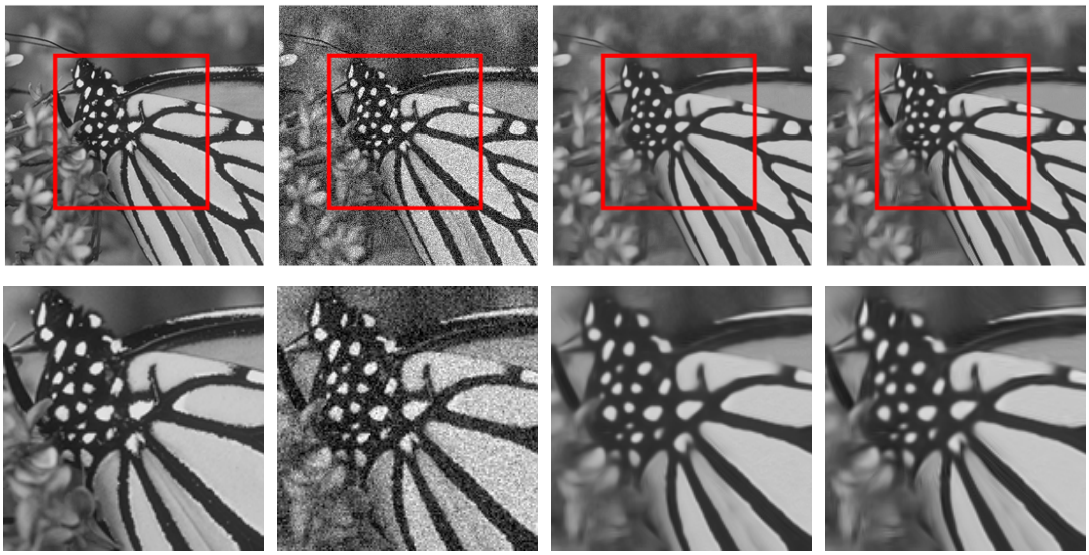


a) X b) Y (σ=25) c) $\hat{X}$ (k=8)



d) $\hat{X}$ (k=16) e) $\hat{X}$ (k=24)

Figure 15. The effect of k on *Butterfly* with parameters n=9 and N=10000 (Level 3).



a) X b) Y (σ=25) c) $\hat{X}$ (n=7) d) $\hat{X}$ (n=9)

Figure 16. The effect of the patch size with parameters k=16 and N=10000 (Level 3).

4.2.2 Results of Creating Dictionary With Resized Images

As it is explained in Section 3.2.2, there is a little difference between Algorithm 1 and Algorithm 2 that are mentioned in Section 3.2.1 and 3.2.2. The difference is that Algorithm 2 uses resized versions of both noisy and clean images, without original images. Therefore, in this section, rather than focusing on analysis of impacts of the parameters, the effect of the mentioned difference is emphasized. To compare the reults visually, the parameters are accepted as n=9, k=16, N=10000, and level 3. In addition to that, resizing coefficient are [0.90, 0.80, 0.75, 0.50, 0.25]. Nevertheless, all statistical results of this algorithm are given in Table 12, 13, 14 and 15 by including all combinations of the paramaters. Furthermore, to distinguish the effect of number of levels and to compare two approaches of transformation matrix learning method, Figure 17 and 18 are presented respectively. The impact of the number of levels can be noticed easily by checking Figure 17. Based on the experience gained from the method that uses only original sized images, this approach is tried up to level 3 because it is expected that all statistics settle down after that level, thus level 3 results are the highest results based on Table 12, 13, 14 and 15. According to the these tables and figures, the performance of Algorithm 2 by considering the parameters n=9, k=16, N=10000, and level 3 is better than the method employing Algorithm 1 with a little difference. This difference is not clear visually as it can be seen in Figure 18. Also, Figure 19 is presented to understand the impact of the algorithm and patch size on

keeping details of images. As it can be observed, noise removing performance is improved significantly.

Table 12. Overall performance of the transformation matrix learning approach by using multi-resolution images in terms of PSNR statistics.

n	***k***	***N***	**PSNR N-O**	**PSNR E-O**		
				Level 1	**Level 2**	**Level 3**
7	**8**	**250**	20,3237	26,5414	28,1635	28,3106
7	**8**	**1000**	20,3237	26,5415	28,1619	28,3121
7	**8**	**5000**	20,3237	26,5412	28,1621	28,3106
7	**8**	**10000**	20,3237	26,5414	28,1620	28,3106
7	**16**	**250**	20,3237	26,5457	28,1374	28,3151
7	**16**	**1000**	20,3237	26,5473	28,1376	28,3084
7	**16**	**5000**	20,3237	26,5491	28,1377	28,3087
7	**16**	**10000**	20,3237	26,5485	28,1375	28,3093
7	**24**	**250**	20,3237	26,5333	27,9915	28,0697
7	**24**	**1000**	20,3237	26,5362	27,9922	28,0773
7	**24**	**5000**	20,3237	26,5362	27,9915	28,0772
7	**24**	**10000**	20,3237	26,5365	27,9915	28,0767
9	**8**	**250**	20,3237	27,3192	28,2420	28,2071
9	**8**	**1000**	20,3237	27,3194	28,2411	28,2059
9	**8**	**5000**	20,3237	27,3190	28,2399	28,2041
9	**8**	**10000**	20,3237	27,3190	28,2400	28,2051
9	**16**	**250**	20,3237	27,3382	28,3021	28,3569
9	**16**	**1000**	20,3237	27,3380	28,3002	28,3612
9	**16**	**5000**	20,3237	27,3377	28,2992	28,3594
9	**16**	**10000**	20,3237	**27,3380**	**28,3002**	**28,3605**
9	**24**	**250**	20,3237	27,2909	28,2254	28,2634
9	**24**	**1000**	20,3237	27,2908	28,2186	28,2649
9	**24**	**5000**	20,3237	27,2905	28,2196	28,2674
9	**24**	**10000**	20,3237	27,2910	28,2208	28,2695

Table 13. Overall performance of the transformation matrix learning approach by using multi-resolution images in terms of SSIM statistics.

n	*k*	*N*	SSIM N-O	SSIM E-O		
				Level 1	Level 2	Level 3
7	**8**	**250**	0,4986	0,71025	0,8137	0,8435
7	**8**	**1000**	0,4986	0,71036	0,8137	0,8435
7	**8**	**5000**	0,4986	0,71024	0,8137	0,8434
7	**8**	**10000**	0,4986	0,7106	0,8137	0,8434
7	**16**	**250**	0,4986	0,7115	0,8141	0,8440
7	**16**	**1000**	0,4986	0,7116	0,8141	0,8440
7	**16**	**5000**	0,4986	0,7117	0,8140	0,8440
7	**16**	**10000**	0,4986	0,7117	0,8140	0,8440
7	**24**	**250**	0,4986	0,7127	0,8121	0,8403
7	**24**	**1000**	0,4986	0,7128	0,8122	0,8404
7	**24**	**5000**	0,4986	0,7128	0,8123	0,8404
7	**24**	**10000**	0,4986	0,7128	0,8122	0,8404
9	**8**	**250**	0,4986	0,7527	0,8352	0,8468
9	**8**	**1000**	0,4986	0,7528	0,8353	0,8468
9	**8**	**5000**	0,4986	0,7527	0,8352	0,8468
9	**8**	**10000**	0,4986	0,7527	0,8352	0,8468
9	**16**	**250**	0,4986	0,7553	0,8375	0,8503
9	**16**	**1000**	0,4986	0,7555	0,8375	0,8504
9	**16**	**5000**	0,4986	0,7554	0,8375	0,8504
9	**16**	**10000**	0,4986	**0,7554**	**0,8375**	**0,8504**
9	**24**	**250**	0,4986	0,7550	0,8364	0,8491
9	**24**	**1000**	0,4986	0,7551	0,8363	0,8491
9	**24**	**5000**	0,4986	0,7550	0,8364	0,8492
9	**24**	**10000**	0,4986	0,7550	0,8364	0,8492

Table 14. Overall performance of the transformation matrix learning approach by using multi-resolution images in terms of VIF statistics.

n	*k*	*N*	VIF N-O	VIF E-O		
				Level 1	Level 2	Level 3
7	**8**	**250**	0,1845	0,3031	0,3805	0,4195
7	**8**	**1000**	0,1845	0,3031	0,3805	0,4193
7	**8**	**5000**	0,1845	0,3031	0,3805	0,4194

Table 14 (continued). Overall performance of the transformation matrix learning approach by using multi-resolution images in terms of VIF statistics.

