UNIVERSITÀ DI PISA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Dottorato di Ricerca in Ingegneria dell'Informazione

Computational Imaging and Vision - Final assessment
Francesco Mancuso, cycle XXXVII

Pisa, 24/04/2023

Contents

1 Image denoising **1**

2 Analysed techniques **2**
2.1 Classical methods 2
2.2 Denoising Autoencoder (DAE) 3

3 Results **3**

THE CODE IS AVAILABLE IN THE FOLLOWING GITHUB REPOSITORY:
HTTPS://GITHUB.COM/MANDUGO/COMPUTATIONAL-IMAGING-VISION.

1 Image denoising

Image denoising is a popular problem in image processing and computer vision that involves removing noise or unwanted artifacts from an image to improve its quality and enhance its visual appearance. It has wide-ranging applications in various fields, including medical imaging, remote sensing, surveillance, automotive imaging, and consumer photography, where high-quality images are desired for further analysis, visualization, or human perception. Noise in images can arise from various sources, such as sensor noise in digital cameras, compression artifacts in JPEG images, or degradation during image acquisition, transmission, or processing. The goal is to **restore the original, noise-free image from a noisy version**.

There are various approaches to image denoising, including classical methods such as spatial filtering and wavelet denoising, as well as more advanced techniques based on machine learning and deep learning. Machine learning-based methods typically involve training a model on a large dataset of noisy and clean image pairs to learn the underlying mapping from noisy to clean images. Deep learning-based approaches, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have shown great success in image denoising, achieving state-of-the-art performance in many cases.

In this study, I will be discussing various methodologies for addressing the challenge of image denoising. The outcomes obtained from these approaches will be compared using visual assessments as well as appropriate evaluation metrics.

2 Analysed techniques

The aim of this report is to compare the performance of the following methods:

1. **Classical methods**
    - Wavelet denoising
    - Non-Local Means (NLM) denoising
    - Bilateral filtering
2. **Denoising Autoencoder (DAE)**

The **MNIST database**[1] of handwritten digits has been utilized to analyze and compare the different techniques. MNIST is a widely used dataset in the field of machine learning and computer vision, consisting of 70.000 grayscale images of handwritten digits ranging from 0 to 9. The dataset is divided into a training set with 60,000 images and a test set with 10.000 images. Each image is of size 28×28 pixels and is labeled with the corresponding digit it represents. MNIST is commonly used as a benchmark dataset for training and evaluating image classification algorithms, making it a suitable choice for evaluating the performance of the different techniques in this study.

2.1 Classical methods

Wavelet denoising works by decomposing an image into different scales or levels using a wavelet transform. The wavelet coefficients at each scale represent the image details or features at different levels of resolution. The higher-scale coefficients usually represent the larger and global features of the image, while the lower-scale coefficients capture the smaller and local details.

Non-Local Means (NLM) has been widely used in various image processing applications, and it has been shown to be effective in preserving image details while reducing noise, making it a popular choice for denoising tasks. The idea behind NLM is to exploit the redundancy in natural images, where similar patches appear in different regions of the image. Instead of only considering the local neighborhood of a pixel, as in traditional filters, NLM takes into account the similarity between patches at different locations in the image.

Bilateral filtering is particularly useful for preserving edges and details in images while reducing noise. The bilateral filter is based on the concept of weighted averaging of pixel values in a local neighborhood. Unlike traditional linear filters that only consider the intensity values of neighboring pixels, bilateral filtering takes into account both the intensity differences and spatial distances between neighboring pixels. This allows the filter to preserve edges and details, while attenuating noise.

[1] http://yann.lecun.com/exdb/mnist/

2.2 Denoising Autoencoder (DAE)

Autoencoders are a type of unsupervised neural networks that are specifically designed to learn a compressed summary or representation of data. This representation should ideally contain enough information to reconstruct the original data, thus enabling data recovery. By obtaining this compressed summary, autoencoders can facilitate tasks such as data compression, data generation, and data denoising. When applied to image data, an autoencoder first encodes the image into a lower-dimensional representation and then decodes that representation back to reconstruct the original image. The autoencoder typically comprises two main components:

- **Encoder**: downsamples the input data into a lower-dimensional representation;
- **Decoder**: reconstructs the original data from the lower-dimensional representation.

