Digital Image Filtering

C. Georgiadis, Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr

Version 4.1

Artificial Intelligence & Information Analysis Lab

Digital Image Filtering

- **Image noise**
- 2D FIR filters
- Moving average filters
- Spatial filters
- Median filters
- Digital filters based on order statistics
- Adaptive order statistic filters
- Anisotropic Diffusion
- Image interpolation
- Neural image filtering

Image noise

- ***White additive noise***:

$$x(i,j) = s(i,j) + n(i,j),$$

- ***White multiplicative noise***:

$$x(i,j) = s(i,j)n(i,j),$$

- ***White signal-dependent noise***:

$$x(i,j) = s^{\gamma}(i,j)n(i,j),$$

- Noise can have various distributions: Gaussian, uniform, Laplacian.

Image noise

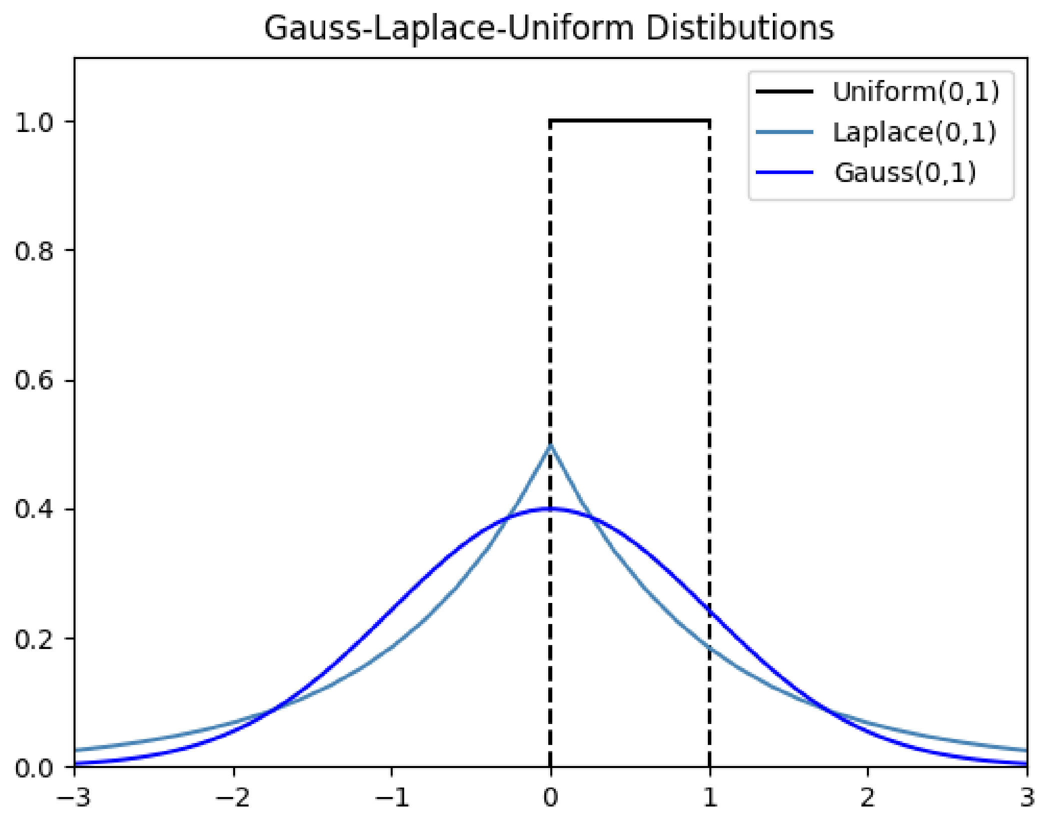
- ***Salt-pepper noise*** consists of black and/or white image impulses:

$$g(i,j) = \begin{cases} z(i,j), & \text{with probability} \quad p. \\ f(i,j), & \text{with probability} \quad 1-p. \end{cases}$$

Image noise

- Uniform noise has a ***short-tailed*** probability distribution.
- Laplacian noise has a ***long-tailed*** probability distribution.

- Gaussian noise is at the borderline between long- and short tailed probability distributions.



from [PIT2000].

Digital Image Filtering

- Image noise
- 2D FIR filters
- Moving average filters
- Spatial filters
- Median filter algorithms
- Digital filters based on order statistics
- Adaptive order statistic filters
- Anisotropic Diffusion
- Image interpolation
- Neural image filtering

Artificial Intelligence & Information Analysis Lab

Digital Image Filtering

- Image noise
- **2D FIR filters**
- Moving average filters
- Spatial filters
- Median filter algorithms
- Digital filters based on order statistics
- Adaptive order statistic filters
- Anisotropic Diffusion
- Image interpolation
- Neural image filtering

Artificial Intelligence & Information Analysis Lab

2D FIR Digital Filters

The output of a 2D FIR filter is given by a ***linear convolution***:

$$y(n_1, n_2) = \sum_{k_1=0}^{M_1-1} \sum_{k_2=0}^{M_2-1} h(k_1, k_2) x(n_1 - k_1, n_2 - k_2).$$

for a ***filter window*** (region of support) $[0, M_1 - 1] \times [0, M_2 - 1]$.

- For centered filter window $[-v_1, v_1] \times [-v_2, v_2]$, $M_i = 2v_i + 1$, $i = 1, 2$:

$$y(n_1, n_2) = \sum_{k_1=-v_1}^{v_1} \sum_{k_2=-v_2}^{v_2} h(k_1, k_2) x(n_1 - k_1, n_2 - k_2).$$

Artificial Intelligence & Information Analysis Lab

Digital Image Filtering

- Image noise
- 2D FIR filters
- **Moving average filters**
- Spatial filters
- Median filters
- Digital filters based on order statistics
- Adaptive order statistic filters
- Anisotropic Diffusion
- Image interpolation
- Neural image filtering

2D FIR Digital Filters

Moving Average filter:

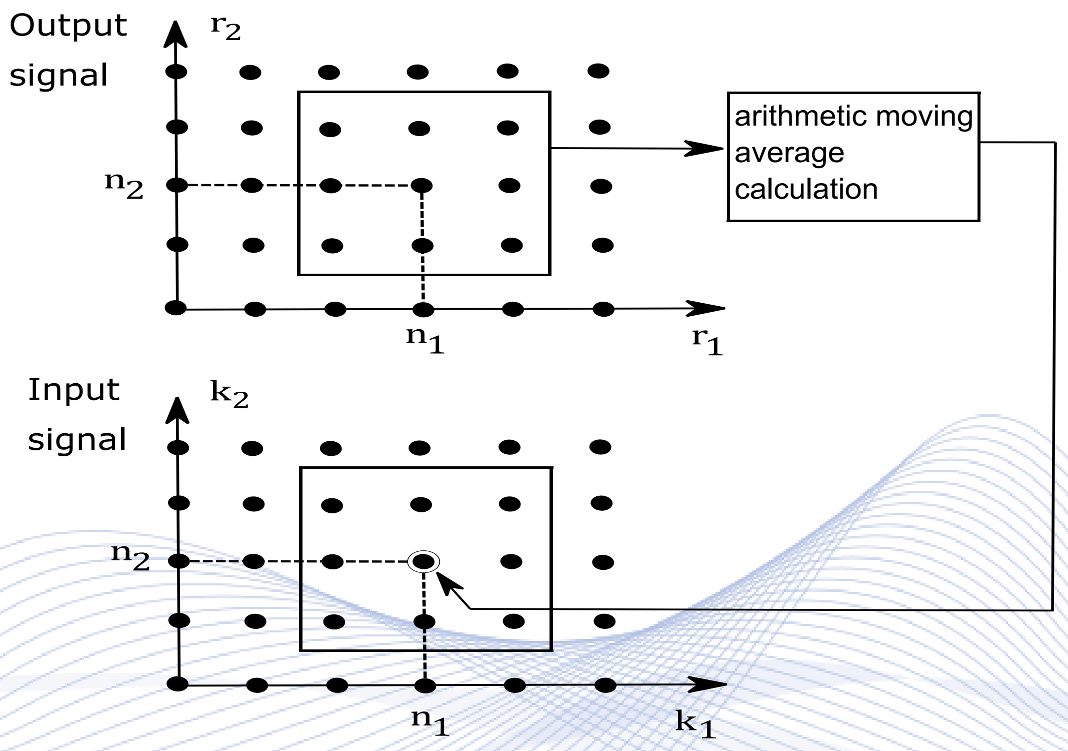
$$y(n_1, n_2) = \left(\frac{1}{M_1 M_2}\right) \sum_{k_1=-\nu_1}^{\nu_1} \sum_{k_2=-\nu_2}^{\nu_2} x(n_1 - k_1, n_2 - k_2),$$

where $M_i = 2v_i + 1,\ i = 1, 2$.

Properties:

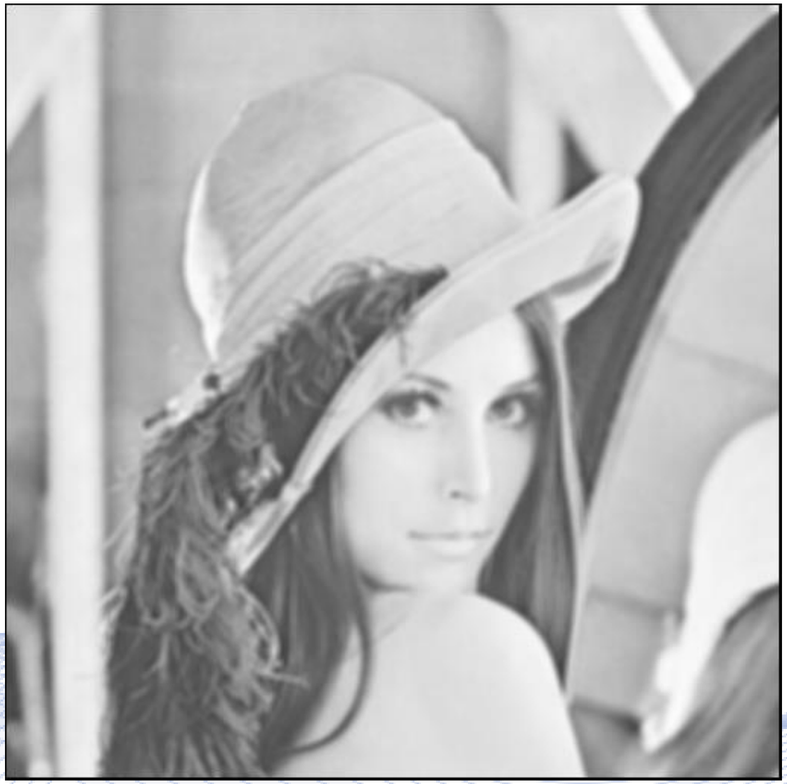
- It is a linear FIR ***low-pass filter***.
- It tends to blur edges and image details (e.g., lines, fine texture).
- It degrades image quality, particularly for large filter windows.