7	**8**	**10000**	0,1845	0,3031	0,3805	0,4194
7	**16**	**250**	0,1845	0,3046	0,3831	0,4230
7	**16**	**1000**	0,1845	0,3046	0,3832	0,4232
7	**16**	**5000**	0,1845	0,3047	0,3831	0,4233
7	**16**	**10000**	0,1845	0,3047	0,3831	0,4232
7	**24**	**250**	0,1845	0,3055	0,3827	0,4216
7	**24**	**1000**	0,1845	0,3055	0,3828	0,4216
7	**24**	**5000**	0,1845	0,3055	0,3829	0,4216
7	**24**	**10000**	0,1845	0,3056	0,3828	0,4216
9	**8**	**250**	0,1845	0,3371	0,4217	0,4504
9	**8**	**1000**	0,1845	0,3371	0,4218	0,4506
9	**8**	**5000**	0,1845	0,3371	0,4217	0,4505
9	**8**	**10000**	0,1845	0,3371	0,4217	0,4505
9	**16**	**250**	0,1845	0,3407	0,4280	0,4579
9	**16**	**1000**	0,1845	0,3408	0,4279	0,4578
9	**16**	**5000**	0,1845	0,3407	0,4280	0,4578
9	**16**	**10000**	0,1845	**0,3407**	**0,4280**	**0,4577**
9	**24**	**250**	0,1845	0,3410	0,4286	0,4592
9	**24**	**1000**	0,1845	0,3411	0,4287	0,4592
9	**24**	**5000**	0,1845	0,3410	0,4287	0,4591
9	**24**	**10000**	0,1845	0,3410	0,4287	0,4590

Table 15. Overall performance of the transformation matrix learning approach by using multi-resolution images in terms of IFC statistics.

n	***k***	***N***	**IFC N-O**	**IFC E-O**		
				Level 1	**Level 2**	**Level 3**
7	**8**	**250**	2,0129	2,4286	2,4851	2,3960
7	**8**	**1000**	2,0129	2,4286	2,4846	2,3962
7	**8**	**5000**	2,0129	2,4285	2,4849	2,3963
7	**8**	**10000**	2,0129	2,4286	2,4846	2,3962
7	**16**	**250**	2,0129	2,4302	2,4984	2,4273
7	**16**	**1000**	2,0129	2,4302	2,4986	2,4258
7	**16**	**5000**	2,0129	2,4303	2,4985	2,4264
7	**16**	**10000**	2,0129	2,4302	2,4983	2,4259
7	**24**	**250**	2,0129	2,4225	2,4818	2,4003

Table 15 (continued). Overall performance of the transformation matrix learning approach by using multi-resolution images in terms of IFC statistics.

7	**24**	**1000**	2,0129	2,4224	2,4819	2,4020
7	**24**	**5000**	2,0129	2,4226	2,4817	2,4019
7	**24**	**10000**	2,0129	2,4225	2,4817	2,4018
9	**8**	**250**	2,0129	2,4784	2,4662	2,3382
9	**8**	**1000**	2,0129	2,4782	2,4661	2,3367
9	**8**	**5000**	2,0129	2,4782	2,4660	2,3367
9	**8**	**10000**	2,0129	2,4783	2,4660	2,3368
9	**16**	**250**	2,0129	2,4875	2,4979	2,3805
9	**16**	**1000**	2,0129	2,4872	2,4972	2,3809
9	**16**	**5000**	2,0129	2,4874	2,4972	2,3811
9	**16**	**10000**	2,0129	**2,4875**	**2,4974**	**2,3812**
9	**24**	**250**	2,0129	2,4819	2,4983	2,3798
9	**24**	**1000**	2,0129	2,4815	2,4977	2,3816
9	**24**	**5000**	2,0129	2,4818	2,4976	2,3816
9	**24**	**10000**	2,0129	2,4818	2,4975	2,3812

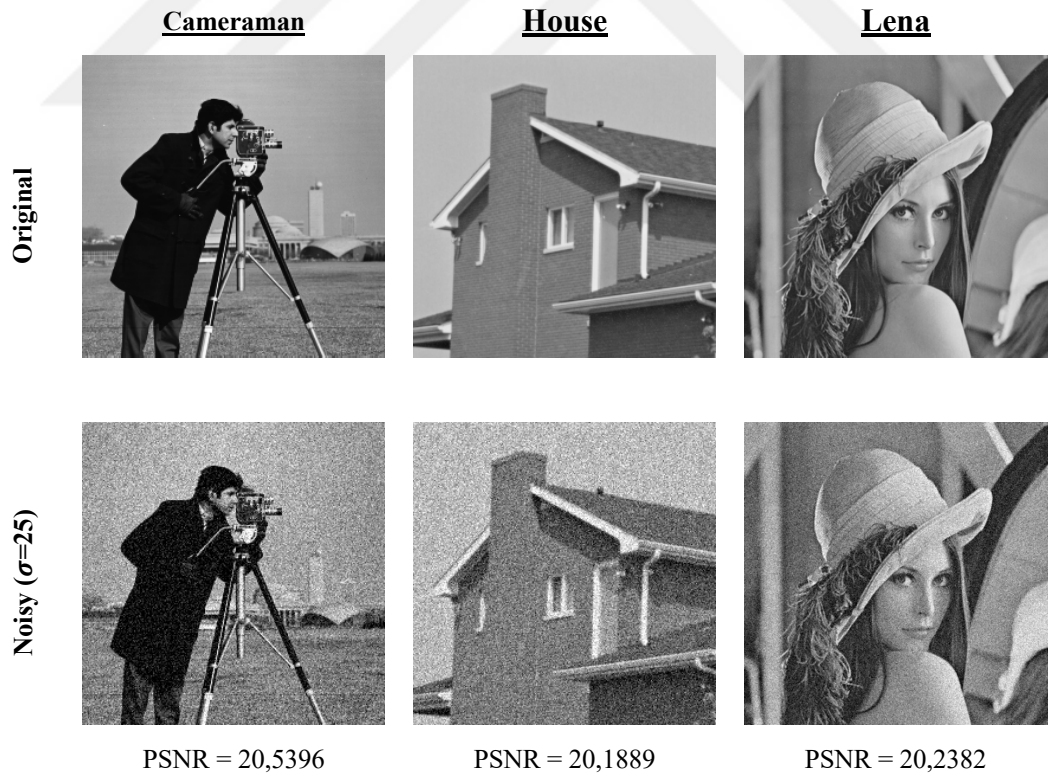


Figure 17. Visual samples for comparison of levels of the transformation matrix learning using multi-resolution images approach with the parameters n=9, k=16, and N=10000.