The lower-dimensional representation obtained from the encoder is often referred to as the latent space representation. Similar to other types of autoencoders, the **Denoising Autoencoder (DAE)** is designed to learn the underlying structure or summary of data. It aims to accurately infer this underlying structure in such a way that it can effectively remove most, if not all, of the corruption in the data when presented with a noisy or corrupted version of the data. The DAE is trained to effectively denoise the data by learning to reconstruct the original, clean version of the data from the noisy or corrupted input.

Notes on the implementation

The implementation was carried out in **Python**, with the classical methods implemented using the reference material from ***scikit-image***[2] library, and the DAE taken from a guide by **Saul Dobilas**[3]. The code for evaluating the Structural Similarity Index is a slightly modified version of a function from a **GitHub collection of image denoising techniques**[4].

3 Results

Three different metrics have been used to quantitatively evaluate the performance of different methods. It's important to use a combination of these metrics to get a comprehensive assessment of the quality of the denoised images and understand the effectiveness of different denoising methods.

[2]https://scikit-image.org/docs/dev/api/skimage.restoration.html

[3]https://towardsdatascience.com/denoising-autoencoders-dae-how-to-use-neural-networks-to-clean-up-your-data-cd9c19bc6915

[4]https://github.com/wenbihan/reproducible-image-denoising-state-of-the-art#commonly-used-image-quality-metrics

The **Root Mean Squared Error (RMSE)** [5], which is calculated by taking the square root of the average of the squared differences between predicted values and actual values:

$$RMSE(y,\hat{y}) = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}(y_i - \hat{y}_i)^2} \tag{1}$$

where y indicates the original image, $\hat{y}$ is the denoised twin, and the subscript index i refers to the i-th pixel. The RMSE is evaluated for each couple of images and then is averaged for all the dataset. A lower RMSE value indicates better accuracy, while a higher RMSE value indicates higher prediction errors.

The **Peak signal-to-noise ratio (PSNR)** [6] measures the ratio between the peak signal level (i.e., the maximum possible pixel intensity) and the mean squared error (MSE) between the denoised image and the original image:

$$PSNR(I) = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE_I}\right) = 20 \cdot \log_{10}\left(\frac{MAX_I}{RMSE_I}\right) \tag{2}$$

where MAX_I is the maximum possible pixel intensity (usually 255 for 8-bit grayscale images, in our case is normalized to 1), and RMSE is the Root Mean Squared Error between the denoised image and the original image. The PSNR is evaluated for each couple of images and then is averaged for all the dataset.

The **Structural Similarity Index (SSIM)** [7] measures the structural similarity between a reference image and a distorted image, taking into account luminance, contrast, and structural information. It provides a more comprehensive assessment of image quality compared to metrics like PSNR and RMSE, as it takes into account not only pixel-level differences but also structural similarities between images. The mathematical definition of the SSIM can be found in literature, it has been omitted here for the sake of brevity. The SSIM index is a decimal value ranging from -1 to 1, where 1 indicates perfect similarity, 0 indicates no similarity, and -1 indicates perfect anti-correlation. The SSIM is evaluated for each couple of images and then is averaged for all the dataset.

Table 1 presents the results of the experiment. The first column shows the metrics evaluated for the noisy images, serving as a reference. The DAE exhibits the best performance among the analyzed techniques, outperforming the others in all metrics (as visually confirmed in Figure 1). The other three techniques show a reduction in RMSE and an increase in PSNR compared to the reference, but the SSIM values remain relatively unchanged. The Non-Local Means denoiser is the second best performing approach. The Figure 1 visually confers the obtained numbers. The images denoised by the DAE are almost identical to the original ones. The NLM-denoised

[5] https://en.wikipedia.org/wiki/Root-mean-square_deviation
[6] https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio
[7] https://en.wikipedia.org/wiki/Structural_similarity

