Final Term Report - Team 10
Comparative study of various Image Denoising techniques

Atharv Ramesh Nair
EE20BTECH11006

Nitish S
EE20BTECH11044

Rahul V
AI20BTECH11030

Saumyaranjan Mohanty
AI23RESCH04001

Abstract

As a part of this course project, we aim to investigate the existing techniques for image denoising. Images play a major role in modern day-to-day life, and various noises are introduced during image acquisition and processing and due to human error, as well. Researchers have proposed various methods of decreasing noise in the image, through classical image processing,hand crafted feature filters and machine learning and deep learning-based approaches. In this report, we provide a formulation of the image denoising problem , a literature survey of both classical and machine learning-based approaches, implementation of some of the more influential algorithms and a comparison of results.

Keywords: Image denoising, convolutional neural network, BM3D, RIDNet, DnCNN

1. Introduction

Noise in an image is introduced during image acquisition, compression, transmission, and storage due to various factors such as environment, acquisition device, transmission channel, etc. As image processing tasks such as object detection, tracking, segmentation, medical image analysis, video processing, etc. depend upon the quality of the image being processed, noisy images adversely affect all the ancillary image and video processing activities. Hence image denoising is a vital pre-processing step for the majority of image processing and analysis systems.
An Image, generally being a 2-D matrix with a lot of local and global correlated patches, noise introduced either locally or globally may adversely affect various subsequent image processing tasks. Hence, researchers have focused on both local and global denoising methodologies to achieve a high correlation of visible features with ground truth images. Noise introduced to images is a high-frequency component. Edges and textures are also high-frequency components and hence create issues in separating noise from vitals features such as edges and textures. Hence, it is vital to remove these high-frequency noise components while retaining high-frequency image features.
From a mathematical point of view, image denoising is an inverse problem, i.e. we are trying to predict the ground truth image back from the noisy image, and hence, there is no unique solution for this task. Hence, a lot of methods have evolved, with a trade-off appearing between compute time, visibility enhancement, and other task-specific denoising methodologies. In this project, we delve in detail into the work carried out in the field of image denoising, covering both classical image processing techniques and rapidly gaining popularity Machine learning-based image denoising techniques and carry out a detailed comparative analysis of various technologies and discuss their pros and cons. As a fundamental aspect of our initial investigation, a comprehensive literature review was undertaken. In this regard, a comprehensive review article [1] served as an excellent starting point for exploring the realm of image denoising.

Generally, image denoising methods can be classified into various sub-categories as shown below:

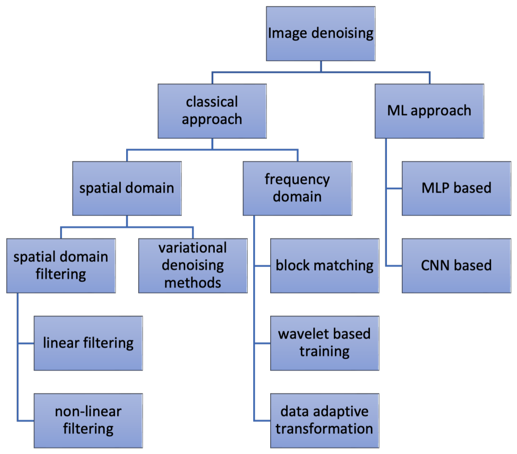


Figure 1. classification of denoising methods

2. Noise Modeling

The presence of noise in images can have detrimental effects, including the appearance of artifacts and blurriness. To achieve efficient image denoising, a reliable estimate of the noise model is crucial. Sources of noise may arise from sensor inaccuracies, camera imperfections, or a noisy channel. Some common types of Noise PDFs found in image processing applications include - Gaussian, Rayleigh, Impulse, etc.
Mathematically, the image denoising problem statement can be modeled as: $y = x + n$, where:
y: observed noisy image
x: unknown clean image

n: additive white Gaussian noise(AWGN)

As denoising is an inverse process, we can not get a unique solution from the noisy image. So generally, the approach followed is to obtain a good estimate of the clean image $\hat{x}$, so that the distance metric between the clean image and the estimated image is minimized.

3. Classical Methods

Spatial Domain

Simple Filters

Averaging filters are a simple denoising method that smooths the image by replacing each pixel value with the average value of its neighbouring pixels. They are effective at reducing noise but can also blur the image and reduce sharpness. Gaussian filters use a weighted average to smooth the image. The weights are determined by a Gaussian function, which assigns higher weights to pixels closer to the centre of the neighbourhood. As a result, they are more effective than averaging filters at reducing noise while preserving edge sharpness to a decent degree.
The median filter is a simple non-linear filter that replaces each pixel value with the median value of its neighbouring pixels. They work well for impulse noise because the median value is less sensitive to outliers than the mean. However, they are not effective at reducing other types of noise, such as Gaussian noise. They can also introduce some blurring in areas with sharp edges or textures, as the median operation can smooth out these high-frequency details.

Wiener Filters

Wiener filter [2] is the MSE-optimal stationary linear filter for images degraded by additive noise and blurring. This filter is implemented under the assumption that the signal and the noise processes are second-order stationary. Hence, only noise processed with zero means is considered.
The method is founded on considering images and noise as random variables, and the objective is to find an estimate $\hat{f}$ of the uncorrupted image f such that the mean squared error (MSE) between them is minimized. The error measured is defined as :

$$e^2 = E\left\{(f-\hat{f})^2\right\} \tag{1}$$

where $E\{.\}$ is the expected value.
below assumptions are made:
1. Noise and image are uncorrelated.
2. One or the other has zero mean.
3. Intensity levels in the estimate are a linear function of the levels in the degraded image.

The frequency domain expression for the Wiener filter is given by:

$$F(u,v) = \frac{H^*(u,v)S_f(u,v)}{S_f(u,v)\left|H(u,v)\right|^2 + S_\eta(u,v)} \tag{2}$$

where:
$F(u,v)$: Wiener filter transfer function
$H(u,v)$: degradation transfer function
$H^*(u,v)$: complex conjugate of $H(u,v)$
$S_\eta(u,v)$: power spectrum of the noise
$S_f(u,v)$: power spectrum of the ground truth image

From the equations, it can be observed that Wiener filtering has two separate parts:
1. An inverse filtering part
2. A noise smoothing part
So it does high pass filtering through inverse filtering and removes the noise with a compression operation, which is a low pass filtering operation.