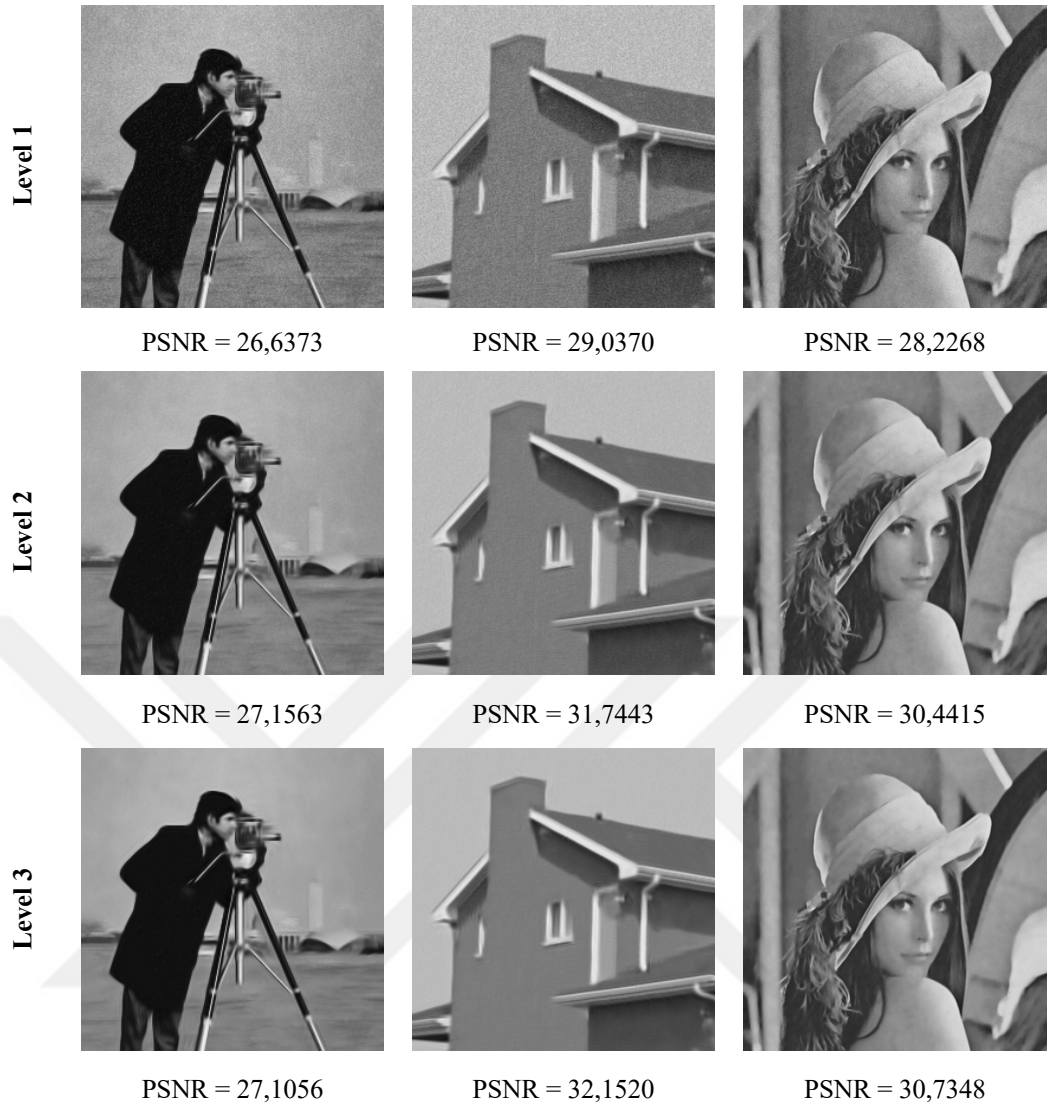


Figure 17 (continued). Visual samples for comparison of levels of the transformation matrix learning using multi-resolution images approach with the parameters n=9, k=16, and N=10000.

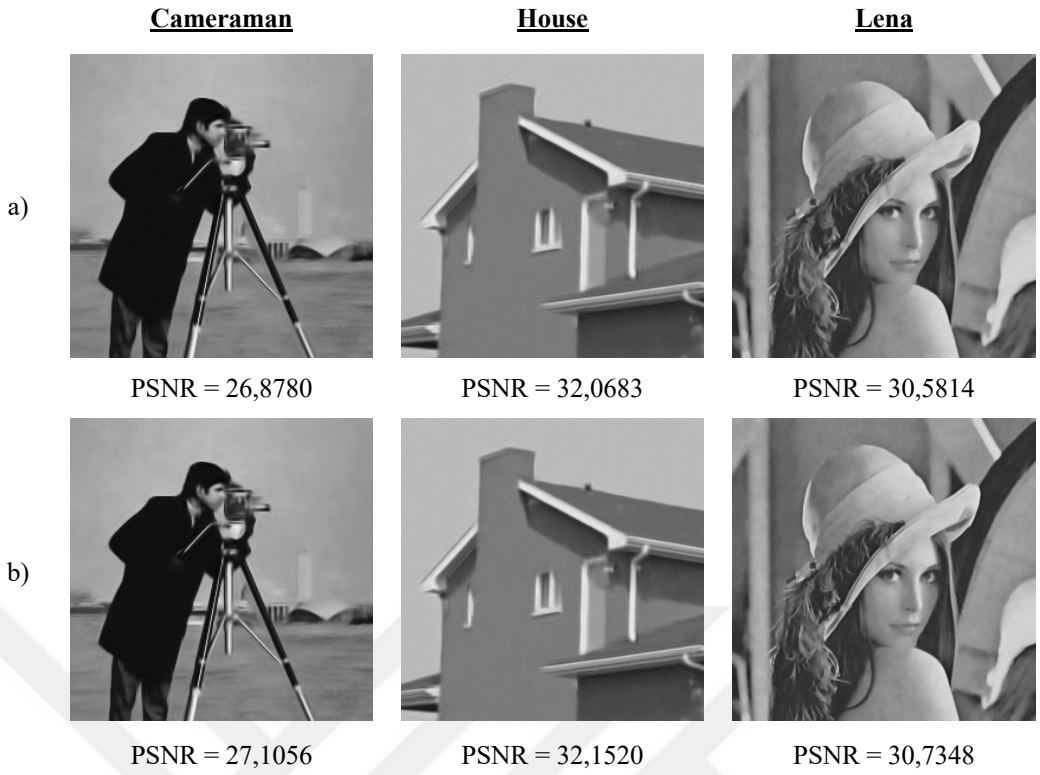


Figure 18. Visual samples for comparison of both transformation matrix learning approaches with the parameters n=9, k=16, N=10000, and level 3. a) The approach using original sized images b) The approach using multi-resolution images

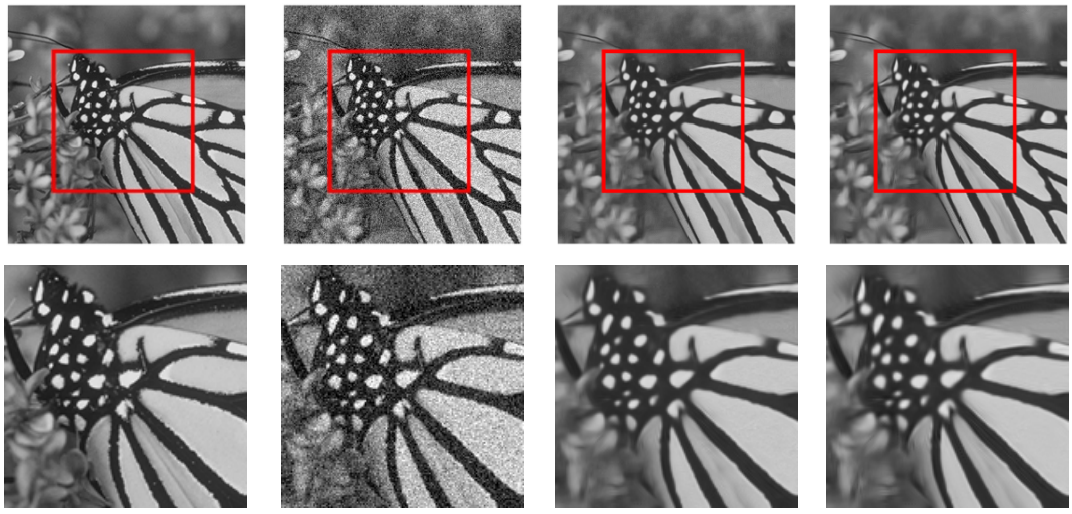


Figure 19. The effect of patch size by employing Algorithm 2 with parameters k=16, N=10000 and level 3.

4.3 Results of Preserving High Frequency Components

In Sections 3.1 and 3.2, the main aim is to remove the noise. According to the experimental results presented in Sections 4.1 and 4.2, denoising performance is satisfying whereas algorithms fail to preserve details, so blurring occurs. Therefore, keeping details should be further studied. The analysis of a feature mapping method is carried out in Section 4.3.1. The results of the alpha rooting based algorithm is presented in Section 4.3.2. These two methods aim to find estimated noise-free image by removing noise and keeping details at the same time. However, the last approach that is explained in Section 3.3.3 is a post-processing of denoising that aims to find the high frequency components (details) to add in the processed images. The results of the last approach are given in Section 4.3.3.

4.3.1 Results of Feature Mapping

The methods using feature mapping such as PCA, Well-Exposedness are expected to increase performace since these maps are suppposed to reveal the important parts of the images. The approach mentioned in Section 3.3.1.3 has the highest statistics for level 2 of Trial 2 and Trial 3. Hence, the statistics of feature mapping for Trials 2 and 3 are given in Table 16. As it is seen, k does not affect the performance considerably, and better results are obtained by considering n=7.

Table 16. The statistics of denoising with feature mapping for Trial 2 (Level 2) and Trial 3.

	n	***k***	**PSNR N-O**	**PSNR E-O**	**SSIM N-O**	**SSIM E-O**	**VIF N-O**	**VIF E-O**	**IFC N-O**	**IFC E-O**
Trial 2 Level 2	5	8	20,3237	27,2896	0,4986	0,7510	0,1845	0,3133	2,0129	**2,3910**
	5	16	20,3237	27,4201	0,4986	0,7670	0,1845	0,3238	2,0129	2,3861
	7	8	20,3237	**27,8105**	0,4986	0,8160	0,1845	0,3758	2,0129	2,3618
	7	16	20,3237	27,7127	0,4986	0,8194	0,1845	0,3843	2,0129	2,3048
	9	8	20,3237	27,5334	0,4986	**0,8275**	0,1845	0,4098	2,0129	2,2666
	9	16	20,3237	26,9757	0,4986	0,8209	0,1845	**0,4152**	2,0129	2,1554

Table 16 (continued). The statistics of denoising with feature mapping for Trial 2 (Level 2) and Trial 3.

