Computer Vision

Image Denoising

Course Instructor:
Dr. Suman Kumar Maji

Denoising

Denoising consists in reducing noise in an image.

Note that it is often not possible to completely cancel the noise.

We start this section by listing the most common noise models, then we present some denoising methods.

Noise sources

- The main sources of noise in digital images are during the acquisition (quantity of photons collected too low, sensor temperature...) or during any transmission (echoes and atmospheric distortions in wireless communication).
- In some cases, noise is also considered to model the inaccuracies in the mathematical model of image formation, the latter being necessarily different compared to reality, like any physical model!
- The noise is by nature a random phenomenon, it is modelled by a probability density which represents the intensity distribution of the noise.
- In the following, we denote by y the observed (and noisy) image, b the noise and x the non-noisy image.

Additive Gaussian white noise

- Additive white Gaussian noise (AWGN) models each pixel (m,n) of the observation y by the sum of the pixel of the noiseless image x and of a pixel of the noise b:

$$\forall\, m, n \quad y(m,n) = x(m,n) + b(m,n)$$

where,

$$b(m,n) \sim \mathcal{N}(0, \sigma^2)$$

This model is simple and facilitates calculations. It is used in most applications, including photography.

Poisson Noise

- Poisson noise (also called shot noise) models the acquisition of photons on a photosite.
- The number of photons is random and depends on the illumination.
- The corresponding Poisson process has a mean equals to the illumination.
- The intensity of each pixel (m,n) of the observation y is:

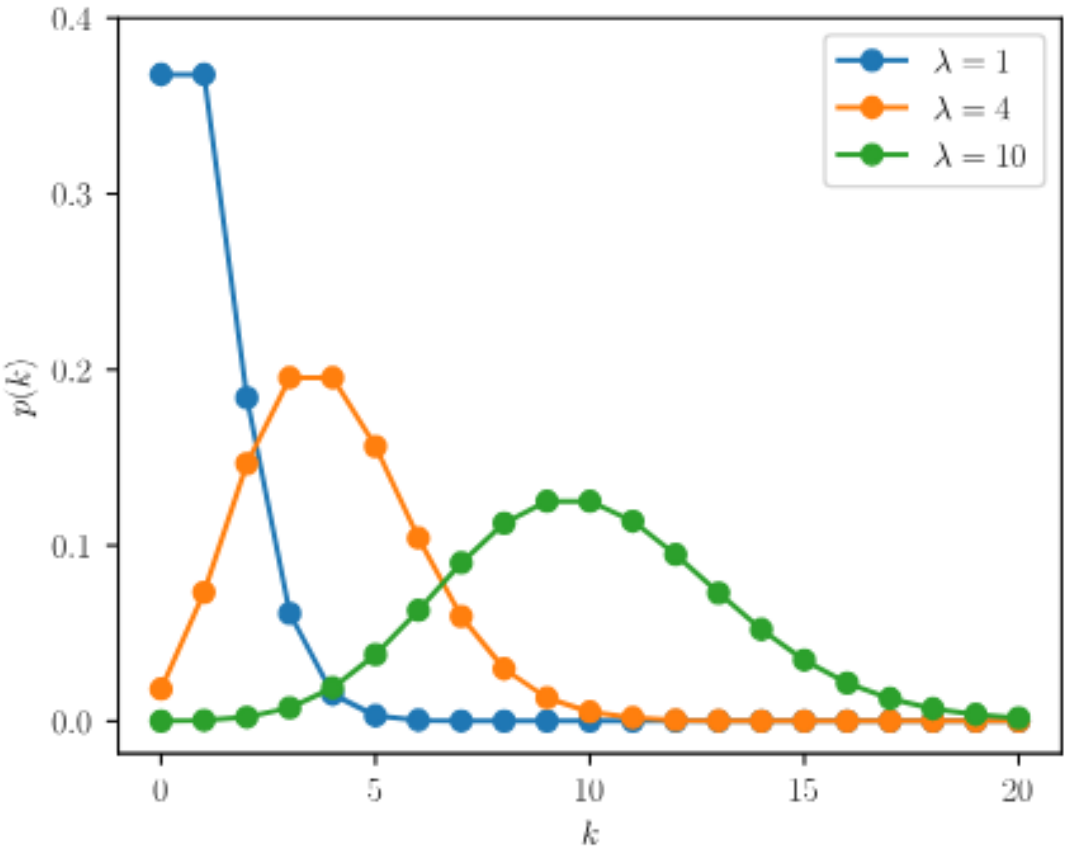
$$\forall\, m, n \qquad y(m,n) \sim \mathcal{P}\big(x(m,n)\big).$$

This model is used in the case of acquisitions with a low number of photons, for example in astronomy.

Note:

- As a reminder, the Poisson distribution $\mathcal{P}(\lambda)$ writes

$$p(k) = \frac{\lambda^k}{k!} e^{-\lambda}$$



Poisson distribution for three values of λ

cont...

- The mean and variance of the Poisson distribution are both equal λ, which depends on the number of incident photons.
- So, the noise b depends on the noiseless image x.
- Moreover, when λ increases, the Poisson distribution tends towards a Gaussian distribution, implying that AWGN becomes a good model of Poisson noise, provided that enough photons are collected.

Salt-and-pepper noise

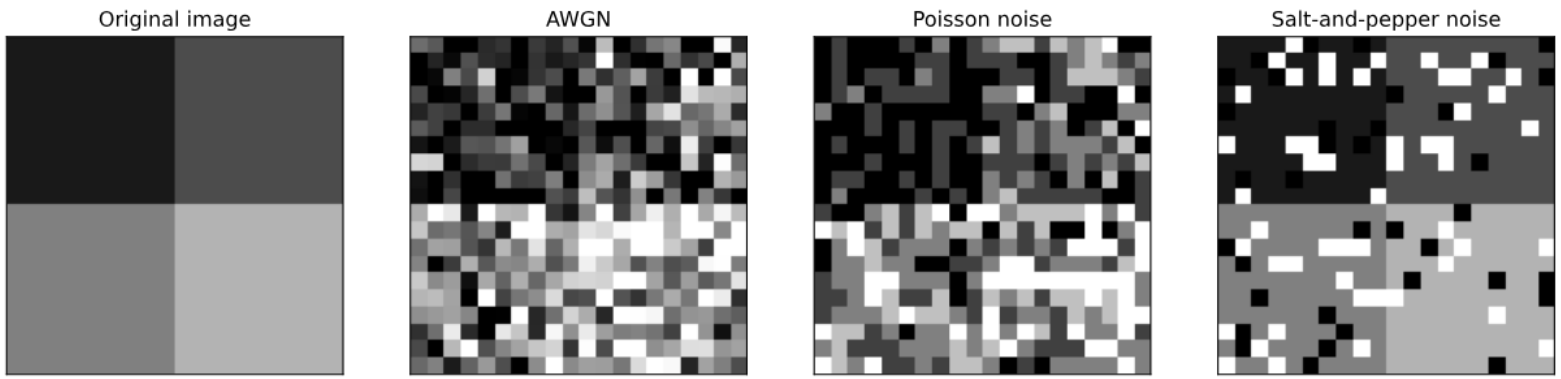
- Salt-and-pepper noise, also called less poetically impulse noise, models saturated or dead pixels (due to photosite malfunction or saturation).

$$\forall\, m, n \quad y(m,n) = \begin{cases} x_{\min} & \text{with probability } p_{\min}, \\ x_{\max} & \text{with probability } p_{\max}, \\ x(m,n) & \text{with probability } 1 - p_{\min} - p_{\max}. \end{cases}$$

where, $x_{\min}$ and $x_{\max}$ are the intensity minimum and maximum.