Bilateral Filtering

Bilateral filters [3] use a weighted average of neighbouring pixels based on both their spatial distance and their intensity difference. Assuming the original image is I, the bilateral filter can be formulated as

$$BF(x) = \frac{1}{W_p(x)} \sum_{y\in\Omega} I(y) w_s(||x-y||) w_r(|I(x)-I(y)|) \tag{3}$$

where I is the input image, x is the current pixel, y is a neighbouring pixel, Ω is the window centered around x, w_s and w_r are the spatial and range weighting functions (gaussian, respectively, and $W_p(x)$ is the normalization factor

Non-Local Means

The Non-Local Means (NLM) algorithm is a popular image-denoising technique used extensively in image processing applications. It was initially introduced by Buades et al. [4]. The fundamental concept behind the NLM algorithm involves the estimation of a noise-free image by analyzing the resemblance of small image patches across various positions in the image

Basic Algorithm

The input, noisy image v, is assumed to come from the classical additive noise model

$$v(x) = u(x) + n(x), x \in \Omega, \tag{4}$$

where u is the original image, Ω is the set of pixels and n is the noise perturbation: $(n(x)), x \in \Omega$ are assumed to be independent and identically distributed (i.i.d.) Gaussian variables with mean 0 and standard deviation $\sigma > 0$. In the case of gray-level images, u and v take real values (integer values in the discrete model) whereas for color images, u and v are vector-valued. This algorithm works well for other types of noises too. The output, denoised image, is the estimator $\hat{u}$ computed as

$$\hat{u}(x,y) = \sum_{y \in \Omega_x} w(x,y)v(y) \tag{5}$$

where the weight w(x, y) estimates the similarity between the pixel x and y in the original image and the sequence of weights satisfies

$$w(x,y) \geq 0 \text{ and } \textstyle\sum_{x,y} w(x,y) = 1, \forall x \in \Omega, y \in \Omega_x$$

The set of pixels $\Omega_x$is a neighbourhood of $x \in \Omega$. The neighbourhood should ideally include the entire image to look for similar pixels in the non-local neighbourhood, but it is recommended to consider a small neighbourhood around x for computational efficiency.
The weight $w(x,y)$ is given by

$$w(x,y) = \frac{1}{n(x)} \exp\left\{ -\frac{\|v(N_x) - v(N_y)\|_2^2}{h^2} \right\} \tag{6}$$

where $n(x)$ is a generic normalization factor and $h > 0$ is a filtering parameter. The similarity between two pixels, x and y, depends on the similarity of the intensity gray level vectors $v(N_x)$ and $v(N_y)$, where N_k denotes a square neighbourhood of fixed size and centred at a pixel k. $\|\|_2$ is the euclidean norm between the image patches. Jacques et al. [5] gives a detailed description of the basic algorithm. We have implemented this in python. The code is not optimised and takes a significant amount of time to run. Hence we have used OpenCV's [6] inbuilt function for comparative analysis. .

Variations

Buades et al. [4] also mentions using a Gaussian-Euclidean Norm instead of the standard one. Jacques et al. [5]explains the same in detail

$$\|v(x) - v(y)\|_{2,a}^2 = \sum_{\{z \in \mathbb{Z}^2 : \|z\|_\infty < d_s\}} K_{G_a}(z) \|v(x+z) - v(y+z)\|_2^2 \tag{7}$$

$$K_{G_a}(z) = \frac{e^{-\frac{\|z\|_2^2}{2a^2}}}{\sum_{\{t \in \mathbb{Z}^2 : \|t\|_\infty < d_s\}} e^{-\frac{\|t\|_2^2}{2a^2}}} \tag{8}$$

and $ds = \frac{d-1}{2}$ being the patch side half-length, d being an odd number. This method is denoted as NLM-Pa, where a is a gaussian kernel parameter. The earlier method is denoted as NLM-P. NLM-P is still mostly preferred rather than NLM-Pa due to the reason the increase in quality provided by the Gaussian kernel is low, while it requires the additional parameter a to be tuned. [5] also gives a description of many ways to improve the computational efficiency of this algorithm, including integral images and FFT-based methods.

Variational Denoising Methods

The goal of denoising methods is to estimate a noise-free image from a noisy one by minimizing an energy function E. The energy function is calculated from the noisy image, and a mapping procedure is used to obtain a low number corresponding to a noise-free image. The denoised image is then obtained by minimizing the energy function. The motivation for these methods is maximum a posteriori (MAP) probability estimation

Total variation Regularization

TV Regularization [7] is based on the statistical fact that natural images are locally smooth, and the pixel intensity gradually varies in most regions. It is defined as follows

$$R_{TV}(x) = \|\nabla x\|_1$$

where ∇x is the gradient of x.
It has achieved great success in image denoising because it can not only effectively calculate the optimal solution but also retain sharp edges. However, it has three major drawbacks: textures tend to be over-smoothed, flat areas are approximated by a piecewise constant surface resulting in a stair-casing effect, and the image suffers from losses of contrast. To improve the performance of the TV-based regularization model, extensive studies have been conducted on image smoothing by adopting partial differential equations. For example, Leonid I. Rudin [7] proposed a non-linear partial differential equations-based denoising algorithm for constrained TV.
Let the observed intensity function $u_0(x,y)$ denote the pixel values of a noisy image for $x, y \in \Omega$. Let $u(x,y)$ denote the desired clean image, so

$$u_0(x,y) = u(x,y) + n(x,y),$$

when n is the additive noise.
We wish to reconstruct u from u_0. The space of Bounded variation functions is the proper class for many basic image processing tasks. Thus, our constrained minimization problem is:

$$\text{minimize } \int_\Omega \sqrt{{u_x}^2 + {u_y}^2}\, dx\, dy$$

$$\text{subject to constraints } \int_\Omega u\, dx\, dy = \int_\Omega u_0\, dx\, dy$$

This constraint signifies the fact that the white noise $n(x,y)$ is of zero mean and

$\int_0 \frac{1}{2}(u-u_0)^2\,dx\,dy = \sigma^2$, where $\sigma > 0$ is given.

The second constraint uses a priori information that the standard deviation of the noise $n(x,y)$ is σ.
Thus we have one linear and one nonlinear constraint. The method is totally general as regards the number and shape of constraints. We arrive at an Euler-Lagrange equation. The solution procedure uses a parabolic equation with time as an evolution parameter, or equivalently, the gradient descent method.