Moving Average Filter



3×3 arithmetic moving average filter structure.

Moving Average Filter



5 × 5 moving average image filtering [PIT2000].

2D FIR Digital Filters

Moving average filter properties:

- It is optimal in removing additive white Gaussian noise:

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\{-\frac{(x-\mu)^2}{2\sigma^2}\}.$$

- Arithmetic mean $\bar{x}$ is the optimal estimator of location μ, as it minimizes the L_2 norm:

$$\sum_{i=1}^{n} (x_i - \bar{x})^2 \rightarrow \text{min}.$$

L_p Mean Filter

***L_p* mean filter**:

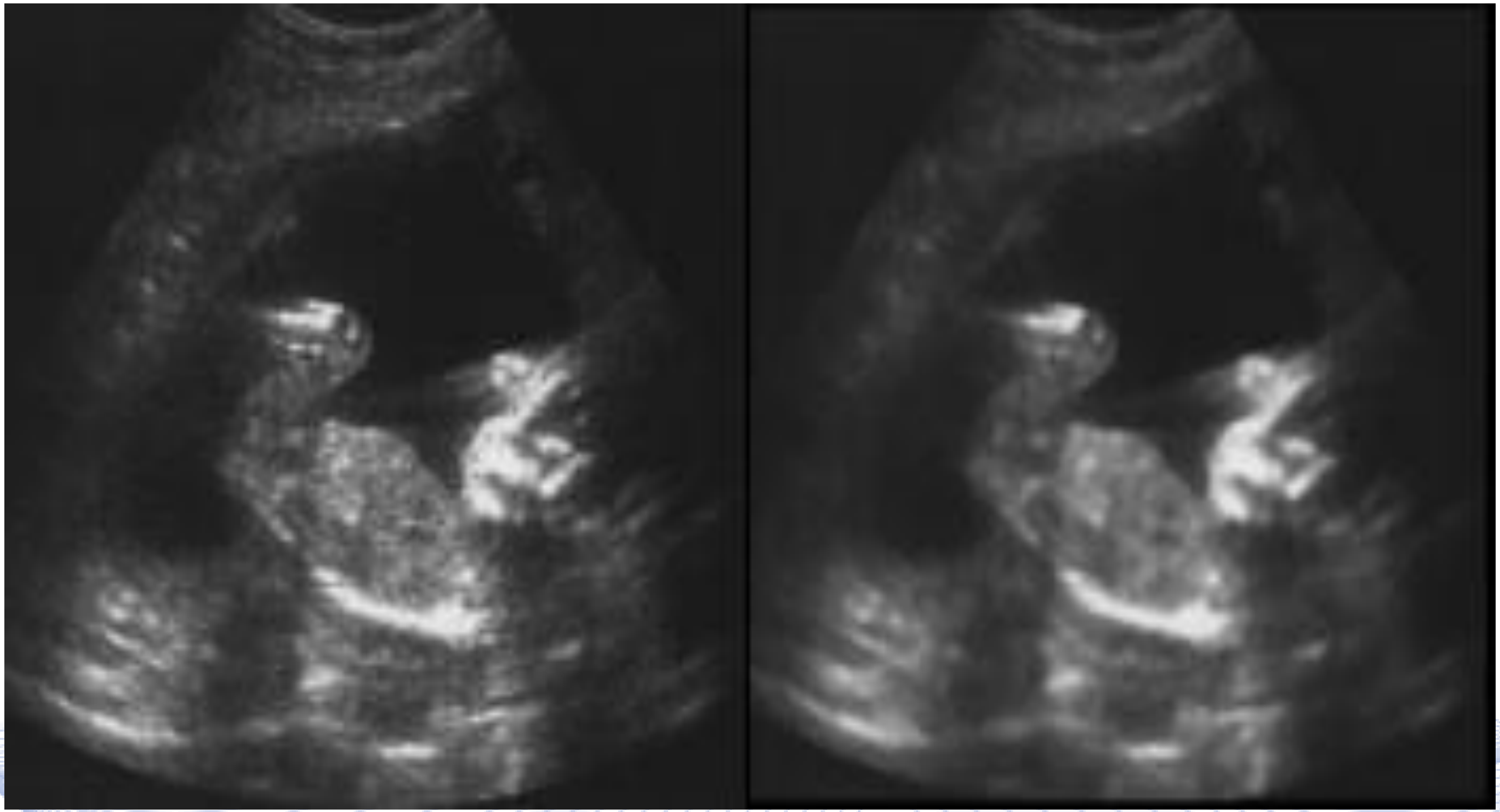
$$y(n_1, n_2) = \frac{1}{M_1 M_2}\Big(\sum_{k_1=-\nu_1}^{\nu_1}\sum_{k_2=-\nu_2}^{\nu_2} x^p(n_1-k_1, n_2-k_2)\Big)^{1/p},$$

where $M_i = 2v_i + 1,\ i = 1, 2$.

Properties:

- For large values, it tends to the maximum filter.
- ***L_2* mean filter** is optimal in removing ***Rayleigh noise*** (e.g., for ultrasound images).

L_p Mean Filter


a) Ultrasound image; b) Output of an L_2 filter [PIT2000].

Digital Image Filtering

- Image noise
- 2D FIR filters
- Moving average filters
- **Spatial filters**
- Median filters
- Digital filters based on order statistics
- Adaptive order statistic filters
- Anisotropic Diffusion
- Image interpolation
- Neural image filtering

Spatial Filters

Gaussian smoothing is performed by the 2D filter kernel:

$$g(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

- This kernel has zero mean.
- σ: ***standard deviation*** of the ***Gaussian kernel***.
- The Gaussian kernel has low-pass frequency characterics:

$$G\left(\omega_x,\omega_y\right)=e^{-2\pi^2({\omega_x}^2+{\omega_y}^2)\sigma^2}.$$

- It can be used to blur images and remove detail and noise.
- The degree of smoothing is determined by σ.

Spatial Filters

$$\frac{1}{273}\begin{array}{|c|c|c|c|c|}\hline 1 & 4 & 7 & 4 & 1 \\ \hline 4 & 16 & 26 & 16 & 4 \\ \hline 7 & 26 & 41 & 26 & 7 \\ \hline 4 & 16 & 26 & 16 & 4 \\ \hline 1 & 4 & 7 & 4 & 1 \\ \hline\end{array}$$

5×5 discrete approximation of a Gaussian kernel for $\sigma = 1$.

Spatial Filters

Unsharp Filter enhances image edges and other high frequency image features, by:

- subtracting a smoothed version of the image from the original to create an edge image.
- Adding the amplified edge image on the original image.

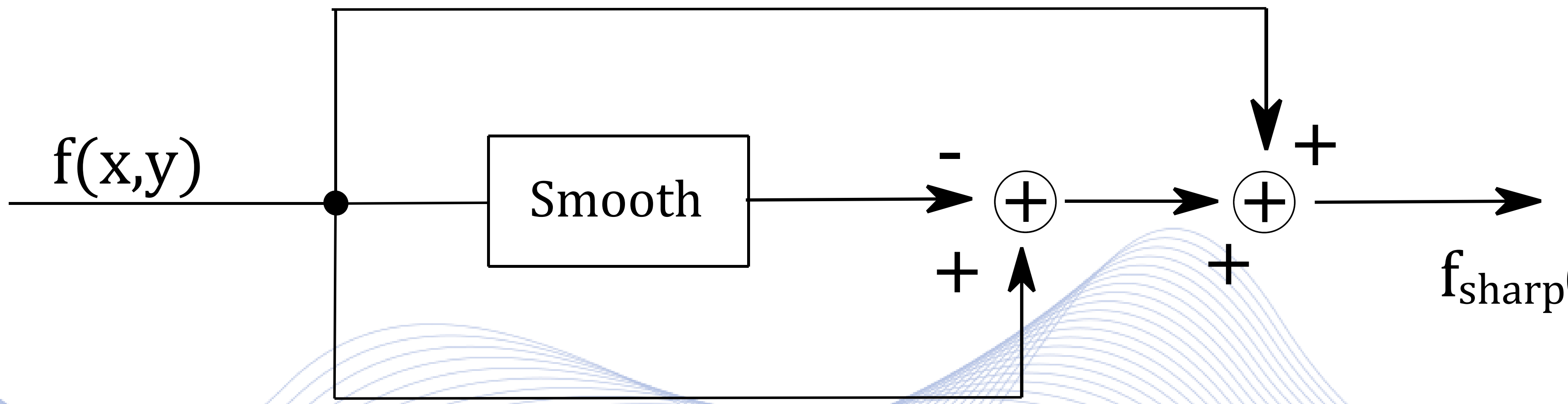
$$f_u(n_1, n_2) = f(n_1, n_2) + kg(n_1, n_2).$$
$$g(n_1, n_2) = f(n_1, n_2) - f_s(n_1, n_2),$$

- $f(n_1, n_2)$: original image.
- $f_s(n_1, n_2)$: smoothed version of $f(n_1, n_2)$.
- $g(n_1, n_2)$: edge image.
- $f_u(n_1, n_2)$: output image.
- k: scaling constant between 0.2 and 0.7.