Noise power

- One can observe that:
  1. for Gaussian noise, the whole image is affected in the same way by the noise,
  2. for Poisson noise, the lighter parts are noisier than the dark parts,
  3. for impulse noise, only a few pixels are modified and they are replaced by black or white pixels.
- Following figure illustrates the effect of the previous noises on the same image:



Example of different types of noise (with almost the same power)

Signal to noise ratio (SNR)

- Signal-to-noise ratio (SNR) is a measure of the noise level.
- It is defined as the ratio between the power of the non-noisy image over the power of the noise, where the power of an image x is defined by:

$$P_x = \frac{1}{M \times N} \sum_{m,n} x(m,n)^2$$

- Because SNR is most often expressed on a logarithmic scale (unit: decibel), it is also defined as:

$$\text{SNR} = 10 \log_{10} \left(\frac{\sum_{m,n} x(m,n)^2}{\sum_{m,n} b(m,n)^2} \right)$$

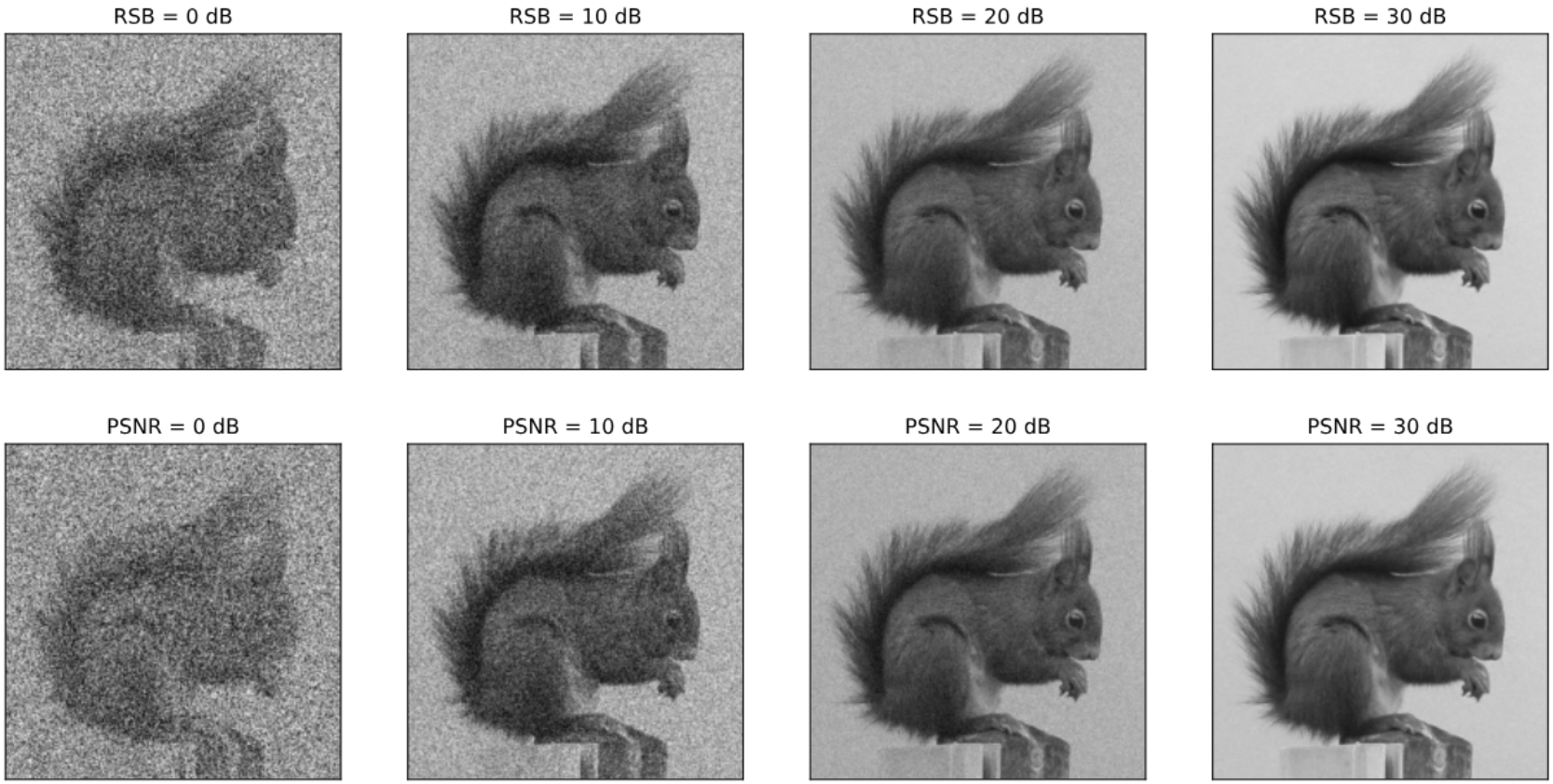
Peak signal to noise ratio(PSNR)

- For additive noise, another measure exists: the **peak signal-to-noise ratio (PSNR)** is the ratio of the squared dynamics of the non-noisy image (difference between maximum intensity and minimum intensity) to the power of the noise:

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{\Delta x^{\,2}}{\frac{1}{M \times N} \sum_{m,n} b(m,n)^2} \right)$$

cont...

The following image represents the same image corrupted with additive white Gaussian noise, at different SNR and PSNR. When the RSB or the PSNR increases, the noise decreases.



Noisy image at different RSB and PSNR

Mean filter

- The mean filter is a very simple denoising method. Each pixel (m,n) of the denoised image $\widehat{x}$ is the average of the pixels of the noisy image y around (m,n):

$$\forall m,\ n \quad \widehat{x}(m,n) = \frac{1}{|V_{m,n}|} \sum_{(u,v)\in V_{m,n}} y(u,v)$$

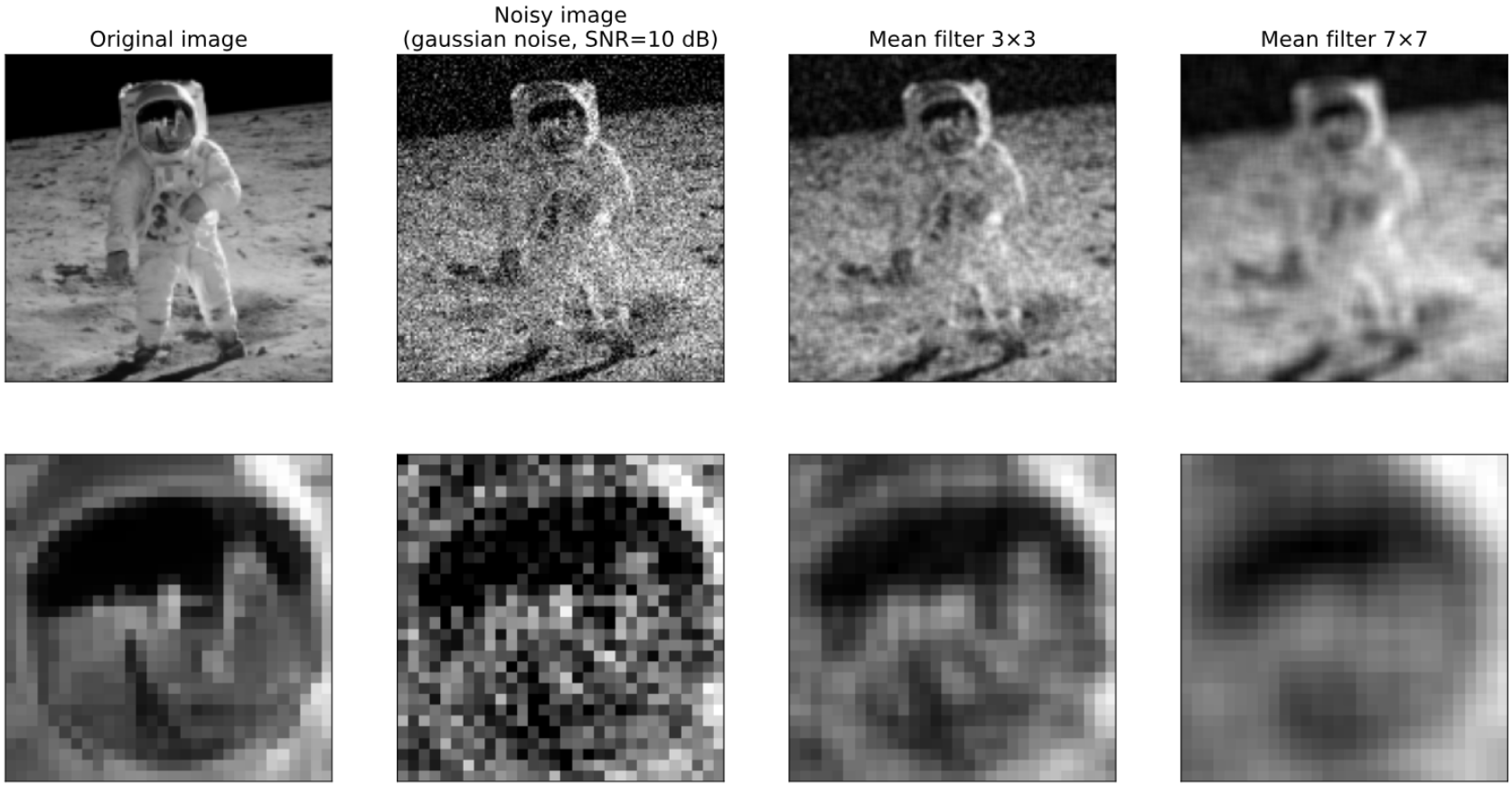
where,

- $V_{m,n}$ is the neighbourhood, that is the set of pixels around (m,n)
- $|V_{m,n}|$ is the cardinality of $V_{m,n}$, that is, the number of pixels in the neighbourhood.

cont...

Following figure illustrates the effect of the mean filter for different sizes of the neighbourhood.

If the neighbourhood grows, then the noise decreases but at the same time the image becomes more blurry.