Weighted Nuclear Norm Minimization

The WNNM [8] algorithm aims to denoise a noisy image by exploiting the low-rank property of image patches. The algorithm operates on small patches of the image and applies a weighted nuclear norm penalty to encourage low-rank solutions. The key idea behind the algorithm is to model the noisy image patches as a linear combination of a few basis patches, where the coefficients of the linear combination are sparse. This is a reasonable assumption for natural images, as they often exhibit a low-rank structure in small patches due to the smoothness of the image content. Given a matrix Y of noisy image patches, WNNM aims to find a low-rank approximation X which minimizes the following cost function

$$\hat{X} = \arg\min_X ||Y-X||_F^2 + ||X||_{w,*} \tag{9}$$

where $||Y-X||_F^2$ is the Frobenius norm between the noisy observations and the low-rank approximation X, and $||X||_{w,*} = \sum_i |w_i \sigma_i(X)|$ is the weighted nuclear norm penalty on X where $\sigma_i(X)$ is the i^{th} singular value of X.

The solution to this problem can be found by applying soft thresholding to the singular values of the matrix X using the corresponding weights. The soft thresholding operation is defined as,

$$\sigma_i(\hat{X}) = S_{w_i}(\sigma_i(X)) = \max(\sigma_i(X) - w_i, 0) \tag{10}$$

Low-rank approximation of X can be obtained by using the soft thresholded singular values of X and the orthonormal basis obtained from the SVD of X. The resulting low-rank approximation X can then be used to reconstruct the denoised image patches. Overall, WNNM is a powerful method for image denoising that outperforms many other state-of-the-art denoising algorithms in terms of both quantitative metrics and visual quality. The way WNNM is applied to image denoising is that we take a patch and we find similar patches to it in the neighbourhood of the original patch. We flatten each of the similar patches found (and the original patch) and stack them to form a matrix. We find the low-rank approximation to this matrix and assign it to be the denoised version of the original patch.

Transform Domain

Transform domain filtering methods involve transforming the noisy image to another domain before applying a denoising procedure. These methods take into account the different characteristics of the image and its noise, with larger coefficients representing the high-frequency parts, i.e., the edges or details, and smaller coefficients representing the noise. The basis transform functions used for the transformation can be either data-adaptive or non-data-adaptive.

Wavelet Transform Denoising

Wavelet denoising is a popular method for removing noise from digital images. It uses a mathematical technique called wavelet transform [9] to decompose an image into multiple scales, each representing different levels of detail. The wavelet transform can separate the image into different frequency components, where high-frequency components represent fine details such as edges and low-frequency components represent the overall structure of the image. Wavelet denoising is often performed by applying a thresholding operation to the wavelet coefficients. [10] use soft-thresholding, which is a nonlinear function that shrinks the magnitudes of the wavelet coefficients toward zero. After this, an inverse transform is performed to obtain the denoised signal. [11] propose an adaptive thresholding scheme for wavelet denoising of medical images. This is based on statistical estimation of noise. Because of superior thresholding techniques, their method outperforms traditional wavelet denoising methods.

Block-Matching and 3D Filtering

BM3D is a widely-used image denoising algorithm that is based on the idea of collaborative filtering. The algorithm was first proposed by Dabov et al. [12]. The basic idea behind BM3D is similar to that of NLM i.e, exploiting the redundancies in natural images to remove noise while preserving image details. It is performed in two steps. Lebrun et al. [13] provide a detailed analysis of the algorithm and its variations. Its notation which is easier to follow, has been used below.

First Denoising Step

Grouping : The original noisy image u is searched in a P -centered $n^{hard} \times n^{hard}$ neighbourhood for patches Q similar to the reference patch P. The set of similar patches is simply defined by

$$P(P) = \{Q : d(P,Q) \leq \tau_{hard}\} \tag{11}$$

where

- τ_{hard} is the distance threshold
- $$d(P,Q) = \frac{\|\gamma'(P) - \gamma'(Q\|_2^2}{(k^{hard})^2}$$
- γ' is a hard thresholding operator

The 3D group denoted $\mathbb{P}(P)$ is then built by stacking up the matched patches $P(P)$. In order to speed up the process, only the N^{hard} patches in $P(P)$ closest to the reference patch are kept in the 3D group.

Collaborative Filtering : Once the 3D-block $\mathbb{P}(P)$ is built, collaborative filtering is applied. A 3D isometric linear transform is applied to the group, followed by a shrinkage of the transform spectrum. Finally, the inverse linear transform is applied to estimate for each patch.

$$\mathbb{P}(P)^{hard} = \tau_{3D}^{hard^{-1}}(\gamma(\tau_{3D}^{hard}(\mathbb{P}(P)))) \tag{12}$$

where γ is a hard thresholding operator. BM3D allows several 3D transform techniques to transform the domain from spatial to frequency. The basic BM3D uses a three- dimensional discrete cosine transforms as a 3D transform. Another approach is to use a combination of 2D bi-orthogonal transform and 1D-Haar or Walsh-Hadamard transform. The latter has proven to have better results.

Aggregation : The basic estimate of the true image is computed by weighted averaging of all of the obtained block-wise estimates that are overlapping.

$$u^{basic}(x) = \frac{\sum\limits_{P} w_p^{hard} \sum\limits_{Q \in P(P)} \chi_Q(x) u_{Q,P}{}^{\text{hard}}(x)}{\sum\limits_{P} w_p^{hard} \sum\limits_{Q \in P(P)} \chi_Q(x)} \tag{13}$$

- $\chi_Q(x) = 1$ iff $x \in Q$, 0 otherwise.
- $u_{Q,P}{}^{\text{hard}}(x)$ is the estimate of the value of the pixel x belonging to the patch Q obtained
- $$w_P^{\text{hard}} = \begin{cases} N_p^{\text{hard}^{-1}} & \text{if } N_p{}^{\text{hard}} \geq 1 \\ 1, & \text{otherwise} \end{cases}$$
- $N_P{}^{hard}$ is the number of retained (non-zero) coefficients in the 3D block after hard thresholding

Second Denoising Step

In this second part of the algorithm, grouping is done based on the basic estimate

$$\mathbb{P}^{\text{basic}}(P) = \{Q : d(P,Q) \leq \tau_{wien}\} \tag{14}$$

For collaborative filtering, instead of hard thresholding, a Wiener filter is used where the basic estimate u^{basic} is used as the oracle.

$$w_p(\xi) = \frac{|\tau_{3D}^{wien}(\mathbb{P})^{\text{basic}}(\mathbb{P})(\xi)|^2}{|\tau_{3D}^{wien}(\mathbb{P})^{\text{basic}}(\mathbb{P})(\xi|^2 + \sigma^2} \tag{15}$$

An estimate of all grouped blocks is produced by applying the inverse 3D transform on the filtered coefficients. A final estimate is computed by aggregating all of the obtained local estimates using a weighted average using the technique mentioned in the first step.