Artificial Intelligence & Information Analysis Lab

Spatial Filters

Unsharp Filter



Block diagram of the unsharp filter.

Spatial Filters

Conservative smoothing assumes that noise has a high spatial frequency.

- It can be attenuated by a local operation which ensures pixel intensity consistency in local image neighborhoods.
- It ensures that pixel intensities are bounded within the intensity ***range*** of its neighbors, defined by their ***minimum*** and ***maximum*** intensity values.
- If the central pixel intensity lies within the intensity range of its neighbors, it remains unchanged.
- If it is greater/smaller than the maximum/minmum value, it is set equal to the maximum/minimum value, respectively.

Spatial Filters

Conservative smoothing

- The central pixel intensity is 150, so it will be replaced with the maximum intensity value (127) of its 8 nearest neighbors.

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

Conservative smoothing in a local pixel neighborhood.

Digital Image Filtering

- Image noise
- 2D FIR filters
- Moving average filters
- Spatial filters
- **Median filters**
- Digital filters based on order statistics
- Adaptive order statistic filters
- Anisotropic Diffusion
- Image interpolation
- Neural image filtering

Median Filters

Median is the middle sample $x_{(\nu+1)}$ of the ordered sample set $x_i, i = 1, \dots, n, n = 2\nu + 1$:

$$x_{(1)} < x_{(2)} < \cdots < x_{(n)},$$

- $x_{(1)}$: ***minimum***, $x_{(n)}$ ***maximum*** data samples.

- Median is a special type of ***order statistics***.
- It minimizes the L_1 norm:

$$\sum_{i=1}^{n} |x_i - \mathrm{med}| \to \min.$$

Artificial Intelligence & Information Analysis Lab

Median Filters

2D median filter:

$$y(i,j)=\text{med}\{x(i+r,j+s),(r,s)\in \mathrm{A},\ (i,j)\in \mathrm{Z}^2\}.$$

Median filter properties:

• They have low-pass characteristics and remove additive white noise.

• They are very efficient in the removal of:

  •impulsive noise,

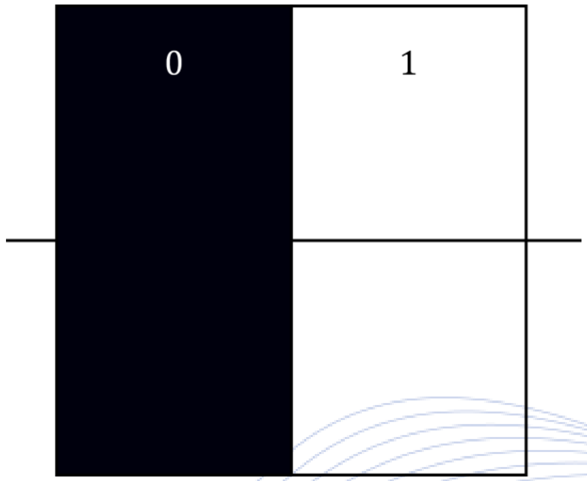
  •noise with long-tailed distribution (e.g., having Laplacian distribution).

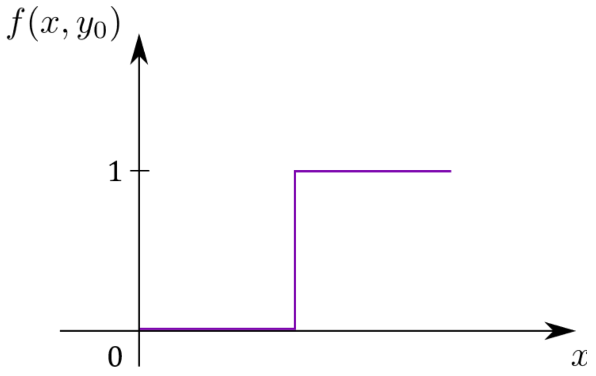
Median Filters

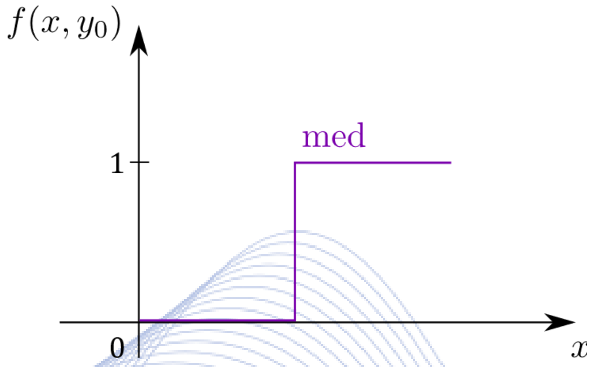
Median filter properties:

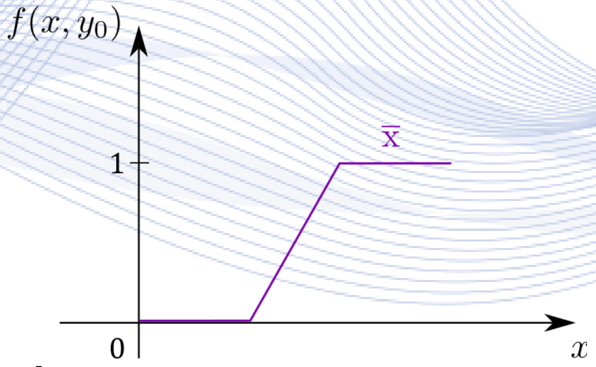
- Median becomes corrupted, if more than 50% of the data samples are outliers.
- Median ***robustness*** renders it very suitable for impulse noise filtering.
- Median filtering preserves and, possibly, enhances image edge sharpness.
- Median filter smooths noise in homogeneous image regions but tends to produce regions of constant or nearly constant intensity (blobs).

Median Filters



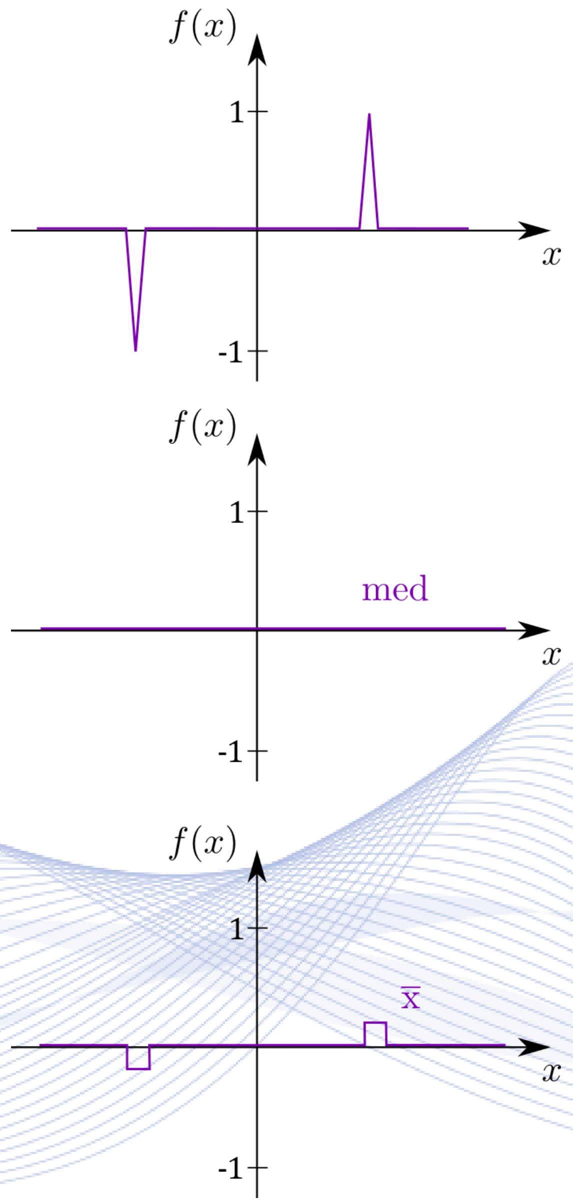






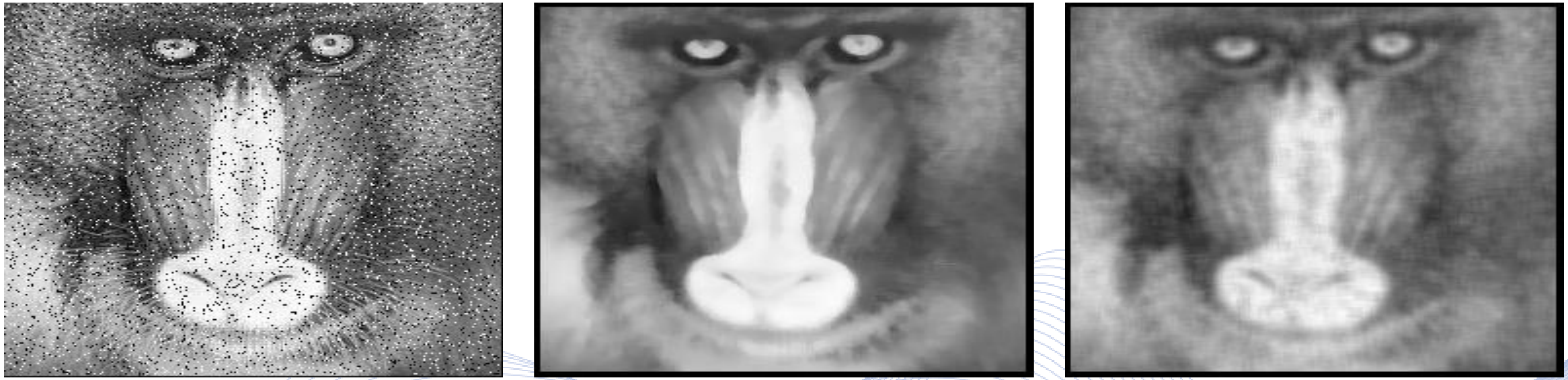
Edge filtering

Median Filters



Impulsive noise filtering

Median Filters



a) Baboon image corrupted by mixed impulsive noise;
b) 7 × 7 median filter output; c) 7 × 7 moving average filter output [PIT2000].