Effect of the size of the mean filter

cont...

- The mean filter can be expressed with a convolution product.
- Indeed, consider the case where the neighbourhood is a square of size N X N pixels, then the definition of the mean filter gives:

$$\widehat{x}(m,n) = \frac{1}{N^2} \sum_{(u,v)\in V_{m,n}} y(u,v) = \sum_{u,v} g(m-u, n-v) y(u,v)$$

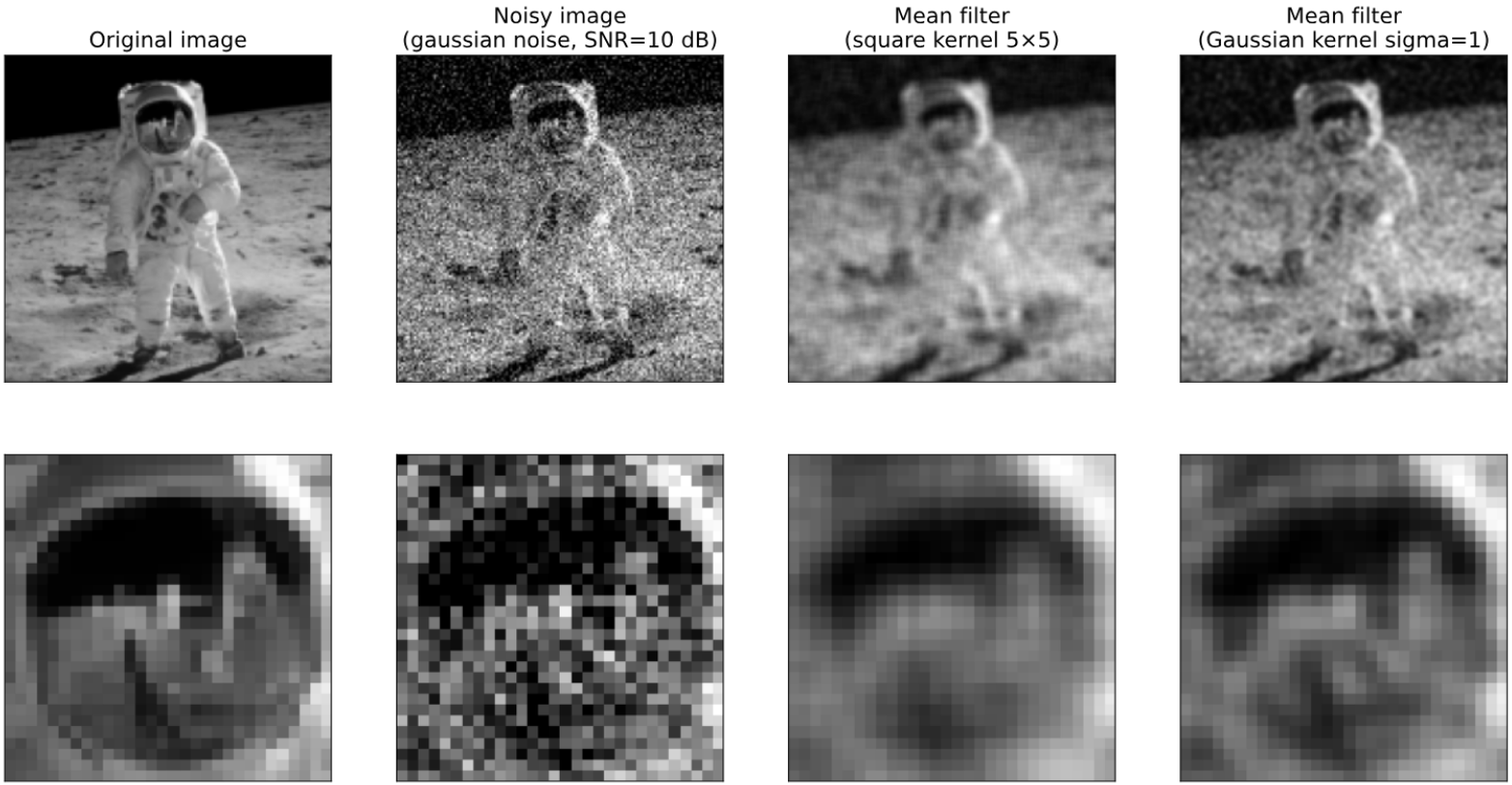
where,

$$g(u,v) = \begin{cases} 1/N^2 & \text{if } u \in \left\{-\frac{N}{2}, \ldots, \frac{N}{2}\right\} \text{ and } v \in \left\{-\frac{N}{2}, \ldots, \frac{N}{2}\right\} \\ 0 & \text{otherwise} \end{cases}$$

This definition can be extended to any type of kernel g.

cont...

- For example, following fig. gives the result for two different kernels



Two mean filters with different kernels

cont...

In summary:

- the mean filter calculates the average of the pixels in a neighbourhood,
- the mean filter can be written as a convolution,
- the noise is reduced by averaging the intensities but the image is blurred.

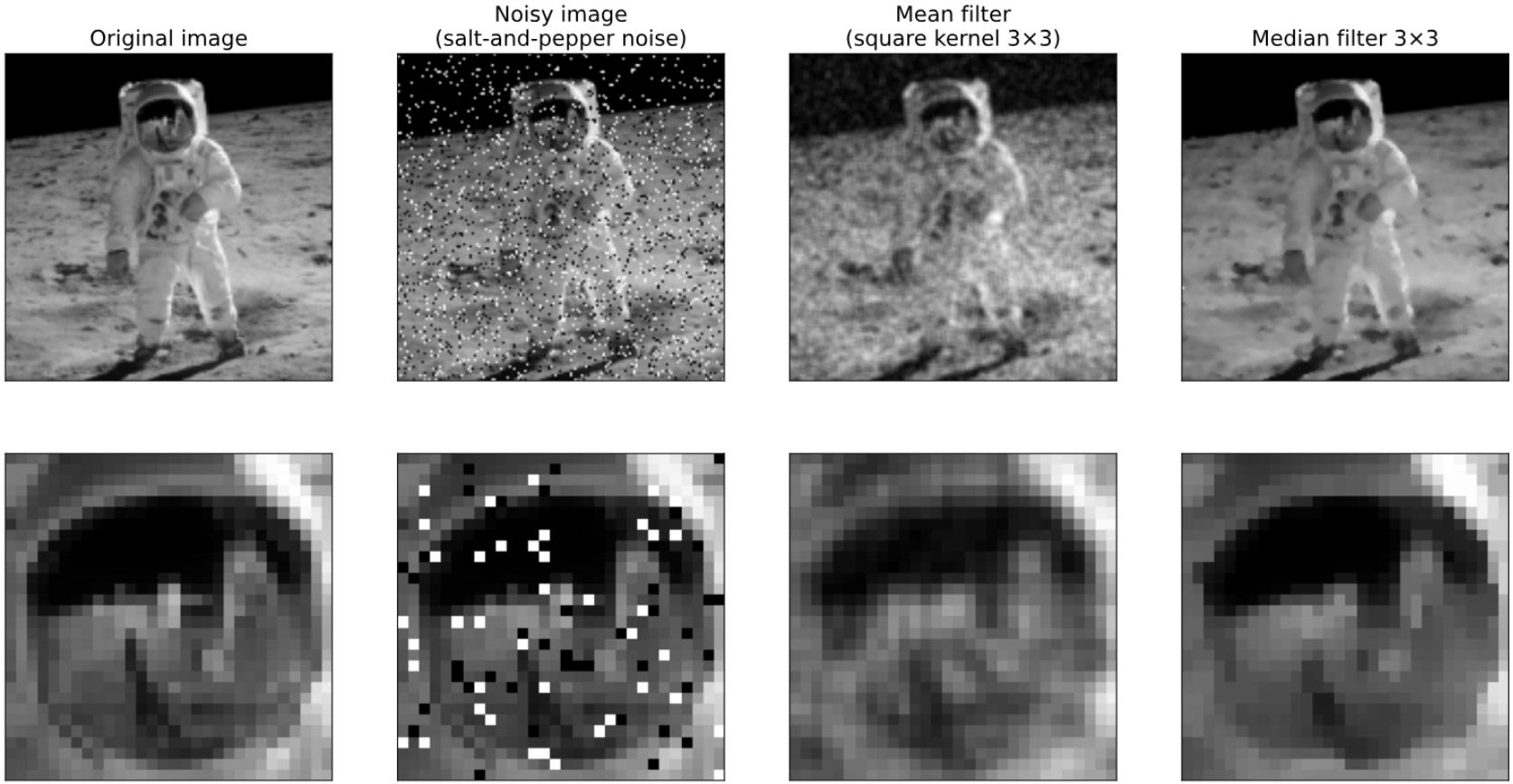
Median filter

- The median of a set of numbers is the element m, of the set such that m there are as many numbers smaller than m as there are numbers larger than m.
- For example, the median of $\{1, 2, 4, 8, 16\}$ is 4.
- The median filter is defined by:

$$\forall m,\ n \quad \widehat{x}(m,n) = \operatorname{median}\big(\{y(u,v) \mid (u,v) \in V_{m,n}\}\big)$$