Trial 3	5	8	20,3237	**27,8102**	0,4986	0,8015	0,1845	0,3441	2,0129	**2,3665**
	5	16	20,3237	27,7602	0,4986	0,8118	0,1845	0,3543	2,0129	2,3475
	7	8	20,3237	27,6995	0,4986	**0,8357**	0,1845	0,4092	2,0129	2,2287
	7	16	20,3237	27,2801	0,4986	0,8314	0,1845	0,4158	2,0129	2,1578
	9	8	20,3237	27,1991	0,4986	0,8281	0,1845	**0,4334**	2,0129	2,0754
	9	16	20,3237	26,5383	0,4986	0,8158	0,1845	0,4327	2,0129	1,9516

4.3.2 Results of Alpha Rooting

In this section, the results of the method explained in Section 3.3.2 are presented. According to the purpose of the alpha rooting based algorithm, the outcome images shoud be denoised successfully by keeping the details. Therefore, Trial 2 in Section 3.2.2 is developed by using iterative hard-thresholding approach. Even if the outcome images have blocking artifacts as it is illustrated in Figure 20, the statistical results are promising to study on.

a) Cameraman (PSNR = 28,3912)

b) House (PSNR = 30,5559)

c) Butterfly (PSNR = 28,2473)

d) Lena (PSNR = 30,3755)

Figure 20. Samples of outputs of the method explained in Section 3.3.2 Trial 2 with the parameters n=5, k=8.

Moreover, Trial 3 is developed based on Trial 2 with a small improvement that is about the stopping criteria of the iterative approach. This improvement does not have significant impact on the performance of the algorithm as it can be seen in Figure 21.

a) Cameraman (PSNR = 28,2209) b) House (PSNR = 31,2091)

c) Butterfly (PSNR = 28,4199) d) Lena (PSNR = 30,9278)

Figure 21. Samples of outputs of the method explained in Section 3.3.2 Trial 3 with the parameters n=5, k=8.

In the fourth trial, the weighting procedure of Trial 2 is changed for the reconstruction part, and variance of similar patches based weighting procedure is applied to remove blocking artifacts. The visual outputs can be seen in Figure 22.

a) Cameraman (PSNR = 28,5285) b) House (PSNR = 30,8554)

c) Butterfly (PSNR = 28,6091) d) Lena (PSNR = 30,5357)

Figure 22. Samples of outputs of the method explained in Section 3.3.2 Trial 4 with the parameters n=5, k=8.

Nevertheless, statistical results are just above the statistical results of previous approaches. Moreover, by focusing on the *Cameraman*'s face and the wing of *Butterfly*, preserving high frequency components is partly accomplished.

4.3.3 Results of High Frequency Component Learning

As it is understood by checking the output samples and the statistics of previous methods, the missing part in the proposed methods is preserving the details. Therefore, in this section, a method is described to find possible details of the denoised images that are processed by previously developed algorithms. Then, final outputs can be calculated as a blended version of denoised images and their estimated details. According to the given explanation of the method in Section 3.3.3, the outputs of the first trial are not reasonable. Therefore, 2D gradient of the noisy images are used for Trial 2. Example outputs of Trial 2 are given in Figure 23. The first idea to reach a final denoised image can be adding possible high frequency components to the denoised image. However, because of intensity values of the details, the result can be

saturated as it presented in Figure 24. Thus, to blend these two results, a constant multiplier or a map of details should be determined by optimizing it.

a) Details of Cameraman b) Details of House

c) Details of Butterfly d) Details of Lena

Figure 23. Visual examples of possible details of images.

Figure 24. Saturated example

The method described in Section 3.3.3 Trial 3 is a transformation matrix learning algorithm. Therefore, the output should be a denoised image. This is the main

difference between Trial 1 and Trial 2. It is developed as the combination of alpha rooting approach and multi-resolution images based learning algorithm. Since both are partly successful, the combination method should be more successful in both denoising and keeping details. To analyze this expectation, this method is employed up to level 3 with parameters n=7, k=8, and N=5000. These parameters are selected to decrease the computational cost, since these parameters provide very close results to those of parameters n=9, k=16, and N=10000. The highest results are obtained with level 3. The statistical results and visual outputs are given Table 17 and Figure 25 respectively.

Table 17. Comparison of overall statistics of the method described in Section 3.2.2 with the parameters n=9, k=16, and N=10000 and the method described in Section 3.3.3 Trial 3 with the parameters n=7, k=8, and N=5000. (Level 3 is considered for both.)

n	***k***	***N***	**PSNR E-O**	**SSIM E-O**	**VIF E-O**	**IFC E-O**
9	16	10000	28,3605	0,8504	0,4577	2,3812
7	8	5000	27,9749	0,8373	0,4073	2,2987

a) Cameraman (PSNR = 26,9785) b) House (PSNR = 30,7343)

c) Butterfly (PSNR = 27,7976) d) Lena (PSNR = 29,6739)

Figure 25. Level 3 sample outputs of the method described in Section 3.3.3 Trial 3 with the parameters n=7, k=8, and N=5000.

4.4 Comparison with Benchmark Studies

According to the statistical results and visual assessments given in Sections 4.1, 4.2 and 4.3, some of our algorithms have very promising results. These methods are selected to compare with well-known benchmark algorithms such as BM3D (Dabov et al., 2007b), KSVD (Elad and Aharon, 2006), EPLL (Zoran and Weiss, 2011), WNNM (Gu et al., 2014). Totally, four proposed approaches are selected which are explained in Section 3.2.2 (transformation matrix learning by using multi-resolution images), Section 3.3.2 Trial 2,3, and 4 (alpha rooting based high frequency learning approach by using iterative thresholding). The statistical comparison is given Table 18 by considering overall performances for Set12. Also, sample outputs of the methods mentioned in Section 3.2.2 and Section 3.3.2 Trial 4 are presented in Figure 26 and 27.The parameters for both methods are given as n=9, k=16, N=10000, level 3 and n=5, k=8 respectively. According to Figure 28 and 29, the proposed algorithm

explained in Section 3.3.2 Trial 4 is partly good at preserving the details, but it cannot eliminate the noise successfully. The other developed algorithm explained in Section 3.2.2, denoising performance is satisfying, but it fails to preserve details. Therefore, the method of Section 3.3.2 Trial 4 has promising statistical and visual results in comparision with the benchmark studies. Moreover, to see examples besides Set12 images, Figure 30 and

Table 19 are given for Castle and Tiger images from BSD68 dataset (Martin et al., 2001). It is obvious that the method mentioned in 3.2.2 makes output images over-smoothed, and the method mentioned as Trial 4 of Section 3.3.2 has more successful in keeping details, but it should be improved for better denoising performance.

Table 18. Statistical comparison with reference studies by using Set12 (σ = 25).

Method	**PSNR N-O**	**PSNR E-O**	**SSIM N-O**	**SSIM E-O**	**VIF N-O**	**VIF E-O**	**IFC N-O**	**IFC E-O**
BM3D	20,3237	29,5504	0,4986	0,8757	0,1845	0,3919	2,0129	2,3244
KSVD	20,3237	29,3354	0,4986	0,8667	0,1845	0,4268	2,0129	2,3462
WNNM	20,3237	**30,1081**	0,4986	**0,8876**	0,1845	**0,4582**	2,0129	**2,5840**
EPLL	20,3237	29,6294	0,4986	0,8796	0,1845	0,3989	2,0129	2,3891
Section 3.2.2	20,3237	28,3606	0,4986	0,8504	0,1845	0,4577	2,0129	2,3812
Section 3.3.2 Trial 2	20,3237	28,6754	0,4986	0,8447	0,1845	0,3360	2,0129	2,3224
Section 3.3.2 Trial 3	20,3237	28,8664	0,4986	0,8554	0,1845	0,3696	2,0129	2,1754
Section 3.3.2 Trial 4	20,3237	28,8748	0,4986	0,8497	0,1845	0,3461	2,0129	2,3185