Table 1: Results of the experiment

	Noisy	Wavelet	NLM	Bilateral	DAE
RMSE	0.274	0.2172	0.1841	0.2396	0.0917
PSNR	11.25 dB	13.27 dB	14.72 dB	12.42 dB	20.93 dB
SSIM	0.29	0.29	0.29	0.3	0.66

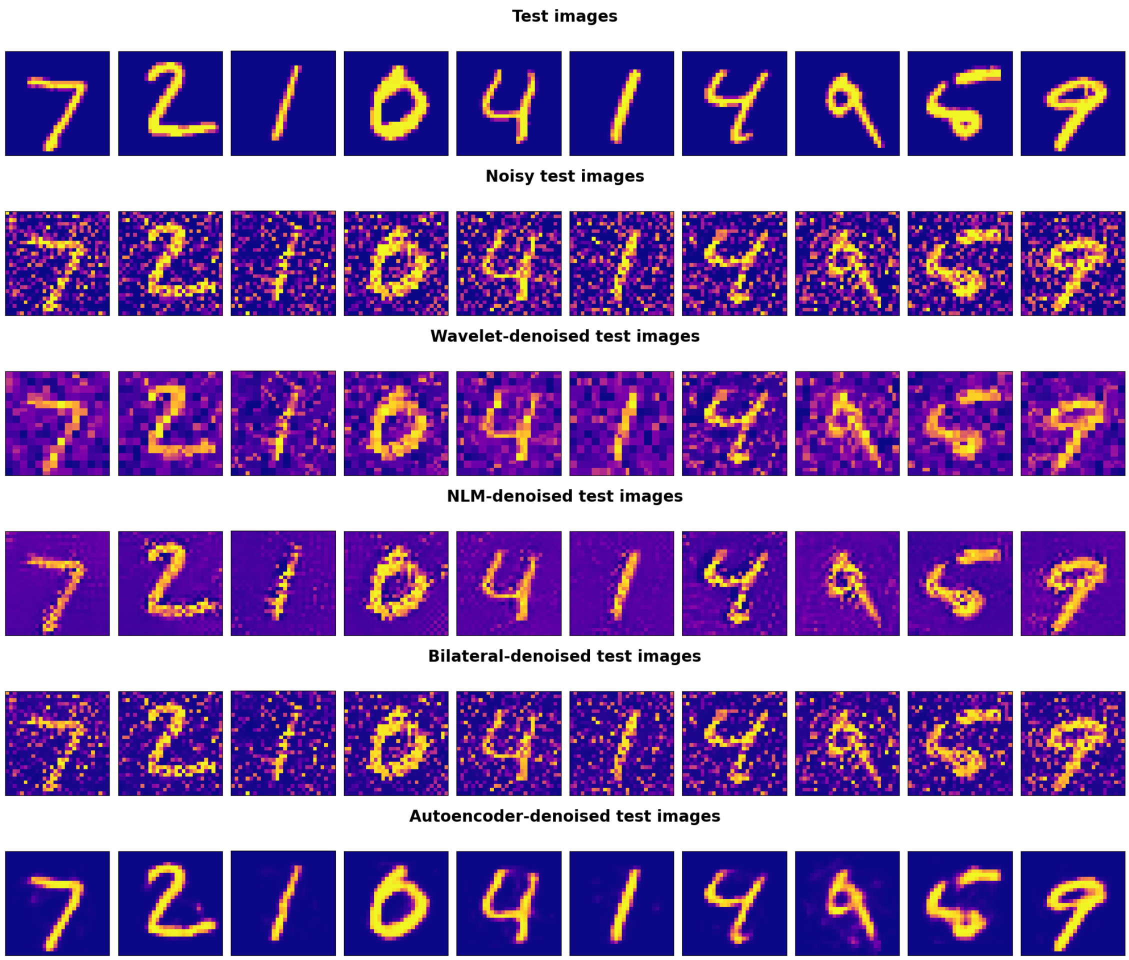

Figure 1: Results of the denosing procedures.

images achieve a good smoothing of the noise and the shape of the digits is preserved. From a computational perspective, classical approaches tend to be faster than autoencoders, if we consider the training time for neural networks. Among the classical methods, Bilateral filtering can have longer processing times, especially when the extent of spatial averaging is increased.