Artificial Intelligence & Information Analysis Lab

Median Filters

Separable 2D median filter:

1D median filtering of length $n=2v+1$ along image rows and columns:

$$y_{i,j}=\text{med}(z_{i,j-v},\ldots,z_{i,j},\ldots,z_{i,j+v}),$$

$$z_{i,j}=\text{med}(x_{i-v,j},\ldots,x_{i,j},\ldots,x_{i+v,j}),$$

- Low computational complexity, compared to non-separable median filter:
  - It sorts n numbers two times, instead of ordering n^2 numbers.

Median Filters

Recursive median filter:

$$y_{i,j}=\mathrm{med}(y_{i-v}, \dots , y_{i-1,}, x_i , \dots , x_{1+v}).$$

- Its output tends to be much more correlated, than that of the standard median filter.
- Recursive median filters have higher immunity to impulsive noise than the non-recursive median filters.

Separable recursive median filter:

$$y_{i,j}=\mathrm{med}(y_{i,j-v}, \dots , y_{i,j-1}, z_{i,j}, \dots ,z_{i,j+v}),$$
$$z_{i,j}=\mathrm{med}(z_{i-v,j}, \dots , z_{i-1,j}, x_{i,j}, \dots , x_{i+v,j}).$$

Artificial Intelligence & Information Analysis Lab

Median Filters

Weighted median is the estimator T that minimizes the weighted L_1 norm:

$$\sum_{i=1}^{n} w_i |x_i - T| \rightarrow \min.$$

It is described by:

$$y_i = \text{med}\{w_{-v} \square x_{i-v}, \ldots, w_v \square x_{i+v}\},$$

where $w \square x$ denotes duplication of x, w times to $\{x, \ldots, x\}$.

Median Filters

Multistage median filter:

$$y_{i,j}=\text{med}(\text{med}(z_1, z_2, x_{i,j}), \text{med}(z_3, z_4, x_{i,j}), x_{i,j}),$$

$$z_1=\text{med}(x_{i,j-v}, \dots, x_{i,j}, \dots, x_{i,j+v}),$$

$$z_2=\text{med}(x_{i-v,j}, \dots, x_{i,j}, \dots, x_{i+v,j}),$$

$$z_3=\text{med}(x_{i+v,j-v}, \dots, x_{i,j}, \dots, x_{i-v,j+v}),$$

$$z_4=\text{med}(x_{i-v,j-v}, \dots, x_{i,j}, \dots, x_{i+v,j+v}).$$

It preserves edges in horizontal, vertical and diagonal directions.

Digital Image Filtering

- Image noise
- 2D FIR filters
- Moving average filters
- Spatial filters
- Median filters
- **Digital filters based on order statistics**
- Adaptive order statistic filters
- Anisotropic Diffusion
- Image interpolation
- Neural image filtering

Artificial Intelligence & Information Analysis Lab

Order Statistics Filters

Ranked order filters:

An $r-\mathrm{th}$ ranked filter y_i output is the $r-\mathrm{th}$ order statistic of signal x_i samples $\{x_{i-v}, \dots, x_i, \dots, x_{i+v}\}$, $n = 2v + 1$ that exist in a ***running filter*** window.

• It introduces a strong bias in the estimation of the mean, when the rank is small or large (tending to ***min*** or ***max filters***).

• The bias is even stronger when the input data have a long-tailed distribution.

Order Statistics Filters

Max/min filters:

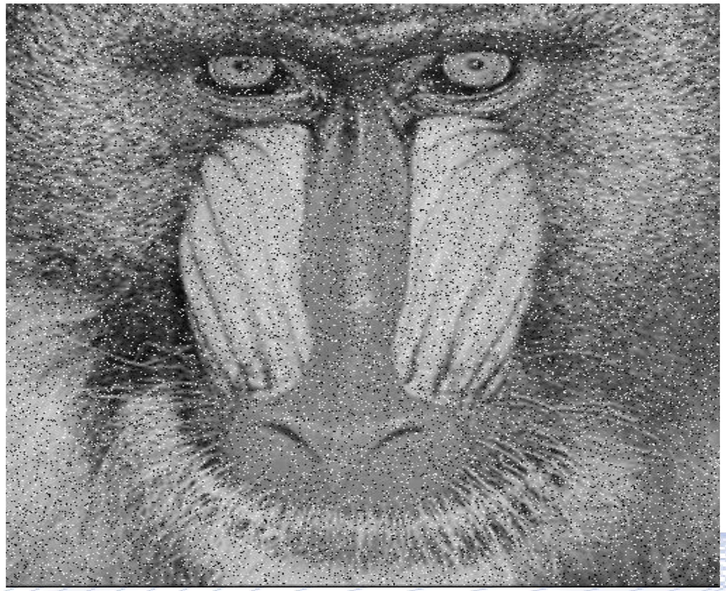
Running maximum $x_{(n)}$ and ***minimum*** $x_{(1)}$ are the two extremes of the ranked-order filters.

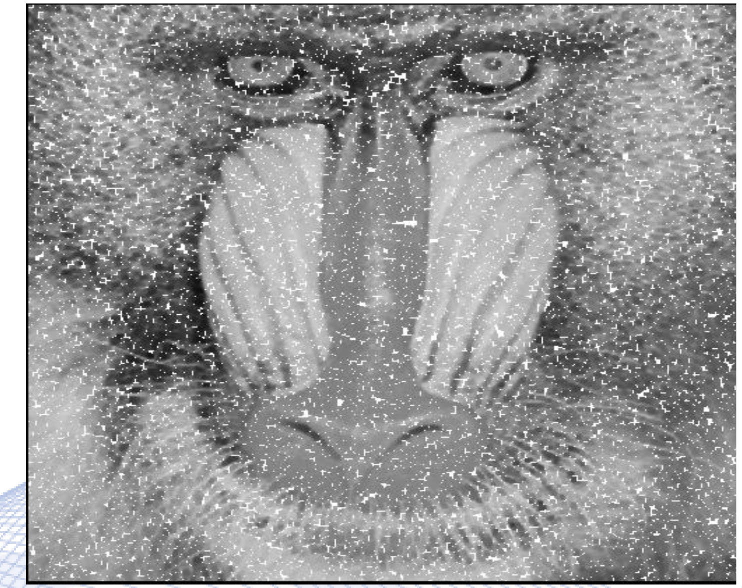
- Maximum filter effectively removes negative impulses in an image.
- Minimum filter removes positive impulses.

- Both filters fail in the removal of mixed impulse noise.
- Both filters have good edge preservation properties (but shift edges).
- Max/min filters tend to enhance bright and dark image regions, respectively.

Order Statistics Filters

Max/min filters





a) Baboon image corrupted by mixed impulsive noise;
b) The output of a cascade of max and min filters [PIT2000].

Order Statistics Filters

Running implementation of max filter:

$$y_i = \begin{cases} x_i, & \text{if } x_i \geq y_{i-1}, \\ y_{i-1}, & \text{if } x_i < y_{i-1} \text{ and } x_{i-n} < y_{i-1}, \\ \max(x_i, \ldots, x_{i-n+1}), & \text{if } x_i < y_{i-1} \text{ and } x_{i-n} = y_{i-1}. \end{cases}$$

- In average, only 3 comparisons are needed.
- A similar algorithm exists for min filtering.

Artificial Intelligence & Information Analysis Lab

Order Statistics Filters

α-trimmed mean filters:

$$y_i = \frac{1}{n(1-2\alpha)} \sum_{j=\alpha n+1}^{n-\alpha n} x_{(j)} .$$

- It rejects α% of the smaller and α% of the larger observation data.
- It is a compromise between the median filter and the moving average filter for varying α.

- Its performance is poor for short-tailed distributions.

Artificial Intelligence & Information Analysis Lab

Order Statistics Filters

Midpoint filter:

$$MP = \frac{1}{2}\left(x_{(1)} + x_{(n)}\right)$$

is optimal for uniform noise.

Order Statistics Filters

Modified trimmed mean filter (MTM):

$$y_{ij} = \frac{\sum\sum_{A} a_{rs} x_{i+r,j+s}}{\sum\sum_{A} a_{rs}},$$

$$a_{rs} = \begin{cases} 1, & |x_{i+r,j+s} - \mathrm{med}\{x_{ij}\}| \leq q \\ 0, & \text{otherwise.} \end{cases}$$

- MTM trims out pixels deviating strongly from the local median.
- It removes outliers.

Order Statistics Filters

Double window modified trimmed mean (***DW MTM***):

- A variation of MTM, it uses two different sized filter windows to achieve good robustness and edge preservation.

Modified nearest neighbour filter (***MNN***):

$$a_{rs} = \begin{cases} 1, & |x_{i+r,j+s} - x_{ij}| \leq q \\ 0, & \text{otherwise.} \end{cases}$$

- MNN trims out pixels deviating strongly from the central pixel value.
- It has good edge preservation properties.