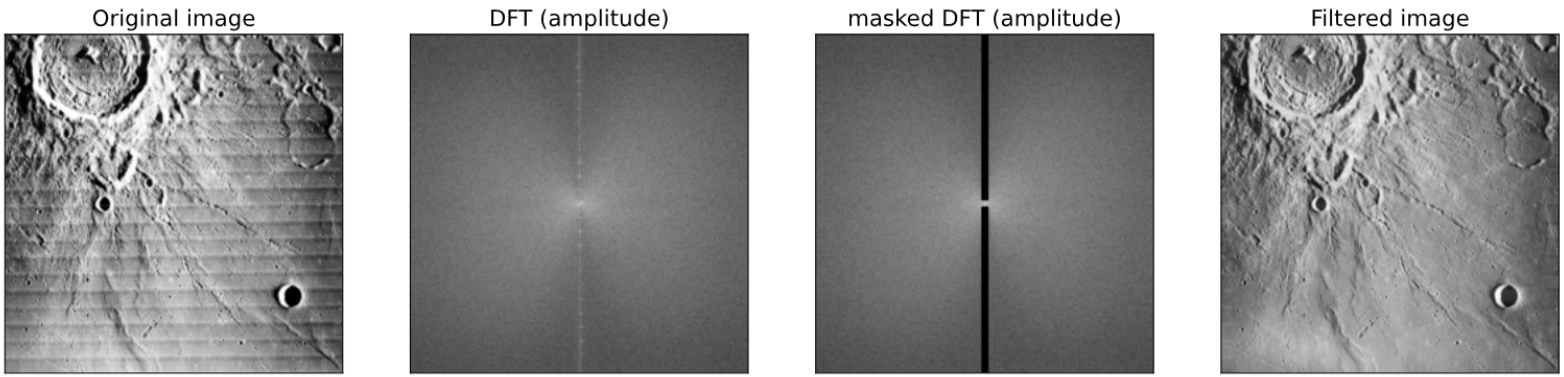
- The median filter is excellent for denoising an image in the case of salt-and-pepper noise because it does not blur the image, as a mean filter would do.

Comparison between a mean filter and a median filter, on an image with salt-and-pepper noise.



Periodic noise filtering

- Periodic noises are characterized by structures in the Fourier transform.
- These structures can be removed in the Fourier domain by cancelling the coefficients using a mask.
- The denoised image is then obtained by an inverse Fourier transform.



Filtering a periodic noise on a photograph of the Moon: the image is cleaned of periodic image artefacts.

TV regularization

From a certain point of view, the goal of denoising is to obtain an image $\widehat{x}$ not only with small variations in intensity between pixels but also close to the observation y.

TV regularization (total variation) [Rudin et al. 1992, Chambolle 2004] is a denoising method that describes these two objectives by mathematical functions (the so-called "criteria").

- The wish to have a denoised image $\widehat{x}$ close to the observation y results in a "data-fit" criterion which measures the difference between x and y.
- A classic choice to measure this difference is the least-squares criterion:

$$E(x, y) = \sum_{m,n} (y(m, n) - x(m, n))^2$$

Indeed, for E to decrease, we must find an image x close to y.

cont...

- The desire to have an image with small variations in intensity results in a "regularization" criterion which measures the difference between the neighbouring pixels of x.
- A simple choice is the "total variation":

$$R(x) = \sum_{m,n} |x(m+1,n) - x(m,n)| + \sum_{m,n} |x(m,n+1) - x(m,n)|$$

- So, for R to decrease, the difference between two consecutive pixels, whether they are in a row or column, must be small. This implies the image x to be nearly constant in intensity.

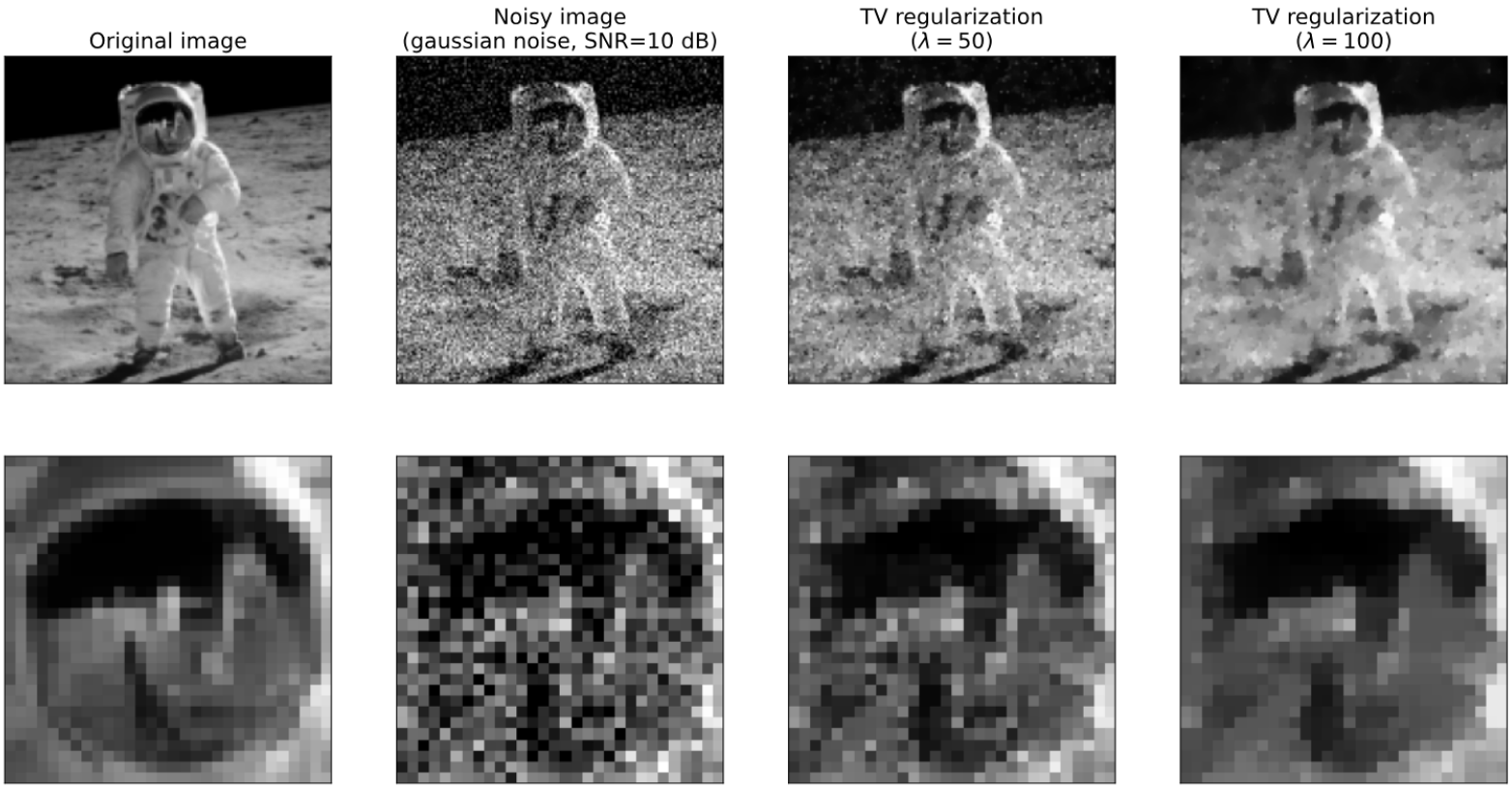
cont...

- The goal is then to find the image x which minimizes both the data-fit and the regularization.
- Mathematically, one look for the image x which minimizes $E(x, y) + \lambda R(x)$, where λ is the "regularization parameter" which is used to adjust the compromise between the two criteria.
- The value of λ is chosen by the user. Mathematically, we write:

$$\widehat{x} = \arg\min_{x} E(x, y) + \lambda R(x)$$

cont...