BM3D is considered one of the best denoising filters. It exhibits remarkable results when compared to other remarkable results when compared to non-learning based techniques

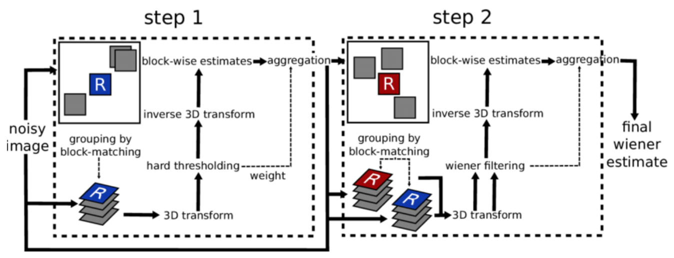


Figure 2. BM3D Steps

Data Adaptive Transformation

Some examples of data-adaptive transform [1] methods include Independent component analysis (ICA) and PCA. They involve transforming the noisy image into a different domain and modifying the transform coefficients to reduce the noise level. However, both methods require a sample of noise-free data or at least two image frames from the same scene, which can be difficult to obtain in some applications. Additionally, both methods have a high computational cost due to their use of sliding windows. Despite these drawbacks, both techniques have been shown to be effective for image denoising.

4. Machine Learning Based Methods

MLP based Methods

Some of the common MLP-based image denoising algorithms include Trainable Nonlinear Reaction-Diffusion (TNRD) proposed by Chen et al. [14], Stacked Sparse Denoising Autoencoders(SSDA) proposed by Xie et al. [15]. TNRD stands for Trainable Nonlinear Reaction-Diffusion and as the name suggests, it uses a class of partial differential equations called Reaction-Diffusion equations to solve the problem of image denoising. These Reaction-Diffusion equations are originally used to characterize the change in the concentration of one or more chemical substances as a function of space and time.

$$\frac{\partial u}{\partial t} = \frac{u_t - u_{t-1}}{\Delta t} = -\sum_{i=1}^{N_k} K_i^{t\top} \phi_i^t(K_i^t u_{t-1}) - \psi^t(u_{t-1}, f) \tag{16}$$

Here, u denotes the input image $K_i \in \mathbb{R}^{N\times N}$ denotes a sparse matrix, and it is implemented using 2D convolution between the image u and the filter kernel k_i. The N_k in the summation denotes the number of filters. We set Δt to be 1, as the functions on the RHS can be scaled freely.
The term containing the summation is called the diffusion term, and $\psi^t(u_{t-1}, f)$ is called the reaction term. A clean image is obtained from a noisy input image by evaluating the right-hand side of the equation and adding it to the current image, and this step is done recursively for a certain number of iterations, after which we get the denoised image. The functions ψ, ϕ and the filters K_i are characterized by learnable parameters which are learned by training. MSE between the final image obtained from the model and the ground truth clean image is used as the loss function for training the model.
Denoising autoencoders propose a general framework for obtaining clean data from corrupted data by using Autoencoders. First, we define an encoder and decoder, which are single fully connected layer and train it to predict the original data from the corrupted input data. Once this model is trained, we move to train the next layer, which takes the hidden layer activations of the original model as inputs to the next layer. This method is called a Stacked denoising autoencoder(SDA). We use the L2 norm of the error between the ground truth clean data and predicted data as the loss function for training.
The Stacked Sparse Denoising Auto-encoders draws overlapping samples of the images from the training set and flatten the image into a vector and feed it as input to the denoising autoencoder, like the one described previously. We use the Mean squared error between the ground truth clean image and the predicted clean image as the loss function, and this is regularized by a sparsity-inducing term(KL divergence), and the Frobenius norm of the weights is used for L2 regularization,

$$L(X,Y;\theta) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2}||y_i - \hat{y}(x_i)||^2 + \beta \mathbf{KL}(\hat{\rho}||\rho) + \frac{\lambda}{2}(||W||_F^2 + ||W'||_F^2) \tag{17}$$

where,

$$\mathbf{KL}(\hat{\rho}||\rho) = \sum_{j=1}^{|\hat{\rho}|} \rho log\frac{\rho}{\hat{\rho_j}} + (1-\rho)log\frac{1-\rho}{1-\hat{\rho_j}} \tag{18}$$

$$\hat{\rho} = \frac{1}{N}\sum_{i}^{N} h(x_i) \tag{19}$$

We choose ρ to be small so that the KL divergence term forces the mean of the hidden layer activations to be sparse.

Deep Learning Based Methods

With the advent of CNNs, deep learning models with increasing depth have been shown to provide higher performance for almost all categories of image processing tasks. In the majority of denoising algorithms developed through classical image processing assume noise to be either Gaussian noise or impulse noise, while that does not completely hold true for real-world images(blind noise), where noises introduced due to various factors are more diverse and sophisticated. Discriminative learning based on deep learning has been used to address the denoising of images having Gaussian noise, while optimization models have been introduced to effectively estimate blind noise.
The general model for deep learning-based denoising methods is formulated as below:
Assume:
x : ground truth image
$\hat{x}$: denoised image
σ : noise variance
θ : parameters of the deep learning model
y : noisy image
The deep learning model generates a mapping from the noisy image to the denoised image through function F, such that

$$\hat{x} = F(y, \sigma; \theta) \tag{20}$$

A loss function is defined to quantify the distance between the denoised image and the ground truth image, and the objective function is to find the minimum value of the loss function while varying the model parameter θ.
Deep neural networks were first introduced for image denoising task by Liang & Liu(2015) [16] They introduced stacked denoising autoencoder and dropout to achieve better performance than the single dropout method while reducing performance time. Stamatios Lefkimmiatis [17] proposed a CNN model to carry out non-local color image denoising, which would benefit from discriminative learning while performing non-local processing. The model exploits the non-local self-similarity property of natural images to efficiently remove color-image noise.
Zhang et al. [18] proposed a deep convolutional neural network for image denoising, where they implemented residual learning to provide separation of noise from noisy im-

ages. They implemented batch normalization along with residual learning and showed that it resulted in speeding up of training process while simultaneously boosting denoising performance. The proposed feed-forward denoising CNNs (DnCNNs) were shown to have the capacity to handle both blind noise (real-world noise) as well as Gaussian noise, which is a big improvement over traditional discriminative approaches. their experiments have shown that DnCNN model inference has comparatively better run time through GPU utilization while improving denoising performance both qualitatively and quantitatively. One drawback of their approach was, although a trained DnCNN can handle both compression and interpolation errors, it performed poorly for noise variances other than the trained noise variance.