Cameraman

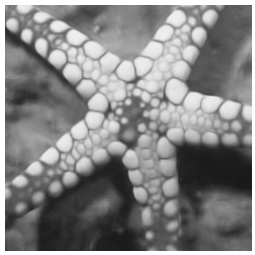
PSNR = 27,1056

House

PSNR = 32,1520

Peppers

PSNR = 28,6858

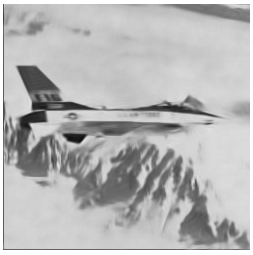


Starfish

PSNR = 26,9175

Butterfly

PSNR = 27,7319



Jetplane

PSNR = 26,3491

Parrot

PSNR = 26,9980

Lena

PSNR = 30,7348

Barbara

PSNR = 28,9613

Boats

PSNR = 28,5058

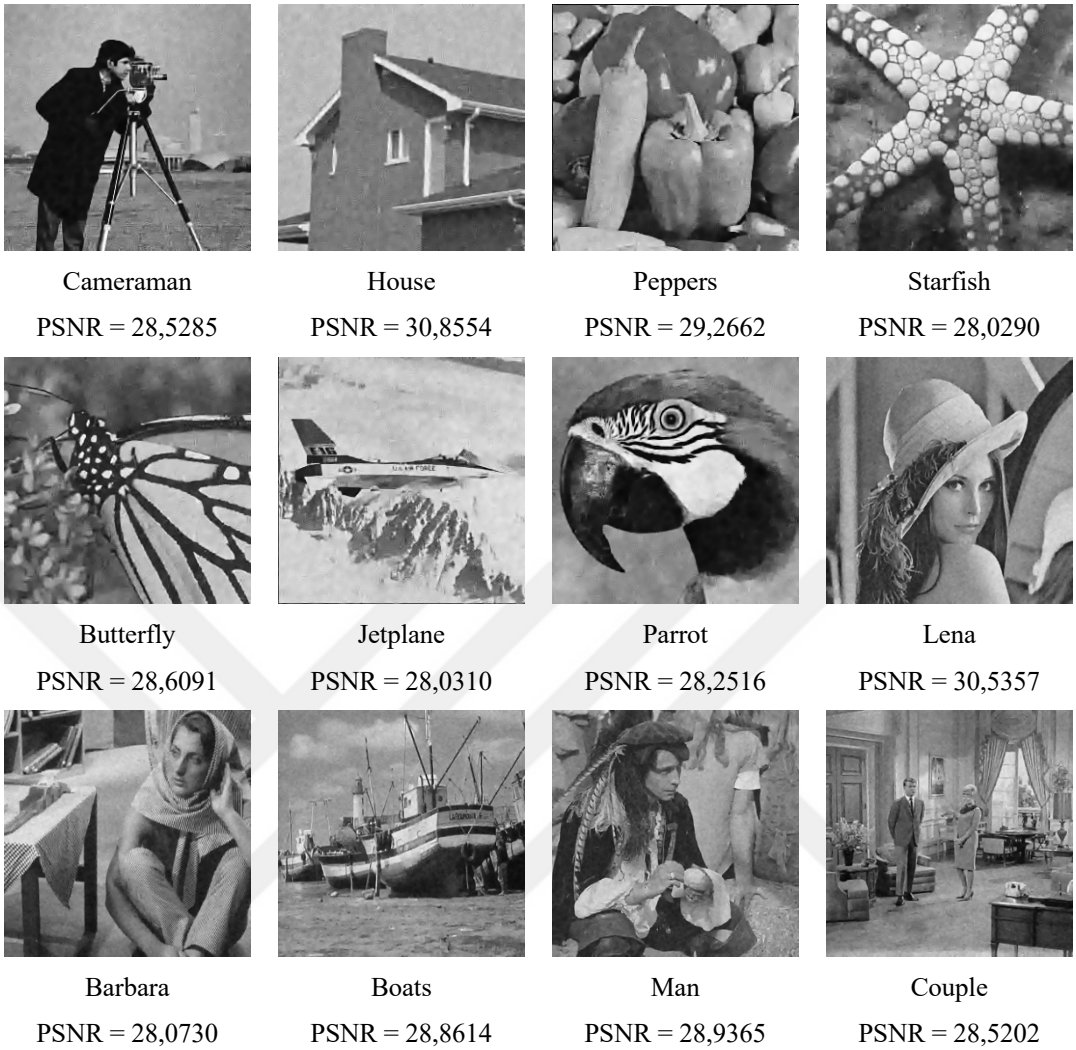
Man

PSNR = 28,2616

Couple

PSNR = 27,9236

Figure 26. Outputs of the method explained in Section 3.2.2 with the parameters defined as n=9, k=16, N=10000 and level 3.



Cameraman	House	Peppers	Starfish
PSNR = 28,5285	PSNR = 30,8554	PSNR = 29,2662	PSNR = 28,0290
Butterfly	Jetplane	Parrot	Lena
PSNR = 28,6091	PSNR = 28,0310	PSNR = 28,2516	PSNR = 30,5357
Barbara	Boats	Man	Couple
PSNR = 28,0730	PSNR = 28,8614	PSNR = 28,9365	PSNR = 28,5202

Figure 27. Outputs of the method explained in Section 3.3.2 Trial 4 with the parameters defined as n=5 and k=8.

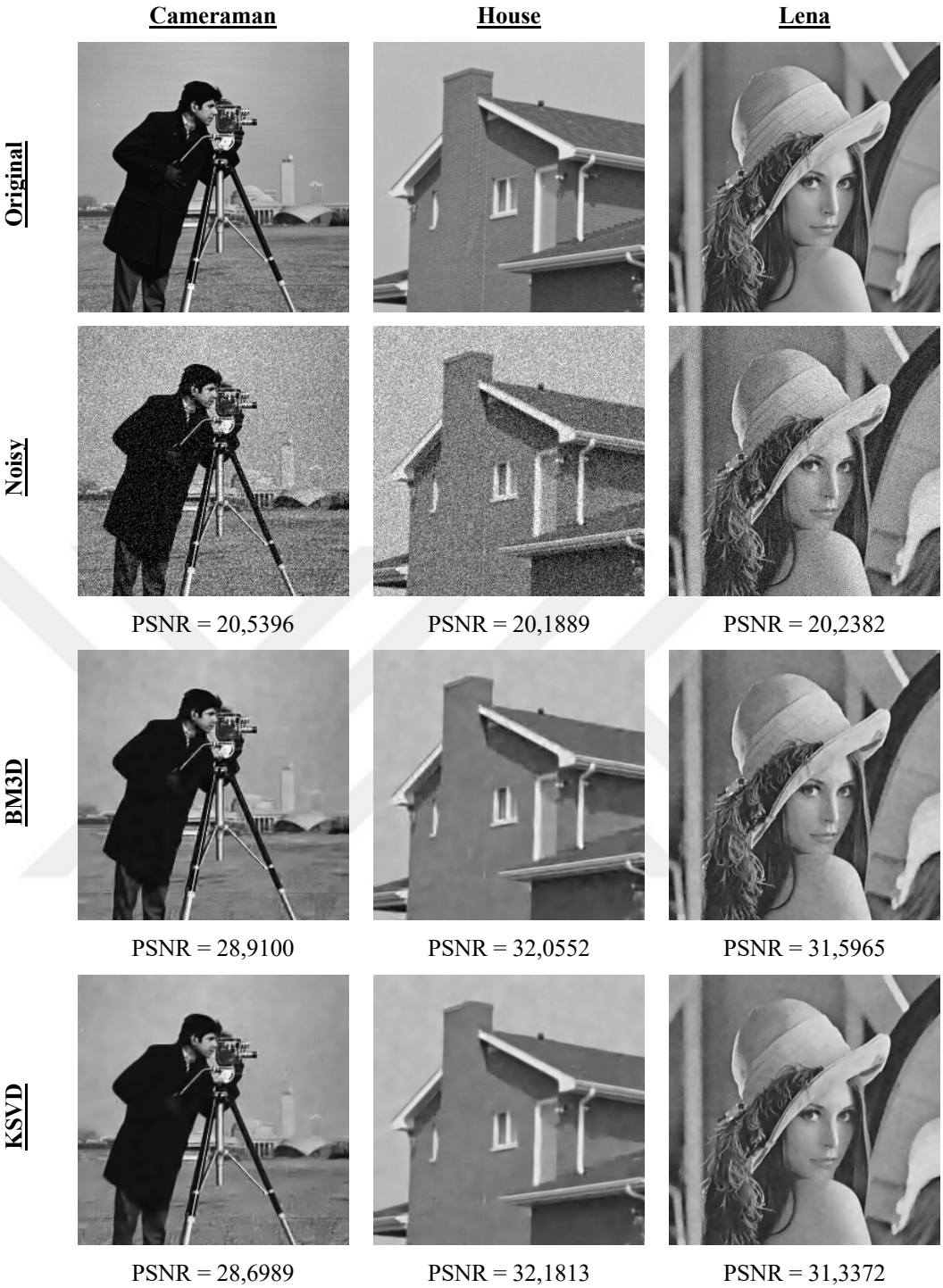


Figure 28. Comparison with the benchmark studies.