- This comes to an optimization problem, and there are a large number of algorithms to minimize $E(x, y) + \lambda R(x)$



Denoising with TV regularization, for two values of λ

Deconvolution

- Usually, images acquired by a vision system suffer from degradation that can be modeled as a convolution.
- For example, some images present a camera shake effect or a blur due to poor focus.
- The goal of deconvolution is to cancel the effect of a convolution.

fig 1: Motion Blur

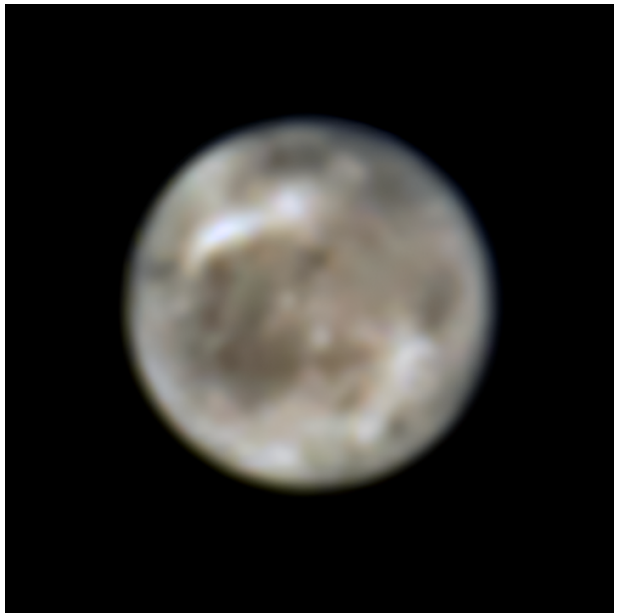


fig 2: Deblurred Image

cont...

- The degradation phenomenon is modeled as in the following figure.

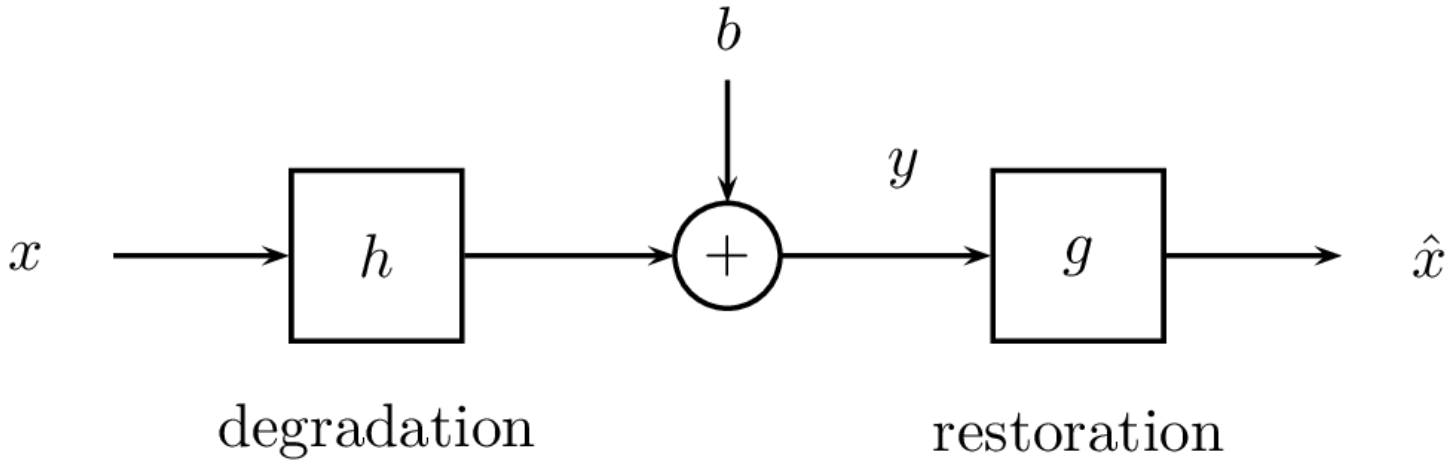


fig 3: Deconvolution model

- The observed image y is degraded by the convolution with a PSF h and, possibly, by a noise b (considered to be additive).

$$y = h * x + b$$

cont...

- The deconvolution computes a deconvolved image $\widehat{x}$ from the observation y.
- We will consider only linear methods, thus deconvolution comes to filtering by g:

$$\widehat{x} = g * y$$

- Deconvolution needs a degradation model, thus having knowledge about both h and b.
  1. The PSF h can be estimated by observation, i.e. by finding in the image some factors to estimate h. For example, a single point object in the image is h.
  2. The PSF can also be estimated by experimentation by reproducing the observation conditions in a laboratory.
  3. So, the image of a pulse gives an estimate of h.
  4. Finally, it is also possible to estimate the PSF from a mathematical model of the physics of the observation.
  5. Note also that some deconvolution methods estimate the PSF h at the same time as x: these are called blind deconvolution methods

Inverse filter

- The inverse filter is the simplest deconvolution method. Since the degradation is modelled $y = h * x + b$, then this equation becomes in the Fourier domain:

$$Y = HX + B$$

$$X = \frac{Y - B}{H}.$$

- We obtain x by calculating the inverse Fourier transform of the previous expression:

$$x = \mathcal{F}^{-1}\left[\frac{Y - B}{H}\right].$$

- As the noise (and therefore its spectrum B) is unknown, we can approximate the expression of x by cancelling B in the previous expression, and thus get the deconvolved image:

$$\widehat{x} = \mathcal{F}^{-1}\left[\frac{Y}{H}\right]$$

cont...

- The result of the inverse filter applied on an image is given in the following fig.
- The result is not usable, and yet the observed image is very little blurred with very little noise

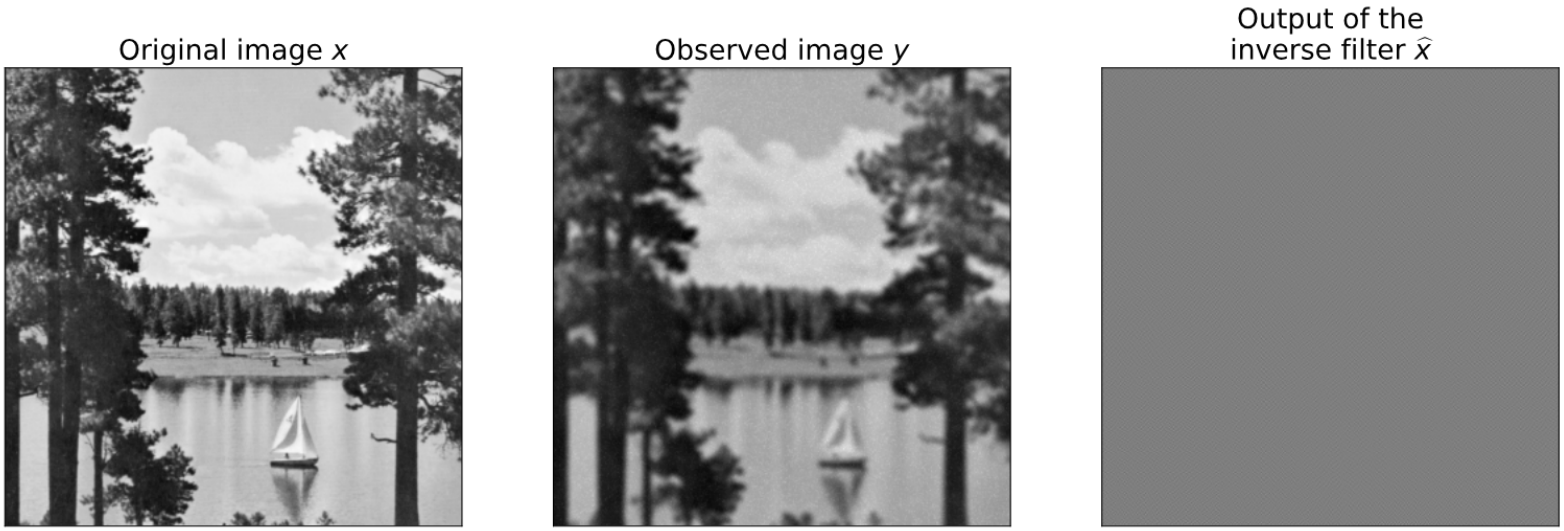


fig 4: Result of the inverse filter

cont...

- The catastrophic result of the inverse filter is due to the fact of having considered the noise to be zero.
- Indeed, according to the definition of $\widehat{x}$ and considering Y = HX + B, then:

$$\widehat{x} = \mathcal{F}^{-1}\left[\frac{Y}{H}\right] = \mathcal{F}^{-1}\left[X + \frac{B}{H}\right] = x + \mathcal{F}^{-1}\left[\frac{B}{H}\right]$$

cont...

- Thus, the deconvolved image $\widehat{x}$ corresponds to x with an additional term which is the inverse Fourier transform of B/H.
- The PSF H is generally a low-pass filter, so the values of $H(m, n)$ tend towards 0 for high frequencies (m, n).
- Because H is in the denominator, this tends to drastically amplify the high frequencies of the noise, and then the term B/H quickly dominates X.
- This explains the result of the above figure.
- One solution consists in considering only the low frequencies of Y/H.
- This is equivalent to truncating the result given by the inverse filter by cancelling the high frequencies before calculating the inverse Fourier transform.
- The result of the deconvolution is much more acceptable, as shown by the following figure, although the result is still far from perfect (there are many variations in intensity around objects, such as tree trunks)...