The authors model the noise to be additive and train a CNN (called DnCNN model) that accepts a noisy image and outputs an image containing the noise present in each pixel(called residual image). MSE between the estimated residual image and the actual residual image is used as the loss function. Say the depth of the CNN is D. If the input image dimensions are $H \times W \times C$, then the first layer is 2D convolution layer containing 64 filters of size $3 \times 3 \times C$ followed by Relu activation function, and from 2nd layer to $(D-1)^{th}$ layer are 2D convolutional layer containing 64 filters of size $3 \times 3 \times 64$ followed by batch norm and Relu. The last layer is a 2D convolutional layer containing C filters of size $3 \times 3 \times 64$. In order to retain the size of the original image, zero padding is done before applying 2D convolutions. The method of outputting the residual image is referred to as residual learning in this paper. This paper also discusses how batch normalization and residual learning help each other in making the model perform better. The authors have trained four models, with and without batch normalization, with and without residual learning, and plotted the average PSNR of the models against the no. of epochs. The plots clearly show that the model trained with batch normalization and residual learning had better average PSNR and faster convergence. This model is further extended to work in other image restoration problems like SISR(Single Image Super Resolution) and JPEG deblocking. The authors also train this model in three variations. One variation is that the model is trained on images with Gaussian noise of specific noise levels(Noise variance); another variation is that the model is trained on images with Gaussian noise of varying noise levels. This task is called the blind fold image denoising task. In the last variation, the authors train one model for performing all three image restoration tasks mentioned above. This model gives better average PSNR values compared to other competitive models.

For unpaired noisy images (where ground truth image is not available), Chen et al. [19] proposed a generative adversarial network (GAN) CNN blind denoiser (GCBD). To tackle the lack of paired training data, they proposed a two-step framework where in the first step, a trained GAN model would estimate noise distribution over the input images and in the second step, noise patches sampled from the first step would be utilized to construct a paired training dataset, which would be used for training a deep CNN for denoising. They showed that the GCBD model performed very well as compared to any other model in blind denoising tasks.
To address the issue of requiring multiple models for denoising images with different noise levels, Zhang et al. [20] proposed FFDNet (Fast and flexible denoising convolutional neural network), with a tunable noise level map as the input. It leverages the processing of down-sampled sub-images to achieve a good trade-off between inference speed and denoising performance. The authors have shown that the model could handle a wide range of noise levels using only a single network while significantly reducing spatially variant noise through a tunable noise level map. Modifying Eq (5) above, FFDNet is modelled as :

$$\hat{x} = F(y, M; \theta) \tag{21}$$

where M is a tunable noise level map.
To improve the generalization capability of the CNN models while working on real-world images with blind noise, Guo et al. [21] proposed training a convolutional blind denoising network (CBDNet) with a more realistic noise model and real-world noisy-clean image pairs. They showed that the denoising performance of a network can be boosted by incorporating both synthetic and real noisy images in training. Their utilization of two sub-networks to estimate noise from real noisy images and to obtain latent clean images resulted in the efficient utilization of asymmetric loss to improve the generalization ability of the model to real-world noise.

To tackle the issue of blind denoising, Zhao et al. [22] proposed a novel pyramid real image denoising network (PRIDNet), which contains three stages. First, the noise estimation stage uses the channel attention mechanism to recalibrate the channel importance of input noise. Second, at the multi-scale denoising stage, pyramid pooling is utilized to extract multi-scale features. Third, the stage of feature fusion adopts a kernel selecting operation to adaptively fuse multi-scale features. The author's experiments on two datasets of real noisy photographs demonstrate that this approach can achieve competitive performance in comparison with state-of-the-art denoisers in terms of both quantitative measure and visual perception quality.

To advance the practicability of denoising algorithms, Saeed Anwar et al. [23] proposed a novel single-stage blind real image denoising network (RIDNet) by employing a modular architecture.

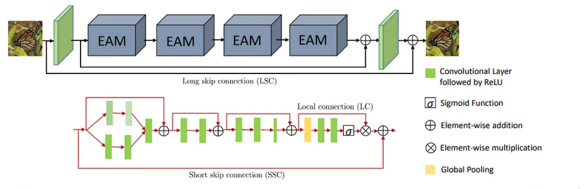

EAM
EAM
EAM
EAM
Long skip connection (LSC)
Local connection (LC)
Short skip connection (SSC)
Convolutional Layer followed by ReLU
Sigmoid Function
Element-wise addition
Element-wise multiplication
Global Pooling

Figure 2. The architecture of the proposed network. Different green colors of the conv layers denote different dilations while the smaller size of the conv layer means the kernel is 1×1. The second row shows the architecture of each EAM.

Figure 3. RIDNet network architecture

The model comprises of feature extraction module, a feature learning residual on the residual module and a reconstruction module. They have implemented cascaded "enhancement attention modules" (EAM) to provide a wide receptive field through kernel dilation. EAM uses a residual on the residual structure with local skip and short skip connections. The model uses Long Skip Connections (LSC), short skip connections (SSC) and Local Connections (LC) to allow low-frequency information to bypass so the network can focus on residual learning. The authors have tested the model extensively to demonstrate the effectiveness of the model.

Recent works on image denoising have utilized deep reinforcement learning for the restoration of images with diverse or unknown noises. Residual Recovery using Reinforcement Learning (R3L) [24] was proposed to achieve better generalizability and robustness in image denoising when the estimated noise level varies.

Another class of neural architectures, Transformers [25], has shown significant improvement in performance on high-level vision tasks. A modified transformer model called Restormer (Restoration Transformer) [26] was proposed that uses several key designs in the building blocks (multi-head attention and feed-forward network) to capture long-range pixel interactions while still remaining applicable to large images. The authors showed that Restormer achieved much better PSNR on several benchmark datasets for grayscale and color image denoising while reducing computational complexity with respect to existing methodologies while achieving better performance on real denoising tasks, generating clean images without compromising fine texture.