Order Statistics Filters

L-filter (or L-order statistic) definition:

$$y_i = \sum_{j=1}^{n} a_j x_{(j)}.$$

Location Invariance constraint:

$$\sum_{j=1}^{n} a_j = \mathbf{a}^T \mathbf{e} = 1, \qquad \mathbf{e} = [1, \dots, 1]^T.$$

Artificial Intelligence & Information Analysis Lab

Order Statistics Filters

In the case of additive noise:

$$x_i = s_i + n_i,$$

the coefficient vector $\mathbf{a}$ can be obtained after MSE minimization:

$$MSE = E\{(s_i - y_i)^2\} = E\left\{\left(\sum_{j=1}^{n} a_j x_{(j)} - s_i\right)^2\right\} = \mathbf{a}\,\mathbf{R}^T\mathbf{a},$$

$$\mathbf{a} = \frac{\mathbf{R}^{-1}\mathbf{e}}{\mathbf{e}^T\mathbf{R}^{-1}\mathbf{e}}.$$

- $\mathbf{R}$: $n \times n$ correlation matrix of vector $\mathbf{n} = \left[n_{(1)}, \dots, n_{(n)}\right]^T$.

Order Statistics Filters

- The optimal L-filter:
  - for Gaussian noise is the moving average.
  - for Laplacian noise is the median filter.
  - for uniform noise is the midpoint.

- L-filter has no streaking effects, provided that its coefficients are not similar to those of the median filter.
- It has greater computational complexity than both the median and the moving average filter.

Order Statistics Filters

- Midpoint filters optimal estimators in the case of additive white uniform noise.
- Arithmetic moving average filters are optimal estimators in the case of additive white Gaussian noise $N(0,1)$:

$$f_X(x) = \frac{1}{\sqrt{2\pi}} \exp\{-\frac{x^2}{2}\}.$$

- Median filters are optimal estimators in the case of additive white Laplacian noise:

$$f_X(x) = \frac{1}{2} e^{-|x|}.$$

Artificial Intelligence & Information Analysis Lab

Digital Image Filtering

- Image noise
- 2D FIR filters
- Moving average filters
- Spatial filters
- Median filters
- Digital filters based on order statistics
- **Adaptive order statistic filters**
- Anisotropic Diffusion
- Image interpolation
- Neural image filtering

Adaptive Order Statistic Filters

Minimal Mean Square Error (***MMSE***) ***filter***:

$$\hat{s}_{ij} = \left(1 - \frac{\sigma_n^2}{\sigma_x^2}\right) x_{ij} + \frac{\sigma_n^2}{\sigma_x^2} \widehat{m}_x,$$

$$x_{ij} = s_{ij} + n_{ij}.$$

- It is an ***adaptive filter***:
  - It performs like arithmetic mean in homogeneous image regions.
  - It performs no filtering close to edges.

- It preserves edges, as it does not filter the noise in edge regions.
- Various choices of the local measures of $\widehat{m}_x$, σ_x^2, σ_n^2.

Adaptive Order Statistic Filters

Decision-directed filters:

• They take into account both edge and noise information.

• Impulses, when detected, can be removed from the estimation of the local mean, median and standard deviation.

• When an edge is detected, the window of the filter can become smaller, so that edge blurring is minimized.

• Adaptive window edge detection (AWED) filter:
  • AWED filter window size/shape can be adapted.

Adaptive Order Statistic Filters

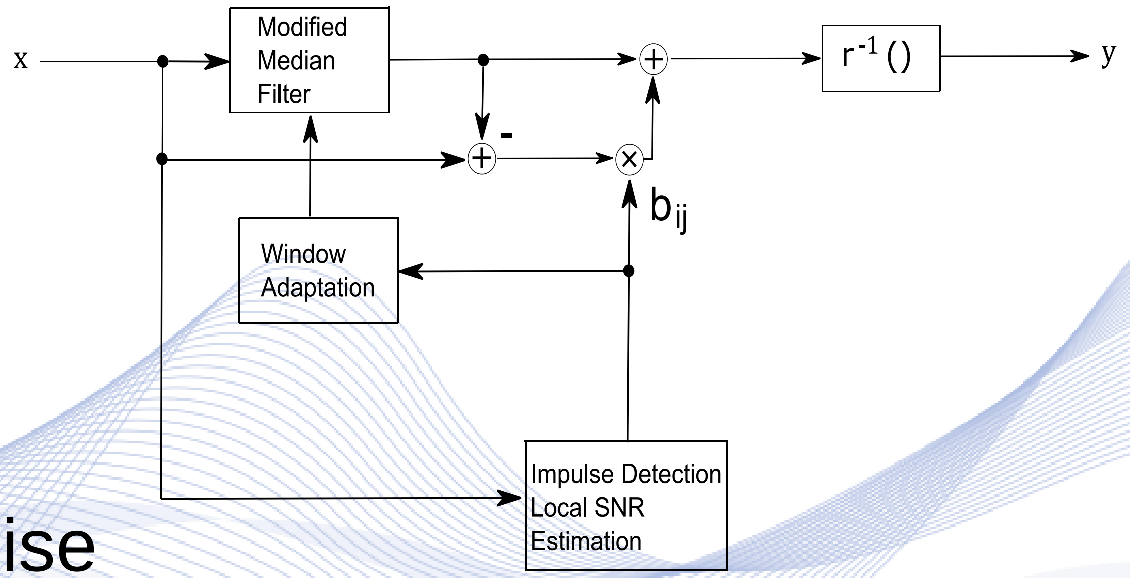
Signal-adaptive median (SAM) filter:

- It is an adaptive filter based on the two-component image model:

$$y_{1ij} = \widehat{m}_x + b_{ij}(x_{ij} - \widehat{m}_x).$$

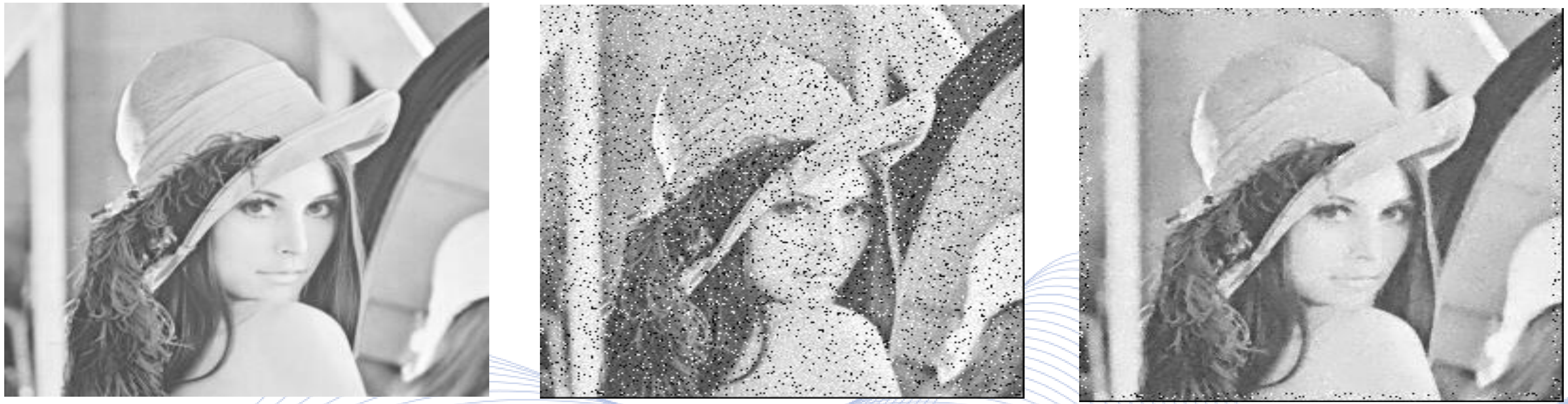
$$y_{ij} = r^{-1}(y_{1ij}).$$

- It has excellent performance in noise filtering, edge and image detail preservation.



Adaptive Order Statistic Filters

Two-component model filters



a) Original image;
b) Image corrupted by Gaussian noise (variance=100) and mixed impulsive noise; c) SAM filter output [PIT2000].

Artificial Intelligence & Information Analysis Lab

Digital Image Filtering

- Image noise
- 2D FIR filters
- Moving average filters
- Spatial filters
- Median filters
- Digital filters based on order statistics
- Adaptive order statistic filters
- **Anisotropic Diffusion**
- Image interpolation
- Neural image filtering

Anisotropic Diffusion

Image intensity $f(i,j)$ can be considered as ***pixel temperature*** that can be diffused over the entire image domain, in an iterative process described by $f(i,j,t)$ over various steps t.

Isotropic diffusion filtering can perform image smoothing:

$$\frac{\partial f}{\partial t} = c\mathrm{div}(\nabla f) = c\left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}\right).$$

- c: diffusion coefficient.
- Diffusion is also used for image segmentation.

Anisotropic Diffusion

Limitations:

- While it smooths noise, isotropic diffusion filtering also blurs important image features, such as edges.
- As iteration number increases, the image will tend to a constant mean average image, hence destroying all image information.
- It dislocates edges, when moving from finer to coarser scales (correspondence problem).

- Some smoothing properties of linear diffusion filtering are only suitable for 1D filtering.

Anisotropic Diffusion

Anisotropic diffusion depends on local image properties, e.g., local image edges.

- It reduces diffusion at image edges:

$$\frac{\partial f}{\partial t} = \mathrm{div}((c(f)\nabla f).$$

- div: divergence operator.
- ∇f: image $f(i,j,t)$ differentiation (edge detection) at iteration t.

- Diffusion close to edges is reduced, because of the form of $c(f)$:

$$c(f) = \frac{1}{1+\frac{|\nabla f|^2}{\lambda^2}}, \qquad \lambda > 0.$$

Artificial Intelligence & Information Analysis Lab

Anisotropic Diffusion

Anisotropic image diffusion equation:

$$\frac{df(i,j,t)}{dt} = div(c(i,j,t)\nabla f) = c(i,j,t)\Delta f + \nabla c \nabla f.$$

- Δ: Laplacian operator.

It performs simultaneous noise reduction and contrast enhancement across image regions, while deriving consistent deterministic scale-space image descriptions.

- It smooths homogeneous image regions while retaining image edges.