Output of truncated inverse filter

fig 5: Result of the truncated inverse filter with very small noise

Wiener Filter

- Wiener filter, denoted by g (with Fourier transform G), applies to the observation y such that:

$$\widehat{x} = g * y \qquad \Leftrightarrow \qquad \widehat{X} = GY.$$

- This filter is established in the statistical framework: the image x and the noise b are considered to be random variables.
- They are also assumed to be statistically independent. As a result, the observation y and the estimate $\widehat{x}$ are also random variables.
- The calculations are done in the Fourier domain for simplicity (since convolutions become multiplications).
- The goal of the Wiener filter is to find the image $\widehat{X} = \mathcal{F}[\widehat{x}]$ closest to $X = \mathcal{F}[x]$, in the sense of the mean squared error $\mathrm{MSE} = \mathbb{E}\left[(\widehat{X} - X)^2\right]$.

cont...

Thereby:

$$\begin{aligned}\mathrm{MSE} &= \mathbb{E}\left[(\widehat{X} - X)^2\right] \\ &= \mathbb{E}\left[(GY - X)^2\right] \\ &= \mathbb{E}\left[\big(G(HX + B) - X\big)^2\right] \\ &= \mathbb{E}\left[\big((GH - I)X + GB\big)^2\right]\end{aligned}$$

where I is an image filled with 1. So:

$$\mathrm{MSE} =$$

$$\mathbb{E}\Big[(GH-I)^*(GH-I)X^*X+(GH-I)^*GX^*B+G^*(GH-I)B^*X+G^*GB^*B\Big]$$

cont...

where $\cdot^*$ denotes the conjugate of the variables. Since the expectation E is linear and only X and B are random variables, we can decompose the previous expression into four terms:

$$\begin{aligned}\mathrm{MSE} =& (GH - I)^*(GH - I)\,\mathbb{E}\left[X^*X\right] \\ &+ (GH - I)^* G\,\mathbb{E}\left[X^*B\right] \\ &+ G^*(GH - I)\,\mathbb{E}\left[B^*X\right] \\ &+ G^*G\,\mathbb{E}\left[B^*B\right].\end{aligned}$$

cont...

- Since X and B are independent, then the covariances $\mathbb{E}\left[X^* B\right]$ and $\mathbb{E}\left[B^* X\right]$ are zeros.
- Moreover, $\mathbb{E}\left[X^* X\right]$ and $\mathbb{E}\left[B^* B\right]$ are the power spectral densities denoted as S_x and S_b (the power spectral density is the expectation of the square of the modulus of the Fourier transform).
- So the mean squared error simplifies into:

$$\mathrm{MSE} = (GH - 1)^*(GH - 1)S_x + G^* G S_b$$

cont...

We look for the filter G that minimizes the MSE, or equivalently, that cancels the derivative of MSE:

$$\begin{aligned}
& \frac{\partial \mathrm{MSE}}{\partial G} = (GH-1)^* HS_x + G^* S_b = 0 \\
\Leftrightarrow \quad & G^* H^* HS_x - HS_x + G^* S_b = 0 \\
\Leftrightarrow \quad & G^*(H^* HS_x + S_b) = HS_x \\
\Leftrightarrow \quad & G^* = \frac{HS_x}{H^* HS_x + S_b} \\
\Leftrightarrow \quad & G = \frac{H^* S_x}{H^* HS_x + S_b} \\
\Leftrightarrow \quad & G = \frac{H^* S_x}{|H|^2 S_x + S_b}
\end{aligned}$$

cont...

- Here we are, we get the expression of the Wiener filter G
- Finally, the deconvolved image is the inverse Fourier transform of GY:

$$\widehat{x} = \mathcal{F}^{-1}\left[\frac{H^* S_x}{|H|^2 S_x + S_b} Y\right]$$

and the term S_b/S_x is replaced by a constant K, which becomes the parameter of the method, to be set by the user.

- Two remarks:
  1. where H vanishes (typically in high frequencies), the problem of noise increase is no longer observed as with the inverse filter, since the inverse filter tends towards 0,
  2. moreover, if the noise in the image is zero, then $S_b = 0$ and Wiener filter comes back to the inverse filter:

$$\widehat{x} = \mathcal{F}^{-1}\left[\frac{H^*}{|H|^2} Y\right] = \mathcal{F}^{-1}\left[\frac{Y}{H}\right]$$

The result of Wiener filter

fig 6: Result of Wiener filter

DnCNN2 Network architecture

- The denoising convolutional neural network DnCNN algorithm framework is mainly composed of three different types of layers, which are displayed in three different colors.
- As shown in the figure, assuming the network depth is d,
  1. **Conv+ReLU:** for the first layer,64 filters with the size of 3×3×c are used to generate 64 feature maps. c represents the number of image channels, that is, when c=1, it is a gray image, and when c=3, it is a color image. Relu is used as an activation function to provide non-linearity for the network.
  2. **Conv+BN+ReLU:** for layers 2(d1),64 filters of size 3×3×64 are used, and batch normalization is added between the convolution layer and ReLU.
  3. **Conv:** in the last layer, only one convolution layer is used, and c filters of the size of 3×3×64 are used to output residual images.

cont...

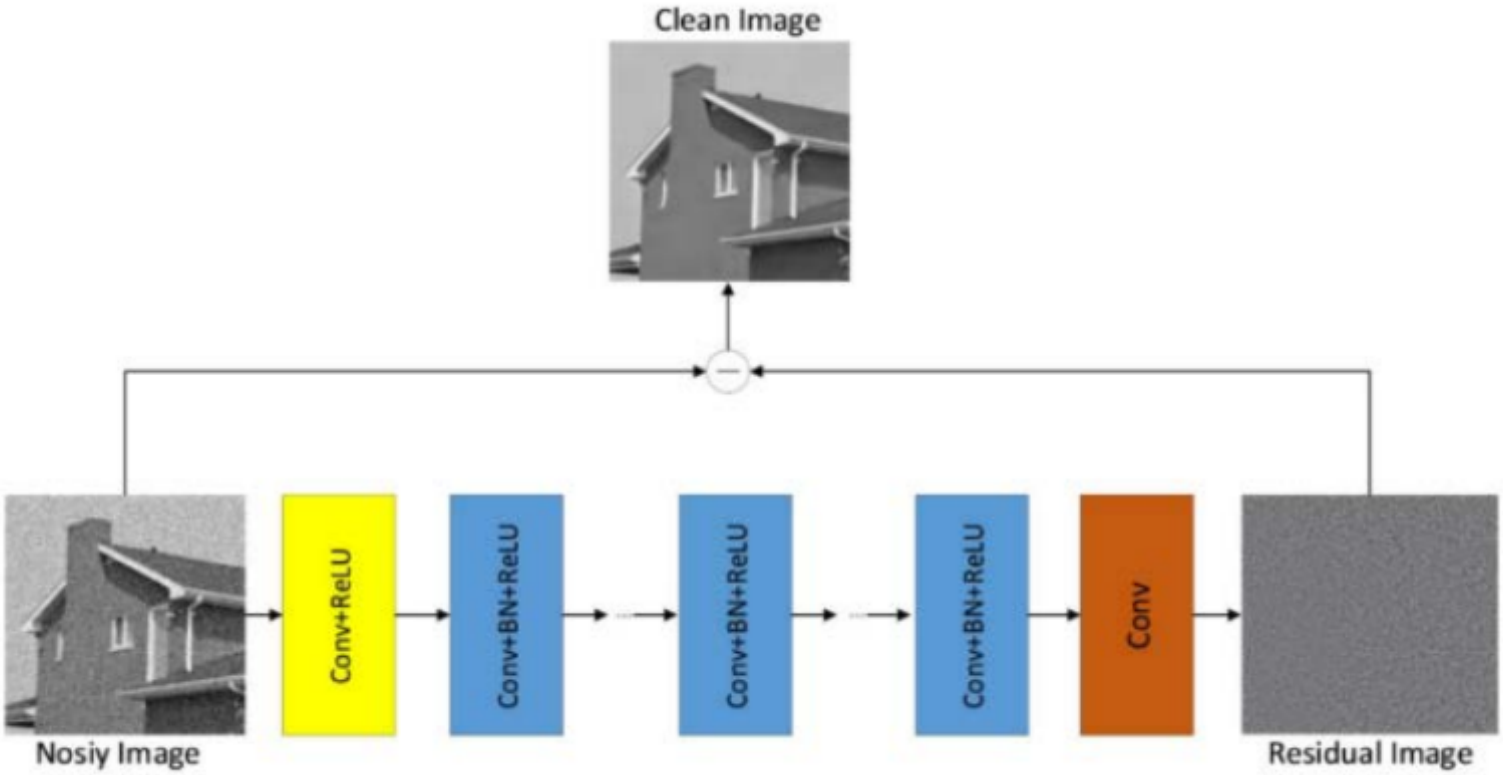


fig 7: Architecture of DnCNN Network

Experimental setup

- In order to verify the effect of noise reduction on Gaussian noise with the known noise level, the training set of this model is BSD68, that is,68 grayscale pictures with 481×321 or 321×481 pixels.
- Gaussian noise images with noise level =15, 25 and 50 were selected to expand the data.
- The designed network depth is 18.L1 loss function and Adam optimizer are used in the training.
- The learning rate of the iterative process was initially set at 104, the minimum batch size is 32.
- The model training lasted 50 times.
- The data set used in the test was BSD68 and SET12.