Combining both CNN and transformer models [25],a kernel basis network(KBNet) was proposed to design a multi-axis feature fusion (MFF) block to encode and fuse channel-wise, spatial-invariant and pixel-adaptive features. The kernel basis attention (KBA) module adopts the learnable kernel bases to model the local image patterns and fuse kernel bases linearly to aggregate the spatial information efficiently. KBNet is shown to achieve state-of-the-art results on popular synthetic noise datasets CBSD68 [27], Kodak24 [28], McMaster [29] and Urban100 [30] and real-world noise datasets SIDD [31] and SenseNoise Dataset [32].

5. Metrics of denoising performance

The performance of image denoising methodology is measured through two quantitative measurements, PSNR and SSIM [33]. Given the original , noise-free image x and denoised image $\hat{x}$ we can calculate Peak Signal to Noise Ratio(PSNR) as:

$$PSNR(x,\hat{x}) = 10.log_{10}\left(\frac{255^2}{\|x-\hat{x}\|_2^2}\right) \tag{22}$$

theoretically, Higher the PSNR value, the better the denoising performance. Although PSNR is easier to calculate, it does not have a very high correlation with human visual perception.

Hence, we use another metric called Structual Similarity Index Measure (SSIM).It is a modern and successful approach that measures the loss of structure in a distorted image. It takes into consideration local measures of similarity of luminance, contrast and structure and then performs a weighted average of local measures across images. The equation for SSIM is given by:

$$SSIM(x,\hat{x}) = \frac{(2\mu_x\mu_{\hat{x}} + C_1)(2\sigma_{x\hat{x}} + C_2)}{(\mu_x^2 + \mu_{\hat{x}}^2 + C1)(\sigma_x^2 + \sigma_{\hat{x}}^2 + C_2)} \tag{23}$$

where:
$\mu_x, \mu_{\hat{x}}$: means of x and $\hat{x}$
$\sigma_x, \sigma_{\hat{x}}$: variances of x and $\hat{x}$
$\sigma_{x\hat{x}}$: covariance between x and $\hat{x}$
C_1, C_2: stabilizing constants

6. Algorithms implemented and dataset used

We have carried out a thorough review of multiple classical image denoising algorithms. Wiener filter [2], bilateral filtering [3], PCA [34] method and NLM (Non-Local Means) [4] have been implemented from the basic pseudo code available in the paper or in review papers. NLM basic implementation resulted in significant computational overhead, and hence opencv built-in function [6] is used for analysis. We have used [35] for the implementation of wavelet-based image denoising program.We have used the official implementation of BM3D [36] and built-in functions of kornia python library [37] for implementation of Total Variation denoising. We have used [38] for the implementation of Weighted Nuclear Norm Minimization (WNNM).

We have used CBSD68 [27] dataset as the common dataset to test our implementations and carry out a comparison of image denoising algorithms. CBSD68 dataset consists of 68 original PNG images and their noisy versions with various levels of AWGN added. For our analysis, we have considered noisy15 and noisy25 subsets of noisy images.

To carry out deep learning based implementation, we have used 400 images from the BSD400 dataset [39]. After the model is trained, the PSNR and SSIM values are tested on noisy15 and noisy25 subsets of noisy images from the CBSD68 dataset.

We have taken the basic architecture for Autoencoder-based denoising model from [40], added Batch Normalization and Dropout layers and used the LeakyReLU activation function. We also used KFold training with 5 splits to chose the best-performing model on the test dataset.

We have used implementation of RIDNet from [40], used LeakyReLU instead of ReLU activation function and added dropout layers and we are able to get SSIM values of 0.937 and 0.828 for noisy15 and noisy25 dataset respectively, which are a significant improvement upon other implementations.

Implementation of CBDNet and PRIDNet has been taken from [40]. DnCNN implementation carried out in [41] is used.

7. Results and discussions

7.1. Classical denoising method comparison

PSNR and SSIM values for all 68 images for different algorithms are shown below for both noise15 and noise25 datasets.

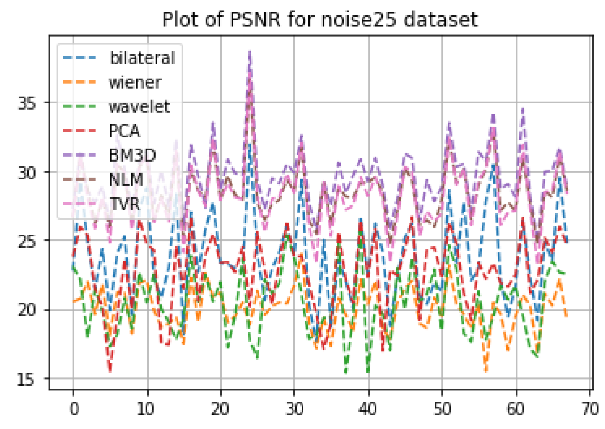


Figure 4. PSNR comparison for noise25 dataset

As can be seen from the plots, BM3D and NLM performed much better as compared to other algorithms. There are some fluctuations in SSIM and PSNR values between different images for the same algorithm. We will calculate the mean and variance of these values to see the performance consistency. Comparison of mean and standard deviation for SSIM and PSNR values for noise25 and noise15 datasets is presented in Table 1 and Table 2:

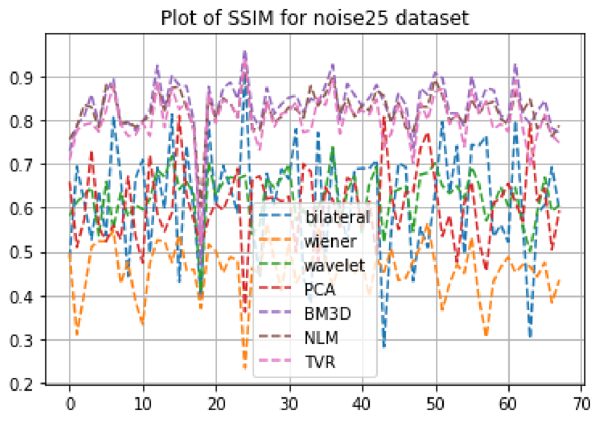


Figure 5. SSIM comparison for noise25 dataset

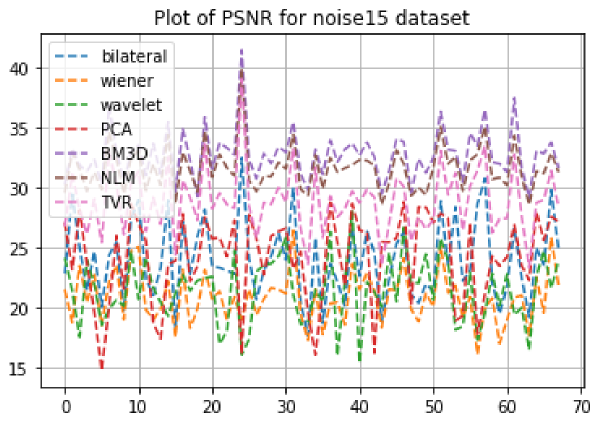


Figure 6. PSNR comparison for noise15 dataset

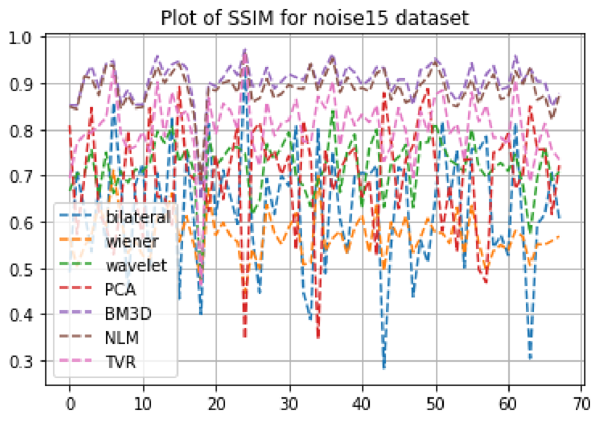


Figure 7. SSIM comparison for noise15 dataset

From the above two tables, it can be clearly seen that, BM3D and NLM denoising method performs more uniformly over a varied dataset as compared to other image denoising techniques. ALso, BM3D and NLM denoising

Table 1. SSIM and PSNR comparison for noise25 dataset

Algorithm	SSIM Mean	SSIM std	PSNR mean	PSNR std
bilateral	0.617	0.128	23.781	3.189
Wiener	0.458	0.061	20.014	1.637
wavelet	0.625	0.058	20.450	2.458
PCA	0.531	0.131	22.748	2.669
BM3D	0.835	0.064	29.574	2.339
NLM	0.822	0.045	28.476	2.132
TVR	0.803	0.059	28.298	2.484

Table 2. SSIM and PSNR comparison for noise15 dataset

Algorithm	SSIM Mean	SSIM std	PSNR mean	PSNR std
bilateral	0.629	0.133	23.954	3.297
Wiener	0.569	0.050	20.809	2.154
wavelet	0.720	0.060	21.420	2.766
PCA	0.695	0.114	24.156	3.621
BM3D	0.905	0.044	32.688	2.110
NLM	0.888	0.040	31.553	1.897
TVR	0.804	0.070	28.801	2.731

methods have higher average PSNR and average SSIM values as compared to other denoising algorithms.
A comparison of all the seven algorithms on a Lena image corrupted by AWGN having a standard deviation of 30 is shown below.

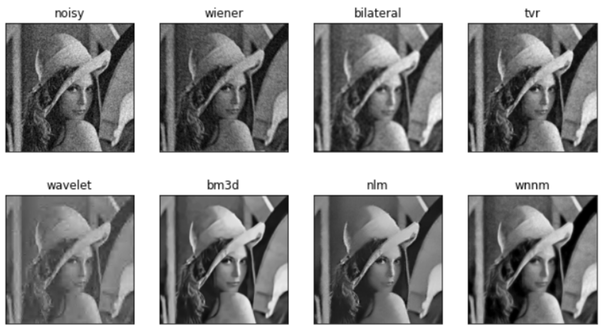


Figure 8. comparative results of different classical algoirthms

As can be seen from Figure 8, although bilateral filtering does much better than others, it does aggressive smoothing, and hence the image is blurred as compared to others. Wavelet-based image denoising method is able to find a trade-off between blurring and denoising. BM3D and NLM provide much smoother performance, while BM3D retains edges better than NLM method.

7.2. Deep learning based denoising method comparison

Table 3 and Table 4 tabulates SSIM mean, SSIM standard deviation, PSNR mean and PSNR standard deviation values for both noisy15 and noisy25 dataset for various deep learning algorithms. We can observe that RIDNet and DnCNN provides much better performance as compared to other algorithms. Application of all the deep learning-based models on a noisy image is shown in Figure 9.

Table 3. SSIM and PSNR comparison for noise15 dataset

Algorithm	SSIM	SSIM std	PSNR	PSNR std
Autoencoder	0.851	0.062	25.221	2.712
DnCNN	0.891	0.038	31.74	2.42
RIDNet	0.937	0.017	14.011	1.251
CBDNet*	0.685	0.103	26.495	0.778
PRIDNet*	0.560	0.075	23.264	1.757

Table 4. SSIM and PSNR comparison for noise25 dataset

Algorithm	SSIM	SSIM std	PSNR	PSNR std
Autoencoder	0.831	0.065	24.823	2.562
DnCNN	0.828	0.058	29.23	2.58
RIDNet	0.910	0.017	13.586	1.069
CBDNet*	0.528	0.125	23.20	0.354
PRIDNet*	0.428	0.086	21.361	1.121

* CBDNet and PRIDNet implementation takes considerable amount of time for completing one epoch in the available hardware setup. Due to a lack of higher computing resource, full cycle of training was not carried out and that is reflected in the lower-than-expected SSIM numbers.

Having gone through various classical and deep learning algorithm to solve image denoising problem, we can conclude that while classical approaches like BM3D achieves very good performance, deep learning-based models such as DnCNN, RIDNet achieves state of the art performance on both AWGN and blind noise dataset.