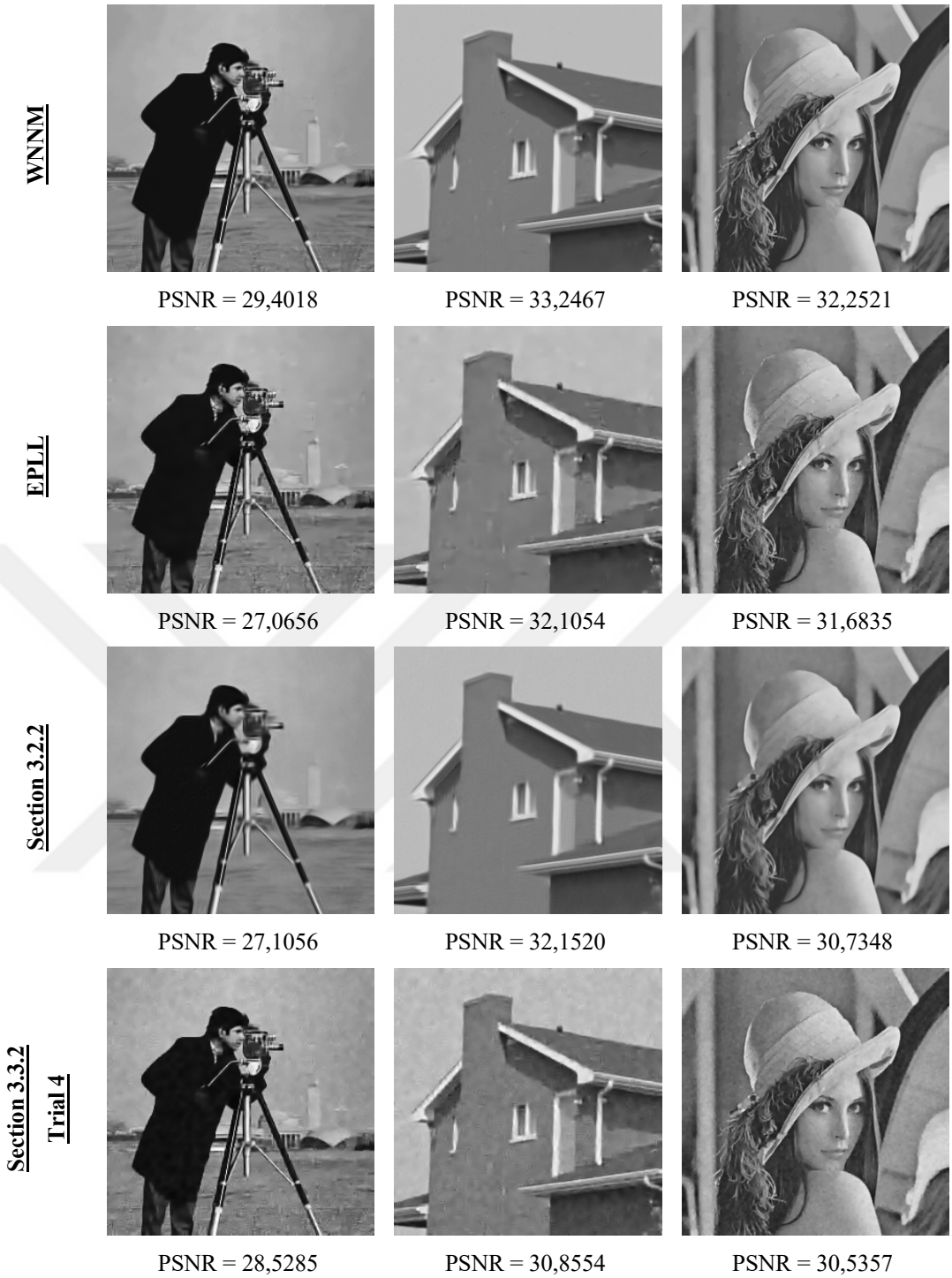


Figure 28 (continued). Comparison with the benchmark studies.

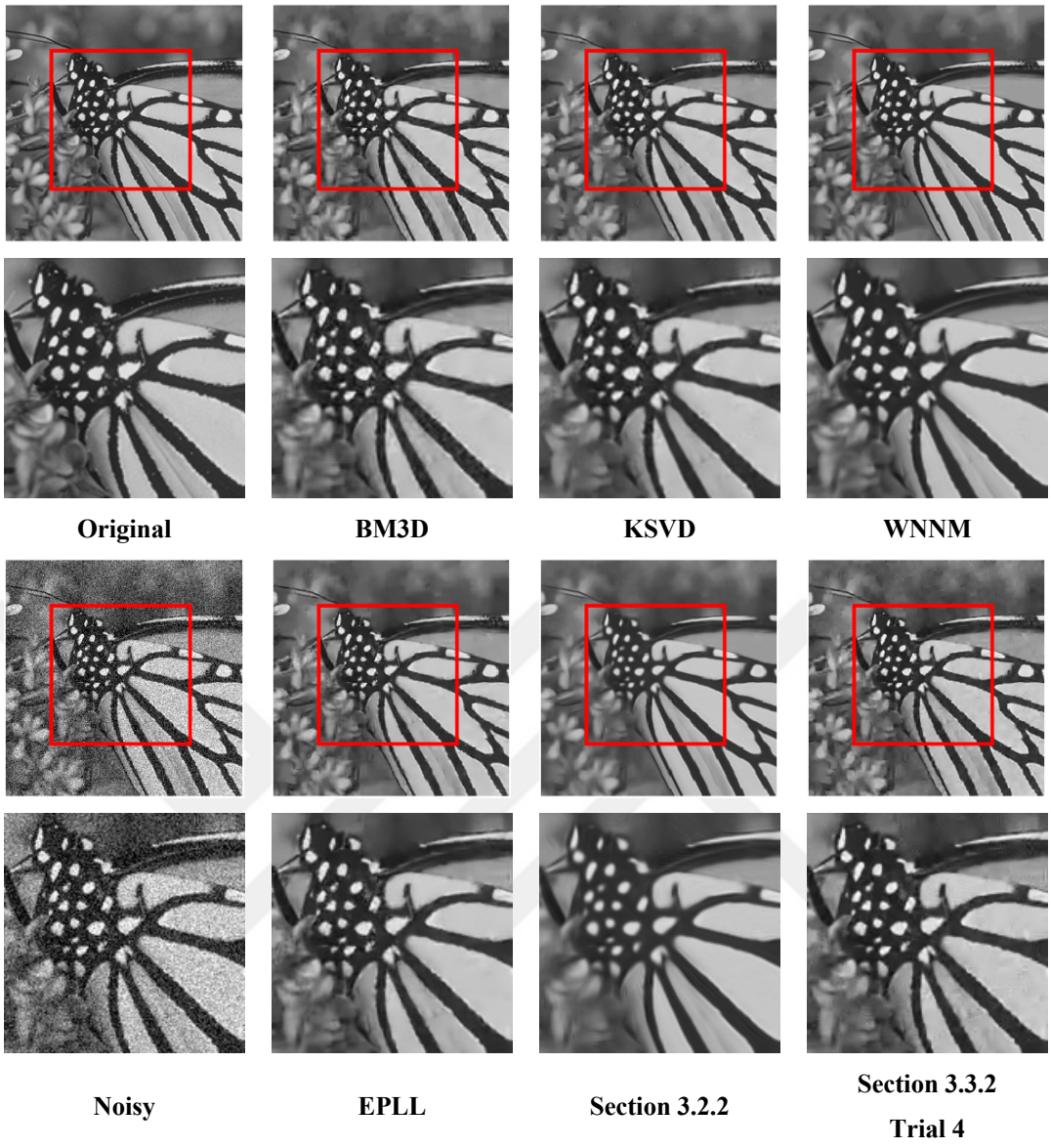


Figure 29. Comparison with the benchmark studies by focusing on details.