Qualitative and quantitative assessment (grayscale image)

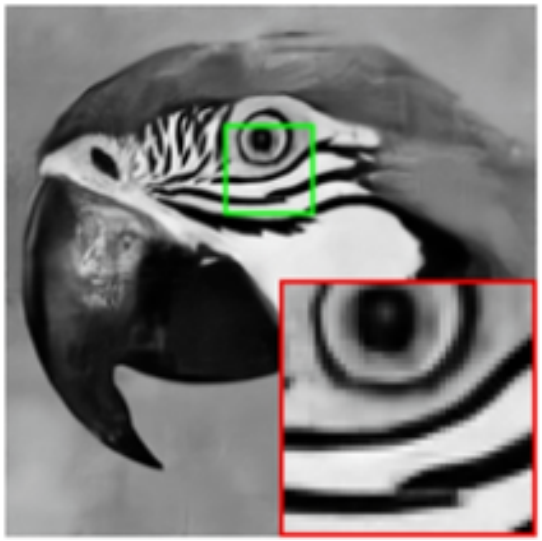


fig 8: Original

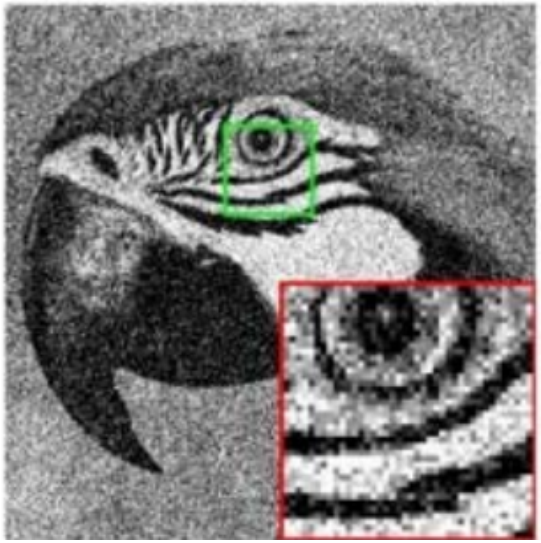


fig 9: Noisy

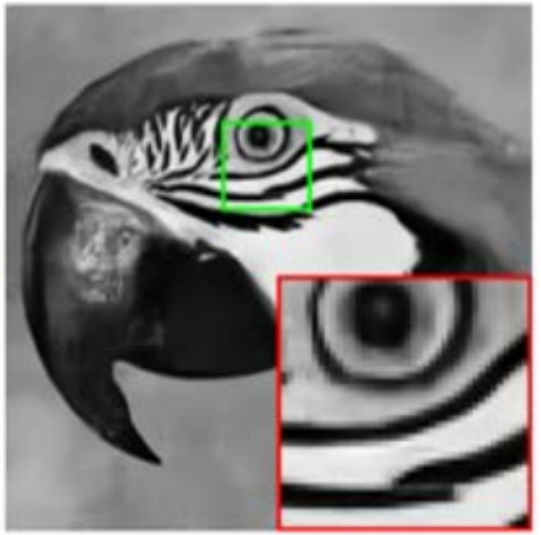
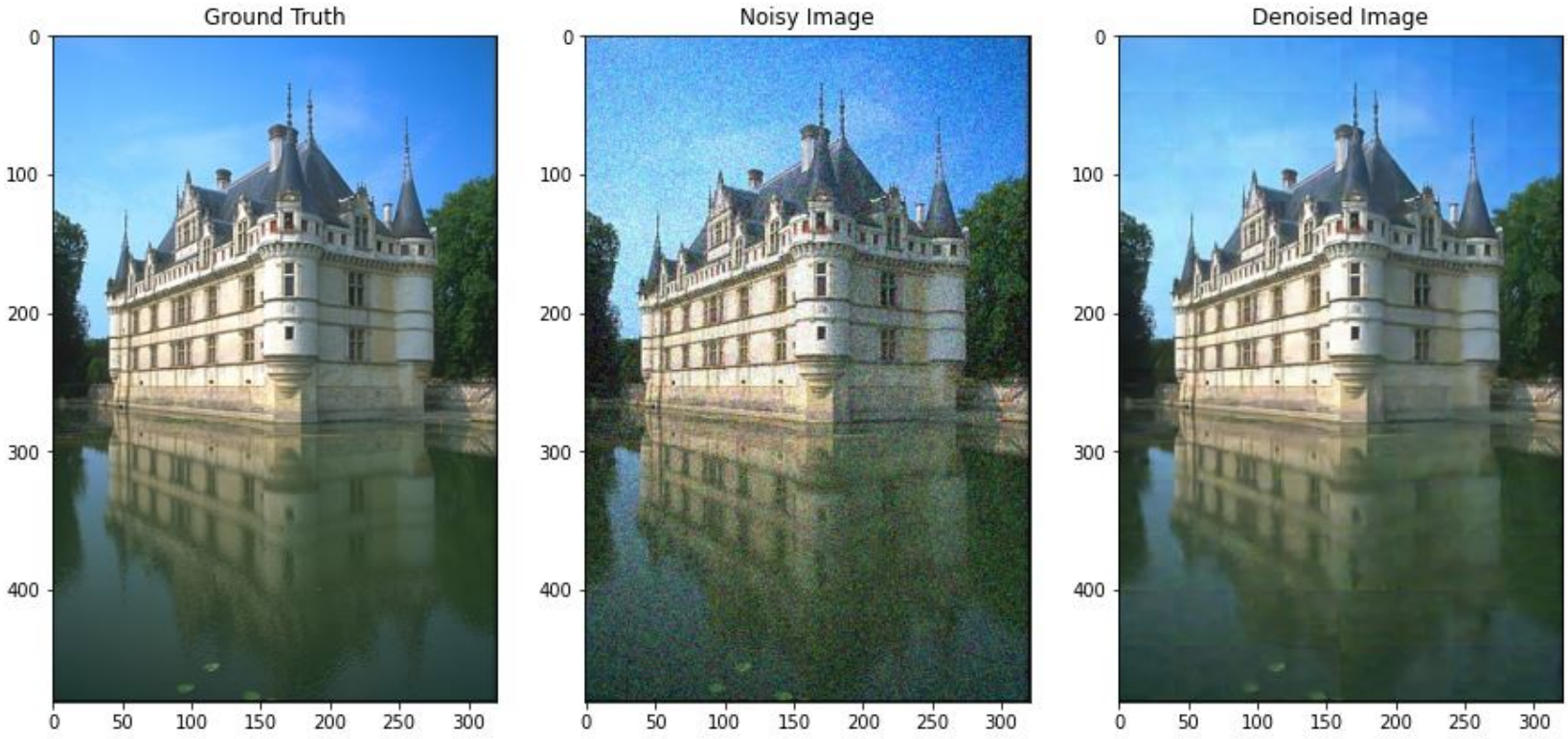


fig 10: Denoised

Original, Noisy, and Denoised images of the parrot

Qualitative and quantitative assessment (colored image)



Original, Noisy, and Denoised images of the castle

Denoising Atuoencoders

- Denoising autoencoders are a type of neural network that we can use to learn a representation (encoding) of data in an unsupervised manner.
- They are trained to reconstruct a clean version of an input signal corrupted by noise.
- This can be useful for tasks such as image denoising or fraud detection, where the goal is to recover the original signal from a noisy version.

cont...

- An autoencoder consists of two parts:
  1. encoder
  2. decoder
- The encoder maps the input data to a lower-dimensional representation or encoding, and the decoder maps the encoding back to the original data space.
- During training, the autoencoder is given a set of clean input examples and the corresponding noisy versions of these examples.
- The goal is to learn a function that maps the noisy input to the clean output using the encoder-decoder architecture.

cont...