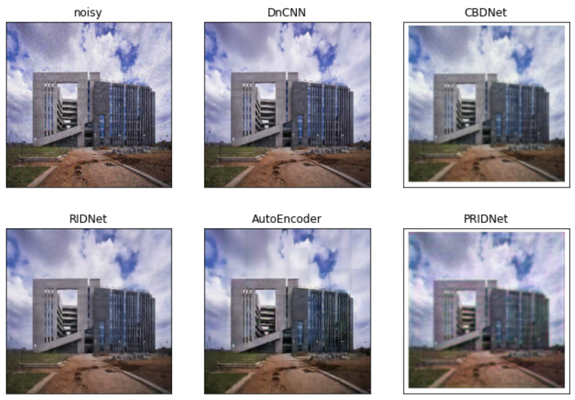
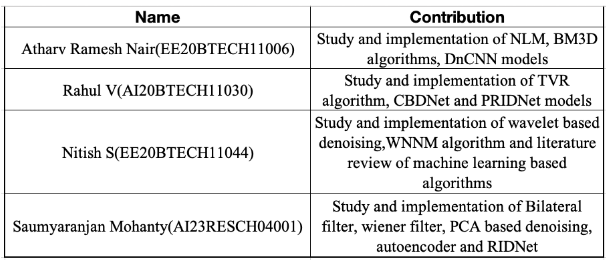


Figure 9. Comparative analysis of the performance of deep learning models

7.3. Work carried out by team members



Name	Contribution
Atharv Ramesh Nair(EE20BTECH11006)	Study and implementation of NLM, BM3D algorithms, DnCNN models
Rahul V(AI20BTECH11030)	Study and implementation of TVR algorithm, CBDNet and PRIDNet models
Nitish S(EE20BTECH11044)	Study and implementation of wavelet based denoising,WNNM algorithm and literature review of machine learning based algorithms
Saumyaranjan Mohanty(AI23RESCH04001)	Study and implementation of Bilateral filter, wiener filter, PCA based denoising, autoencoder and RIDNet

Figure 10. Contribution by each member.

References

[1] Linwei Fan, Fan Zhang, Hui Fan, and Caiming Zhang. Brief review of image denoising techniques. *Vis. Comput. Ind. Biomed. Art*, 2(1):7, July 2019. 1, 5

[2] Jacob Benesty, Jingdong Chen, and Yiteng Huang. Study of the widely linear wiener filter for noise reduction. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 205–208, 2010. 2, 8

[3] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 839–846, 1998. 2, 8

[4] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65 vol. 2, 2005. 2, 3, 8

[5] Jacques Froment. Parameter-Free Fast Pixelwise Non-Local Means Denoising. *Image Processing On Line*, 4:300–326, 2014. https://doi.org/10.5201/ipol.2014.120. 3

[6] OpenCV: Denoising — docs.opencv.org. https://docs.opencv.org/3.4/d1/d79/group__photo__denoise.html. [Accessed 03-Apr-2023]. 3, 8

[7] Leonid Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60:259–268, 11 1992. 3

[8] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2862–2869, 2014. 4

[9] S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989. 4

[10] D.L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995. 4

[11] Hossein Rabbani and Saeid Gazor. Wavelet denoising of medical images using thresholding: A new adaptive thresholding scheme based on noise estimation. *IEEE Transactions on Medical Imaging*, 23(12):1430–1444, 2004. 4

[12] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. 4

[13] Marc Lebrun. An analysis and implementation of the bm3d image denoising method. *Image Processing On Line*, 2:175–213, 08 2012. 4

[14] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1256–1272, 2017. 5

[15] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 5

[16] Stamatios Lefkimmiatis. Non-local color image denoising with convolutional neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, July 2017. 6

[17] Jianglin Liang and Ruifang Liu. Stacked denoising autoencoder and dropout together to prevent overfitting in deep neural network. In *2015 8th International Congress on Image and Signal Processing (CISP)*. IEEE, October 2015. 6

[18] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Trans. Image Process.*, 26(7):3142–3155, July 2017. 6

[19] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 2018. 7

[20] Kai Zhang, Wangmeng Zuo, and Lei Zhang. FFDNet: Toward a fast and flexible solution for CNN based image denoising. *IEEE Trans. Image Process.*, 27(9):4608–4622, May 2018. 7

[21] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2019. 7

[22] Yiyun Zhao, Zhuqing Jiang, Aidong Men, and Guodong Ju. Pyramid real image denoising network. In *2019 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, December 2019. 7

[23] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3155–3164, 2019. 7

[24] Rongkai Zhang, Jiang Zhu, Zhiyuan Zha, Justin Dauwels, and Bihan Wen. R3l: Connecting deep reinforcement learning to recurrent neural networks for image denoising via residual recovery. In *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, September 2021. 8

[25] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-Trained image processing transformer. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2021. 8

[26] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2022. 8

[27] D Martin, C Fowlkes, D Tal, and J Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. IEEE Comput. Soc, 2002. 8

[28] True Color Kodak Images — r0k.us. https://r0k.us/graphics/kodak/. [Accessed 13-Mar-2023]. 8

[29] Xiaolin Wu. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic Imaging*, 20(2):023016, April 2011. 8

[30] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2015. 8

[31] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 2018. 8

[32] Yi Zhang, Dasong Li, Ka Lung Law, Xiaogang Wang, Hongwei Qin, and Hongsheng Li. IDR: Self-supervised image denoising via iterative data refinement. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2022. 8

[33] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, April 2004. 8

[34] D D Muresan and T W Parks. Adaptive principal components and image denoising. In *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*. IEEE, 2004. 8

[35] GitHub - gopi231091/Wavelet-Transform-and-Application-to-Image-Denoising — github.com. https://github.com/gopi231091/Wavelet-Transform-and-Application-to-Image-Denoising. [Accessed 03-Apr-2023]. 8

[36] bm3d — pypi.org. https://pypi.org/project/bm3d/. [Accessed 03-Apr-2023]. 8

[37] Denoise image using total variation x2014; Kornia documentation — kornia.readthedocs.io. https://kornia.readthedocs.io/en/v0.5.0/tutorials/total_variation_denoising.html. [Accessed 03-Apr-2023]. 8

[38] GitHub - kanavalau/EE367_Project_WNNM_denoising: Stanford EE367 (Computational Imaging) Final Project: Implementation of the WNNM for image denoising. — github.com. https://github.com/kanavalau/EE367_Project_WNNM_denoising. [Accessed 03-Apr-2023]. 8

[39] github. Bsd400 dataset. https://github.com/saumya221/image_denoising_project/tree/main/Datasets/BSD400. [Accessed 29-Apr-2023]. 9

[40] Sharath Solomon. Image Denoising using Deep Learning — medium.com. https://medium.com/analytics-vidhya/image-denoising-using-deep-learning-dc2b19a3fd54. [Accessed 28-Apr-2023]. 9

[41] GitHub - cszn/DnCNN: Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising (TIP, 2017) — github.com. https://github.com/cszn/DnCNN. [Accessed 29-Apr-2023]. 9