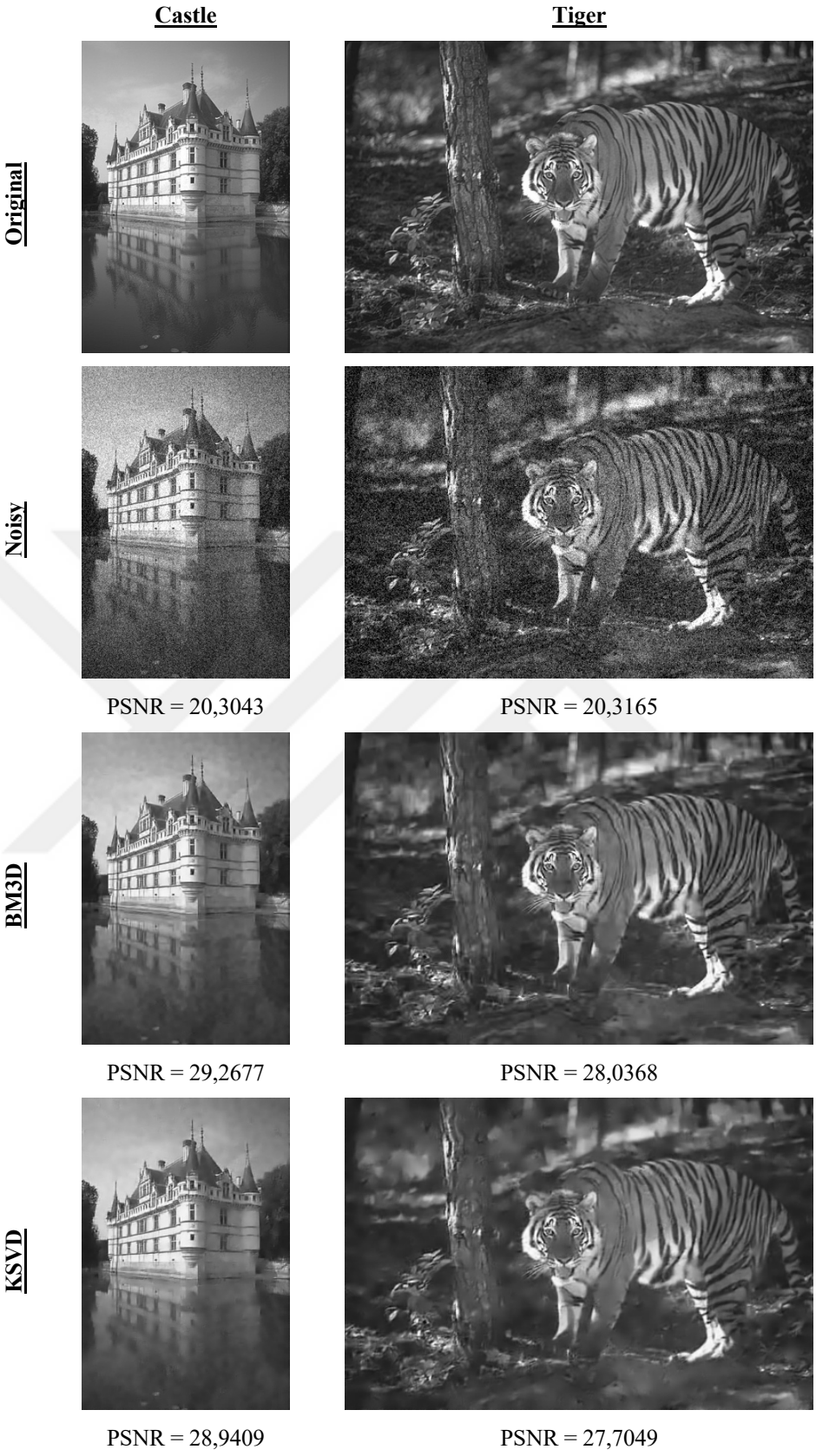


Figure 30. Comparison with the benchmark studied by using samples except Set12 images.

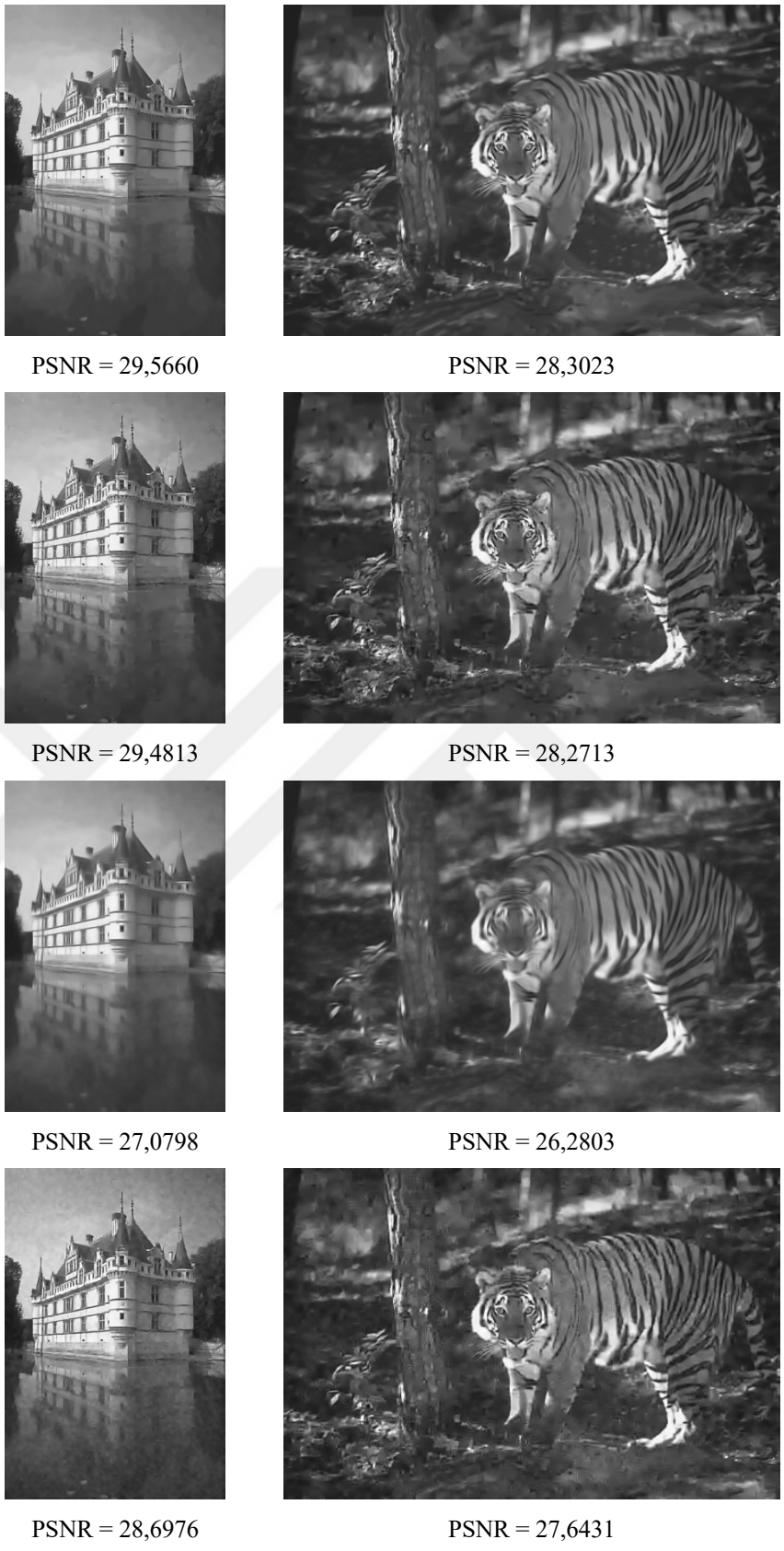


Figure 30 (continued). Comparison with the benchmark studied by using samples except Set12 images.

Table 19. Statistical comparison with the benchmark studies by using samples except Set12 images.