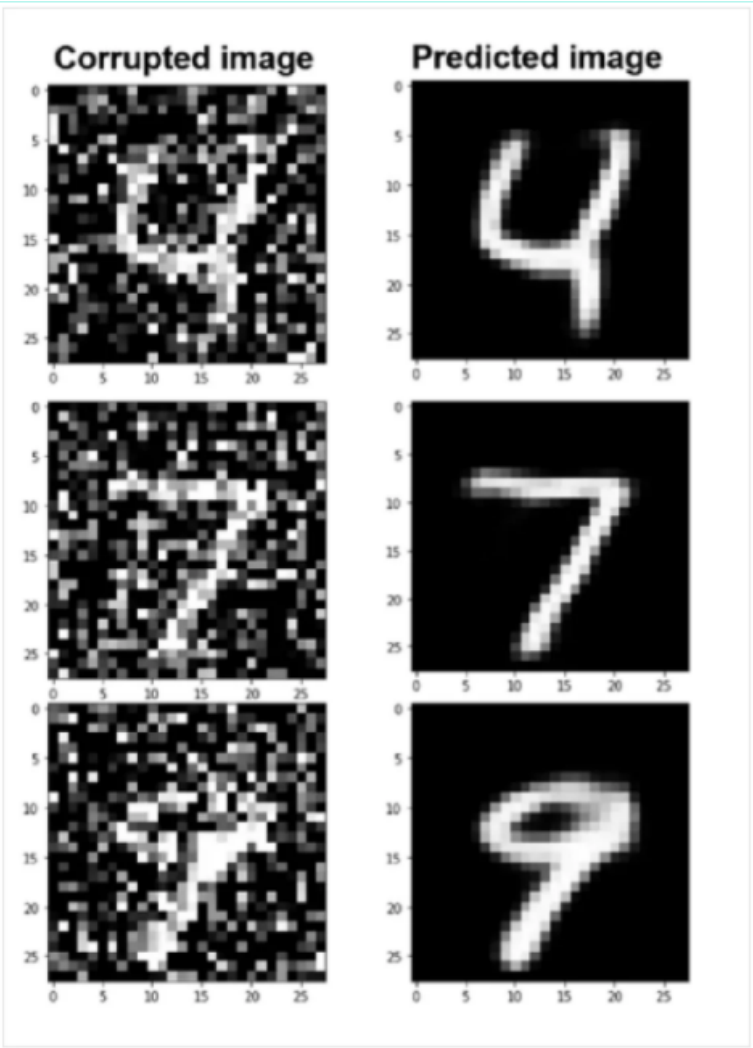


fig 11: Denoised images using autoencoder

Architecture of DAE

- The architecture of a denoising autoencoder (DAE) is similar to that of a standard autoencoder.
- It consists of an encoder, which maps the input data to a lower-dimensional representation, or encoding, and a decoder, which maps the encoding back to the original data space.
- **Encoder:**
  1. Encoder is a neural network with one or more hidden layers.
  2. It takes noisy data as input and outputs an encoding.
  3. Encoding has fewer dimensions than the input data, so the encoder is a compression function.
- **Decoder:**
  1. Decoder is an expansion function that reconstructs original data from compressed encoding.
  2. Input to the decoder is the encoding from an encoder, and output is a reconstruction of the original data.
  3. Decoder is a neural network with one or more hidden layers.

cont...

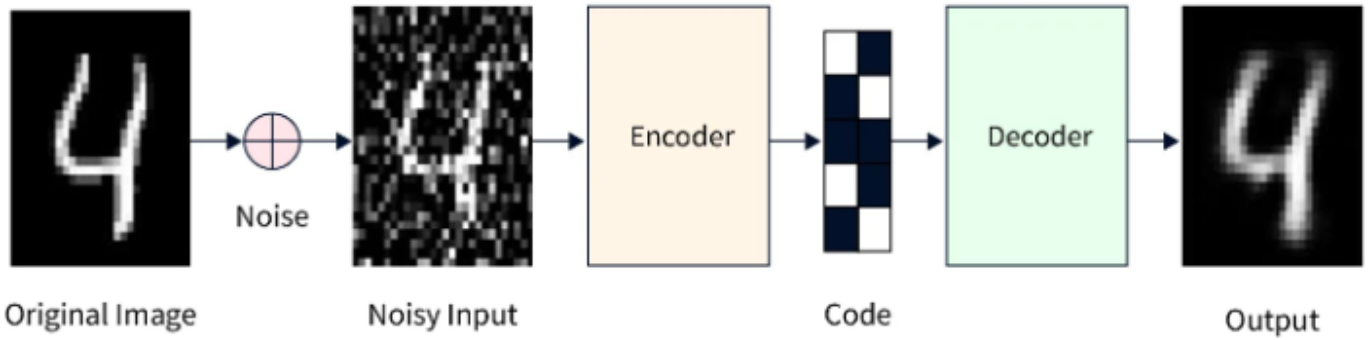


fig 12: Architecture of DAE

DAE training process

- During training, the DAE is given a set of clean input examples and the corresponding noisy versions of these examples.
- The goal is to learn a function that maps the noisy input to the clean output using the encoder-decoder architecture.
- This is typically done using a reconstruction loss function that measures the difference between the clean input and the reconstructed output.
- The DAE is trained to minimize this loss by updating the weights of the encoder and decoder through backpropagation.

Example denoising autoencoder

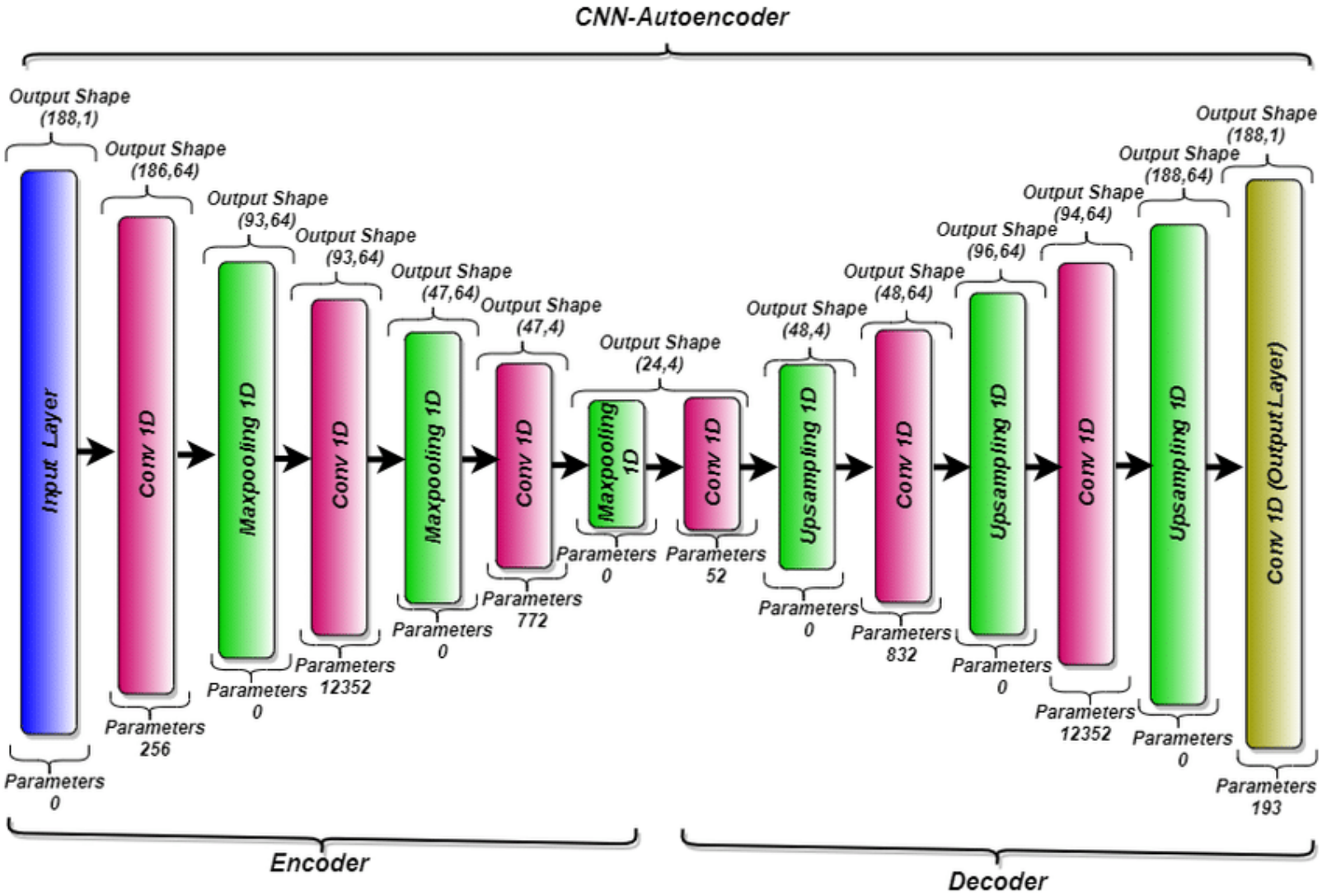


fig 13: CNN Autoencoder