Anisotropic Diffusion

The 4-nearest North, South, East and West neighbors of the Laplacian operator can be used:

$$f(i,j,t+1) = f(i,j,t) + \lambda[c_N \nabla_N f + c_s \nabla_S f + c_E \nabla_E f + c_W \nabla_W f](i,j,t).$$

- $0 \leq \lambda \leq \frac{1}{4}$: ensures numerical stability.
- $\nabla_N, \nabla_S, \nabla_E, \nabla_W$ are nearest-neighbor differences:

$$\nabla_N f(i,j,t) \triangleq f(i-1,j,t) - f(i,j,t),$$
$$\nabla_S f(i,j,t) \triangleq f(i+1,j,t) - f(i,j,t),$$
$$\nabla_E f(i,j,t) \triangleq f(i,j+1,t) - f(i,j,t),$$
$$\nabla_W f(i,j,t) \triangleq f(i,j+1,t) - f(i,j,t).$$

Artificial Intelligence & Information Analysis Lab

Anisotropic Diffusion

- Iterating this scheme can be thought as moving towards coarser image resolutions in ***scale-space***.
- Diffusion coefficients are updated at every iteration as a function of the image intensity gradient:

$$c_N(i,j,t) = g\left(\left\|\nabla f\left(i+\frac{1}{2},j,t\right)\right\|_2^2\right),$$

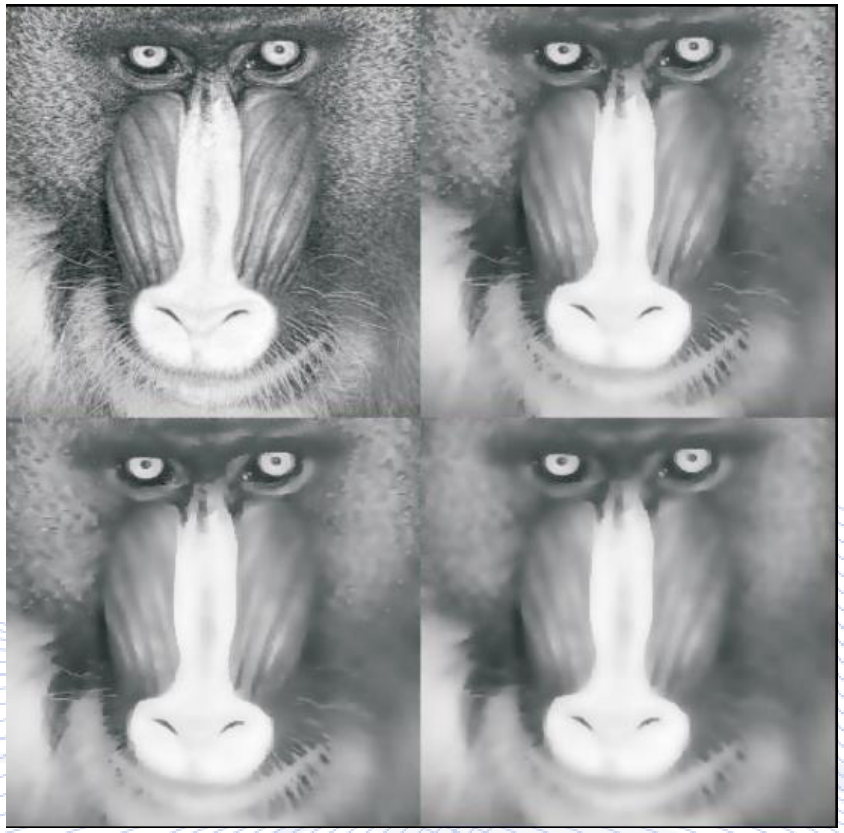
$$c_S(i,j,t) = g\left(\left\|\nabla f\left(i-\frac{1}{2},j,t\right)\right\|_2^2\right),$$

$$c_E(i,j,t) = g\left(\left\|\nabla f\left(i,j+\frac{1}{2},t\right)\right\|_2^2\right),$$

$$c_W(i,j,t) = g\left(\left\|\nabla f\left(i,j-\frac{1}{2},t\right)\right\|_2^2\right).$$

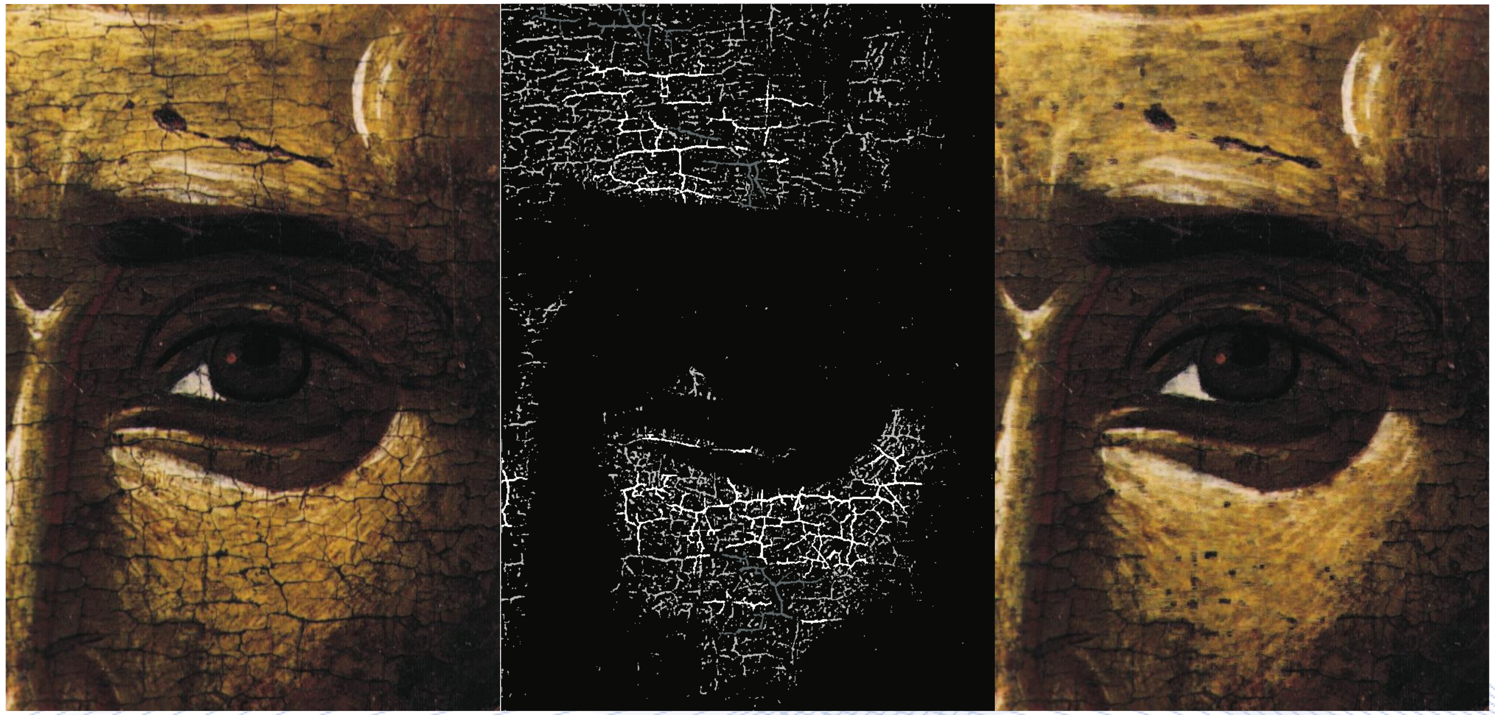
Artificial Intelligence & Information Analysis Lab

Anisotropic Diffusion



a) Original image; b-d) Various anisotropic diffusion iterations.

Anisotropic Diffusion



a) Original Byzantine painting with cracks.
b) Localized cracks.
c) Filled cracks using anisotropic diffusion.

Digital Image Filtering

- Image noise
- 2D FIR filters
- Moving average filters
- Spatial filters
- Median filters
- Digital filters based on order statistics
- Adaptive order statistic filters
- Anisotropic Diffusion
- **Image interpolation**
- Neural image filtering

Artificial Intelligence & Information Analysis Lab

Image Interpolation

Image interpolation is an important operation with many applications:

- Image zooming (e.g., for video games)
- Image upsampling (e.g., in neural autoencoders or in neural semantic region segmentation.
- Image magnification/upsampling.
- Video format conversion.

Image Interpolation

Zero-order (hold) interpolation: pixel (x, y) is assigned the value of the geometrically closest pixel in the image array:

$$f_i(n_1, n_2) = f([n_1/2], [\, n_2/2]).$$

- Repeated application: zooming by a factor of $2^n \times 2^n$.
  - For large n, regions of constant intensity (image blobs) are visible.

- It is sometimes used in video effect creation.

Image Interpolation

Linear interpolation:

$$f(x,y) = (1-\Delta_1)\,(1-\Delta_1)f(n_1,n_2) + (1-\Delta_1)\Delta_2 f(n_1,n_2+1)$$
$$+\Delta_1(1-\Delta_2)f(n_1+1,n_2) + \Delta_1\Delta_2 f(n_1+1,n_2+1),$$

where:

$$\Delta_1 = \frac{x-n_1T_1}{T_1}, \qquad \Delta_2 = \frac{y-n_2T_2}{T_2}.$$

- It is a first-order polynomial interpolation.
- It produces smoother interpolated images.