Method	Image	PSNR N-O	PSNR E-O	SSIM N-O	SSIM E-O	VIF N-O	VIF E-O	IFC N-O	IFC E-O
BM3D	Castle	20,3043	29,2677	0,3103	0,8411	0,1461	0,3617	1,5920	1,7128
	Tiger	20,3165	28,0368	0,4378	0,7988	0,2077	0,3655	2,3692	2,4408
KSVD	Castle	20,3043	28,9409	0,3103	0,8344	0,1461	0,3835	1,5920	1,7229
	Tiger	20,3165	27,7049	0,4378	0,7720	0,2077	0,3884	2,3692	2,3792
WNNM	Castle	20,3043	**29,5660**	0,3103	**0,8522**	0,1461	**0,4066**	1,5920	**1,8321**
	Tiger	20,3165	**28,3023**	0,4378	0,8036	0,2077	0,3949	2,3692	2,5463
EPLL	Castle	20,3043	29,4813	0,3103	0,8491	0,1461	0,3713	1,5920	1,8055
	Tiger	20,3165	28,2713	0,4378	**0,8083**	0,2077	0,3676	2,3692	**2,5483**
Section 3.2.2	Castle	20,3043	27,0798	0,3103	0,7989	0,1461	0,3976	1,5920	1,7294
	Tiger	20,3165	27,2803	0,4378	0,7416	0,2077	**0,4309**	2,3692	2,3809
Section 3.3.2 Trial 4	Castle	20,3043	28,6976	0,3103	0,7955	0,1461	0,3008	1,5920	1,7703
	Tiger	20,3165	27,6431	0,4378	0,7825	0,2077	0,3370	2,3692	2,4945

CHAPTER 5: CONCLUSION

Image denoising is one of the important problems to be solve in image processing , since the noise is inevitable truth for images. Although, the technology and successful methods are developed for image capturing, encoding, decoding and broadcasting, the noise can affect image quality easily. Therefore, denoising is a fundamental step for various types of image processing applications. Although it has been studied for decades, still the ultimate and perfect solution can not be found, so it is a challenging and active research topic.

In this study, the main aim is to develop Locally Linear Embedding (LLE) based image denoising algorithms, because LLE is not a common approach for image denoising. LLE is known as a dimensionality reduction algorithm in data science. By utilizing the data representation and reconstruction properties of LLE, estimated noise-free images are obtained from self-information cintained in noisy images by using traditional patch-based approaches and basic learning algorithms. During this research journey, several well-known and successful algorithms such as BM3D (Dabov et al., 2007b), KSVD (Elad and Aharon, 2006), WNNM (Gu et al., 2014) are used as inspiration sources.

During the study, many approaches have been tested and analyzed as much as possible. It was important to understand the impacts of the parameters and to guide the proposed approach. As a further study, the implementation codes of these algorithms can be converted into more effecient programming languages to test the computational cost, to transfer the process into GPUs to utilize parallel processing. This further study may reveal unclear relationships and impacts, that could not been seen, of parameters by considering wide range of parameter values and by solving more complex optimization problems in a short simulation times. Also, larger datasets containing wide variety of image resolution and content should be employed by considering different levels of noise to obtain more robust blind image denoising algorithms for different noise types. As it is mentioned, Set12 has been used in this study, but it may not be sufficient to make an exact comment about the performance of the algorithms, especially for the learning-based structures. In addition to that, more complex learning algorithms may be more successful. Moreover, some parameters, such as patch size and hard-

thresholding limit, could be more sensitive to image content and resolution to prevent blocking artifacts and over-smoothness.

As a result of these improvements, for any level of noise and any size of images, an algorithm can be designed for blind denoising of grey scale images, even be extended to color images. Additionally, the patch size and its shape may be adaptive based on image content. For example, smaller patch size could work more effectively for the regions that are containing details. Finally, this study has been completed with comprehensive experiments and promising image denoising results. The proposed denoising performance can compete against well-known algorithms in literature.

REFERENCES

Buades, A., Coll, B. and Morel, J.M. *A non-local algorithm for image denoising*, *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Diego, California, USA. 20-26 June 2005.

Dabov, K., Foi, A., Katkovnik, V. and Egiazarian, K. (2007a). *Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space*. *Proceedings of International Conference on Image Processing*. San Antonio, Texas, USA. 16-19 September 2007.

Dabov, K., Foi, A., Katkovnik, V. and Egiazarian, K. (2007b) *Image denoising by sparse 3-D transform-domain collaborative filtering*, IEEE Transactions on Image Processing, Vol. 16(8), pp. 2080–2095.

Dabov, K., Foi, A., Katkovnik, V. and Egiazarian, K. (2007c). *Joint image sharpening and denoising by 3D transform-domain collaborative filtering*. *Proceedings of the International TICSP Workshop on Spectral Methods and Multimedia Signal Processing.* Moscow, Russia. 1-2 September 2007.

Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K. *BM3D Image Denoising with Shape-Adaptive Principal Component Analysis*. *Proceedings of International Workshop on Signal Processing with Adaptive Sparse Structured Representations.* Saint-Malo, France. 06-09 April 2009.

Elad, M. and Aharon, M. (2006) *Image denoising via sparse and redundant representations over learned dictionaries*, IEEE Transactions on Image Processing, Vol. 15(12), pp. 3736–3745.

Fei, D.G. and Medioni, S.G. *Locally linear denoising on image manifolds*. *Proceedings of the International Conference on Artificial Intelligence and Statistics*. Sardinia, Italy. 13-15 May 2010.

Gu, S., Zhang, L., Zuo, W. and Feng, X. *Weighted nuclear norm minimization with application to image denoising*. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Columbus, OH, USA. 23-28 June 2014.

Karakaya, D., Ulucan, O. and Turkan, M. *Pas-Mef: Multi-Exposure Image Fusion Based On Principal Component Analysis, Adaptive Well-Exposedness And Saliency Map. Proceedings of the IEEE International Conference on Acoustics, Speech and*

Signal Processing. Singapore, China. 7-13 May 2022.

Kun, A. (2012) *ISO Sensitivity | Learn How ISO Sensitivity Works* [Online]. Available at: https://www.exposureguide.com/iso-sensitivity/. (Accessed: 4 June 2022).

Li, H. and Liu, F. *Image denoising via sparse and redundant representations over learned dictionaries in wavelet domain*. *Proceedings of the International Conference on Image and Graphics*. Xi an, Shanxi, China. 20-23 September 2009

Mairal, J., Bach, F., Ponce, J., Sapiro, G. and Zisserman, A. *Non-local sparse models for image restoration*. *Proceedings of the IEEE International Conference on Computer Vision*. Kyoto, Japan. 29 September – 2 October 2009.

Martin, D., Fowlkes, C., Tal, D. and Malik, J. *A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics*. *Proceedings of the IEEE International Conference on Computer Vision*. Vancouver, British Columbia, Canada. 7-14 July 2001.

Mertens, T., Kautz, J. and Van Reeth, F. (2009) *Exposure fusion: a simple and practical alternative to high dynamic range photography*, Computer Graphics forum, Vol. 28(1), pp. 161–171.

Pierazzo, N., Morel, J.M. and Facciolo, G. (2017) *Multi-Scale DCT Denoising*, Image Processing On Line, Vol. 7, pp. 288–308.

Roweis, S.T. and Saul, L.K. (2000) *Nonlinear Dimensionality Reduction by Locally Linear Embedding*, Science, Vol. 290(5500), pp. 2323–2326.

Scetbon, M., Elad, M. and Milanfar, P. (2021) *Deep K-SVD Denoising; Deep K-SVD Denoising*, IEEE Transactions on Image Processing, Vol. 30, p. 2021.

Shi, R., Shen, I.F. and Chen, W. *Image denoising through locally linear embedding*. *Proceedings of the Conference on Computer Graphics, Imaging and Vision: New Trends 2005*. Beijing, China. 26-29 July 2005.

Türkan, M., Thoreau, D. and Guillotel, P. *Self-content super-resolution for ultra-HD up-sampling. Proceedings of the Conference on Visual Media Production.* London, United Kingdom. 5-6 December 2012.

Türkan, M., Thoreau, D. and Guillotel, P. *Optimized neighbor embeddings for single-image super-resolution*. *Proceedings of the IEEE International Conference on Image Processing.* Melbourne, Australia. 15-18 September 2013.

Wold, S., Esbensen, K. and Geladi, P. (1987) *Principal component analysis*, Chemometrics and Intelligent Laboratory Systems, Vol. 2(1–3), pp. 37–52.

Yang, D. and Sun, J. (2018) *BM3D-Net: A Convolutional Neural Network for Transform-Domain Collaborative Filtering*, IEEE Signal Processing Letters, Vol. 25(1), pp. 55–59.

Yaroslavsky, L.P. (1985) *Digital Picture Processing: An Introduction*. 1st Edition, Berlin, Heidelberg: Springer Verlag.

Zhang, M. and Gunturk, B.K. (2008) *Multiresolution bilateral filtering for image denoising*, IEEE Transactions on Image Processing, Vol. 17(12), pp. 2324–2333.

Zoran, D. and Weiss, Y. *From learning models of natural image patches to whole image restoration*. *Proceedings of the IEEE International Conference on Computer Vision*. Barcelona, Spain. 6-13 November 2011.