Image Interpolation

In ***p*-order interpolation**, the image is interpolated with zeros:

$$f'(n_1,n_2)=\begin{cases} f\left(\frac{n_1}{p},\frac{n_2}{p}\right) & \textit{if}\ \ n_1=pk, n_2=pl \\ 0 & \text{otherwise.}\end{cases}$$

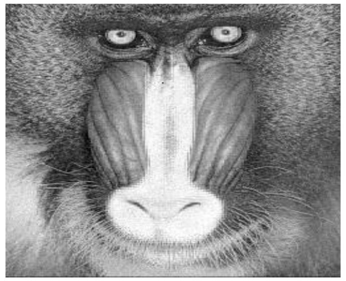
- Then, image f' is convolved p times with convolution matrix $\mathbf{H}$.
- Example of a convolution matrix $\mathbf{H}$:

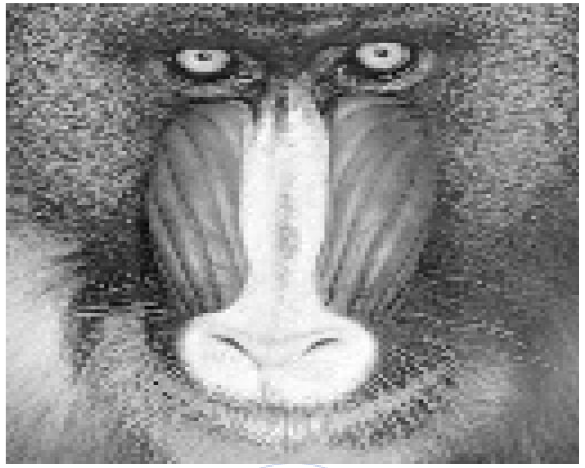
$$\mathbf{H}=\begin{bmatrix}1/4 & 1/2 & 1/4\\ 1/2 & 1 & 1/2\\ 1/4 & 1/2 & 1/4\end{bmatrix}.$$

- $p=3$ for ***cubic spline interpolation***.

Image Interpolation

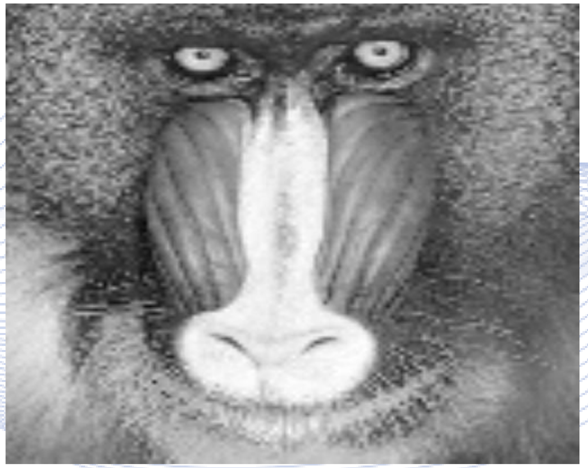
BABOON Image.

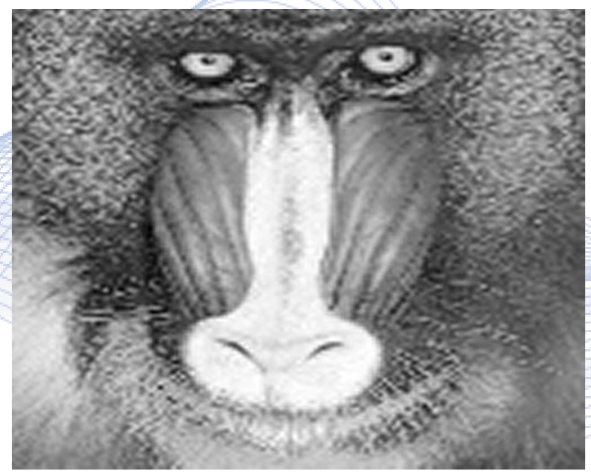




Zero-order interpolation.

Linear Interpolation.





Cubic spline Interpolation.

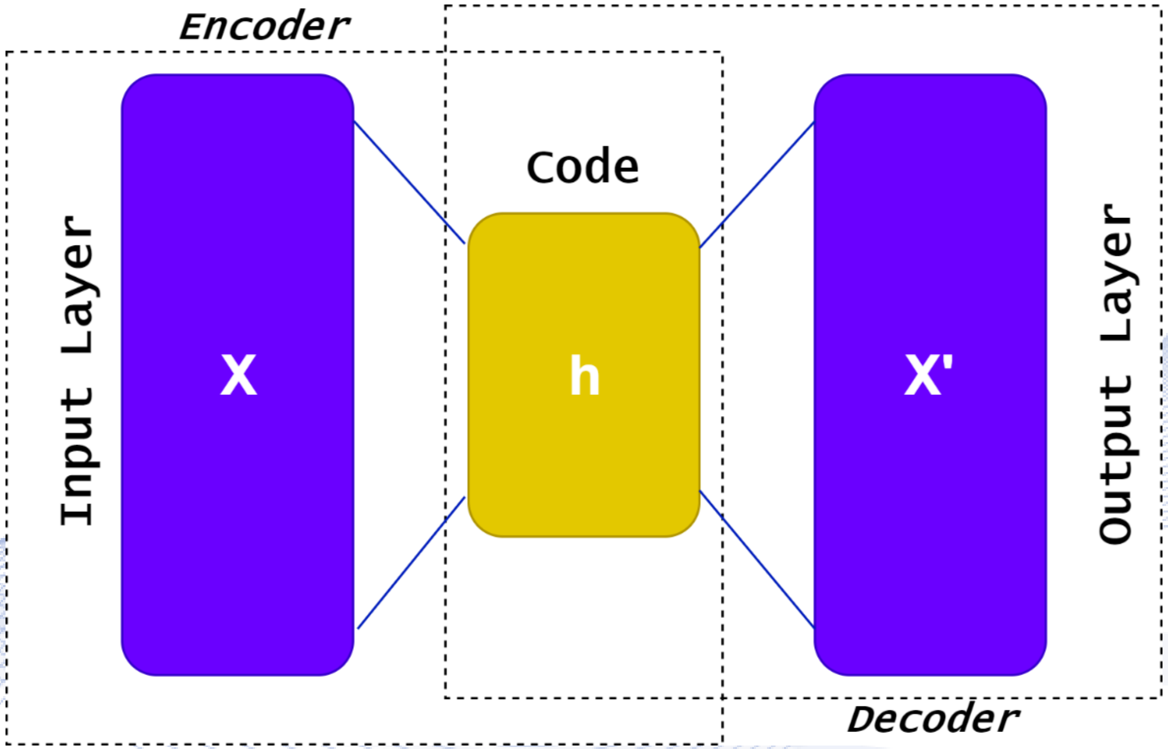
Digital Image Filtering

- Image noise
- 2D FIR filters
- Moving average filters
- Spatial filters
- Median filters
- Digital filters based on order statistics
- Adaptive order statistic filters
- Anisotropic Diffusion
- Image interpolation
- **Neural image filtering**

Neural image filtering

A classic autoencoder consists of:

- *Encoder layers*
- *Latent View Representation (code)*
- *Decoder layers*



Autoencoder architecture.

Neural image filtering

1

In general…

$$\boldsymbol{\varphi} : \mathbf{x} \rightarrow \mathbf{y}$$
$$\boldsymbol{\psi} : \mathbf{y} \rightarrow \mathbf{x}$$
$$\boldsymbol{\varphi}, \boldsymbol{\psi} = \ argmin_{\boldsymbol{\varphi},\boldsymbol{\psi}} \|\mathbf{x} - (\boldsymbol{\psi} \circ \boldsymbol{\varphi})\mathbf{x}\|$$

Where:

- $\mathbf{x}$ is the input vector
- $\mathbf{y}$ is the latent vector
- $\boldsymbol{\varphi}$ is the ***encoding*** function
- $\boldsymbol{\psi}$ is the ***decoding*** function
- $\boldsymbol{\psi} \circ \boldsymbol{\varphi}$: function synthesis

2

In the simplest case…

$$\mathbf{y} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$$
$$\mathbf{x}' = \sigma'(\mathbf{W}\mathbf{x} + \mathbf{b})$$
$$L(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|^2 = \|\mathbf{x} - \sigma'\mathbf{W}'(\sigma(\mathbf{W}\mathbf{x} + \mathbf{b}) + \mathbf{b}')\|^2$$

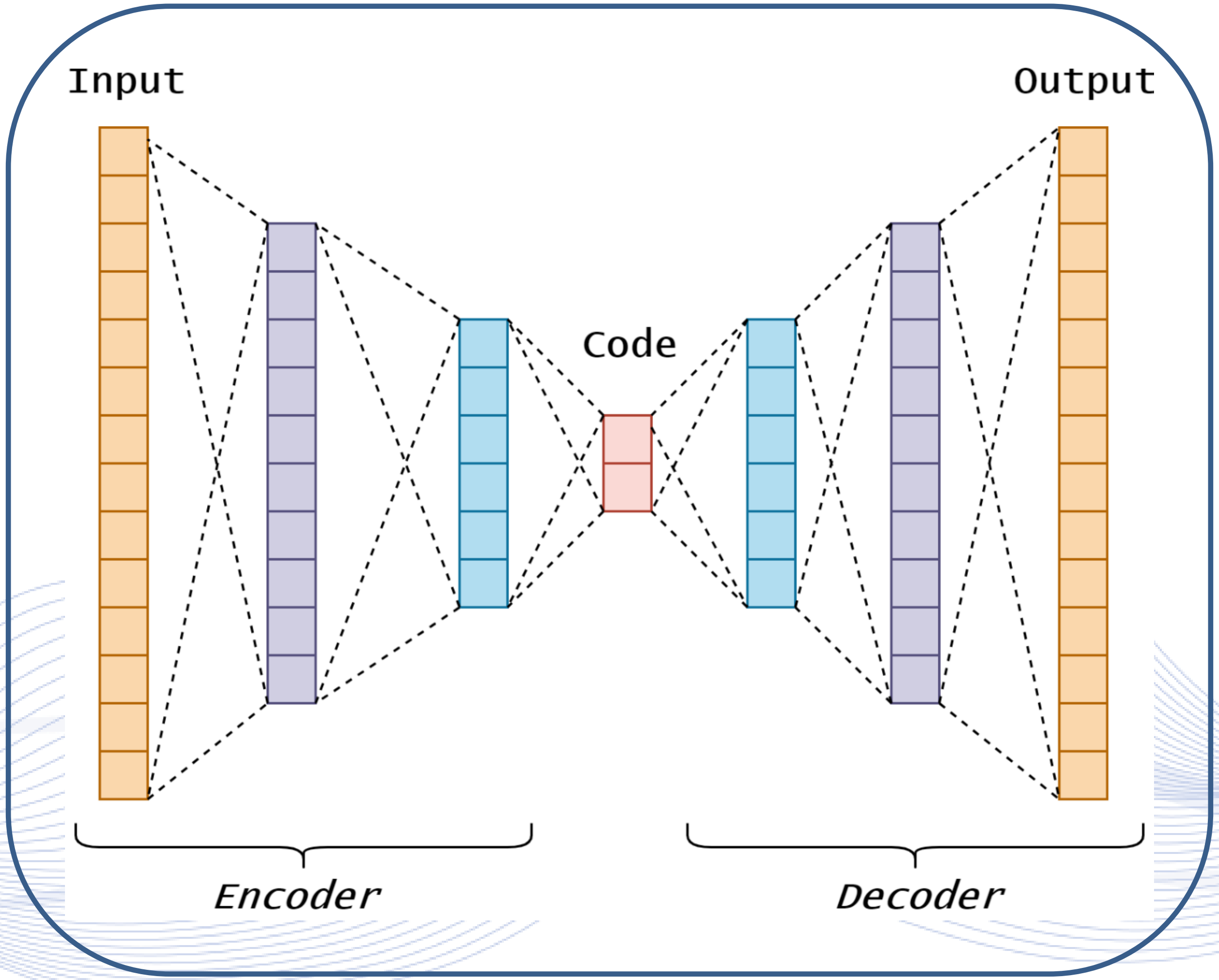
Where:

- **x'** is the reconstructed input
- **W** & **W'** are the weight matrixes
- σ & σ' are the activation functions
- **b** & **b'** are bias factors

Artificial Intelligence & Information Analysis Lab

Neural image filtering

More complex datasets require more complex architectures



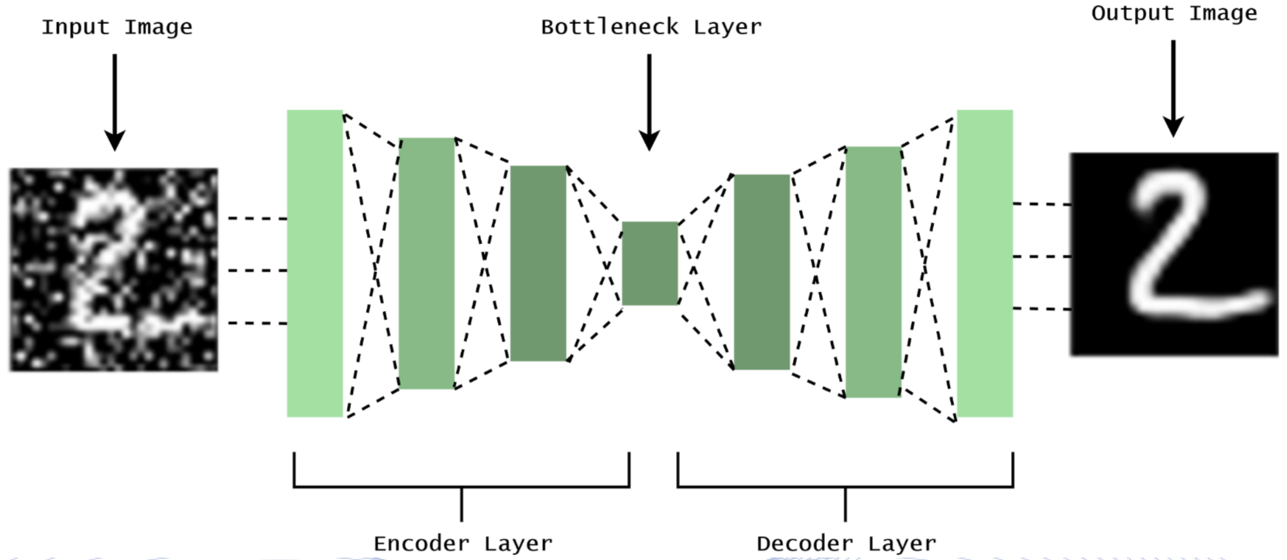
Deep Autoencoder.

A deep autoencoder consists of two, symmetrical deep-belief networks

Neural image filtering

Tries to:

1. Encode the input from a corrupted version of it
2. Undo the effect of the corruption process



Data corruption typically in 30-50% of the pixels

In the loss function the output values are compared with the original input & not the corrupted output!

Denoising Autoencoder.

Neural image filtering

Medical image denoising using convolutional denoising autoencoders.

Objective:

- Denoise medical images as a preprocessing step in medical image analysis

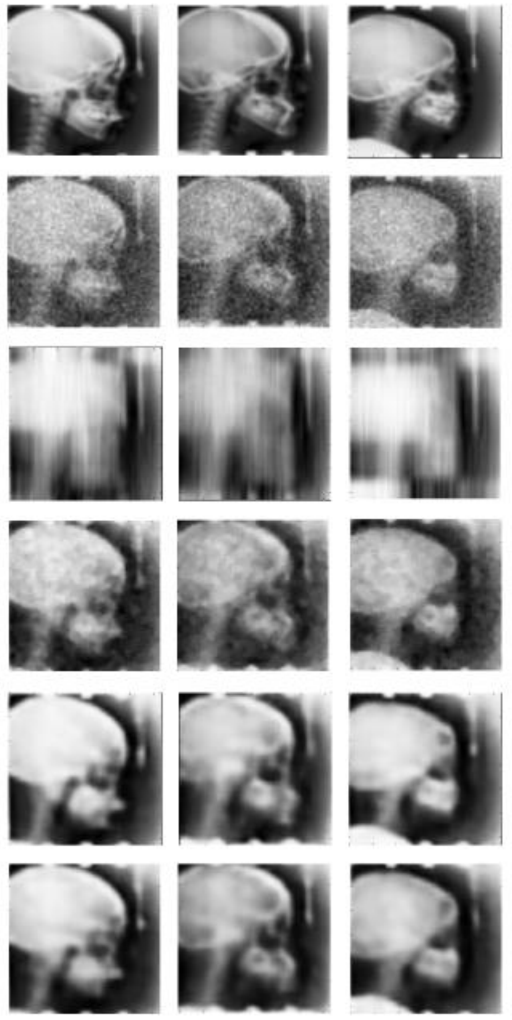
Methodology:

- Combination of convolutional, denoising & stacked autoencoder
- 2 datasets used, consisting of 722 high resolution images
- Gaussian & Poisson distribution introduced, with various noise proportion.

Neural image filtering

Medical image denoising using convolutional denoising autoencoders

Results:



Real Images

Noiser version with minimal noise

Denoising result of NL (Non-local mean filtering) means

Results of median filter

CNN DAE using smaller dataset (300 training samples)

CNN DAE using larger combined dataset

Source [MIN2017].

Neural image filtering

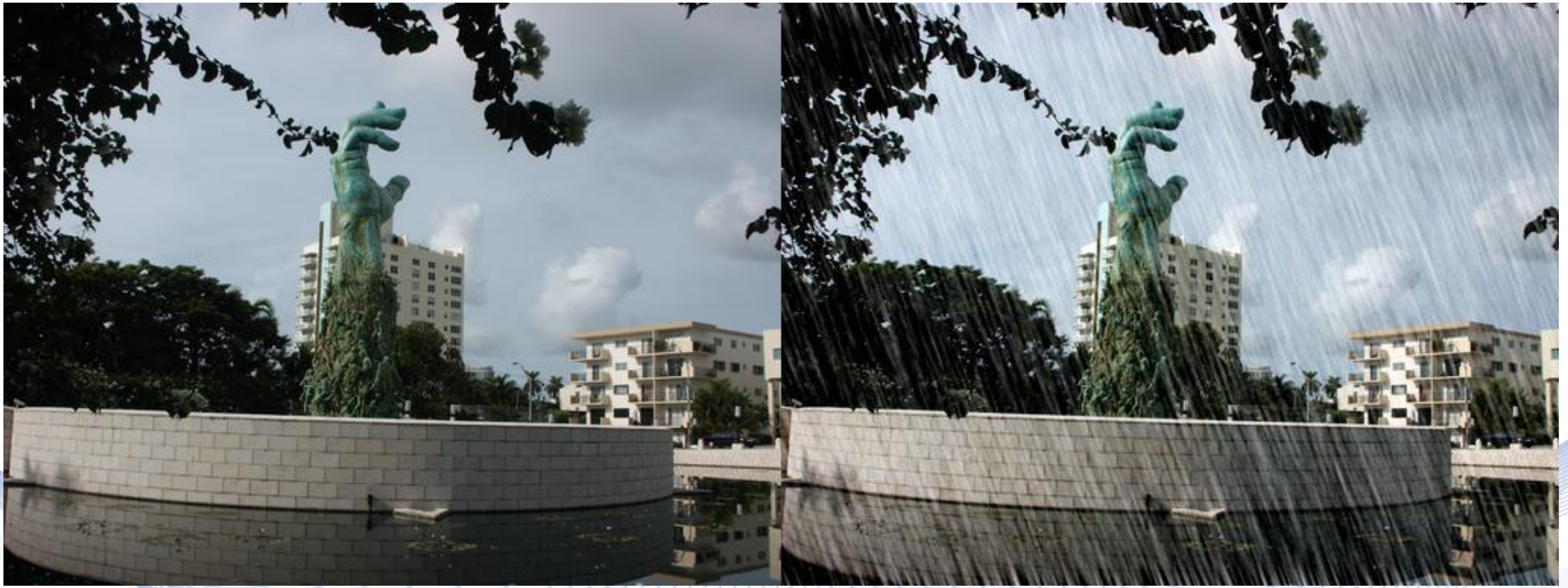


Image de-raining [DER2023].

Neural image filtering

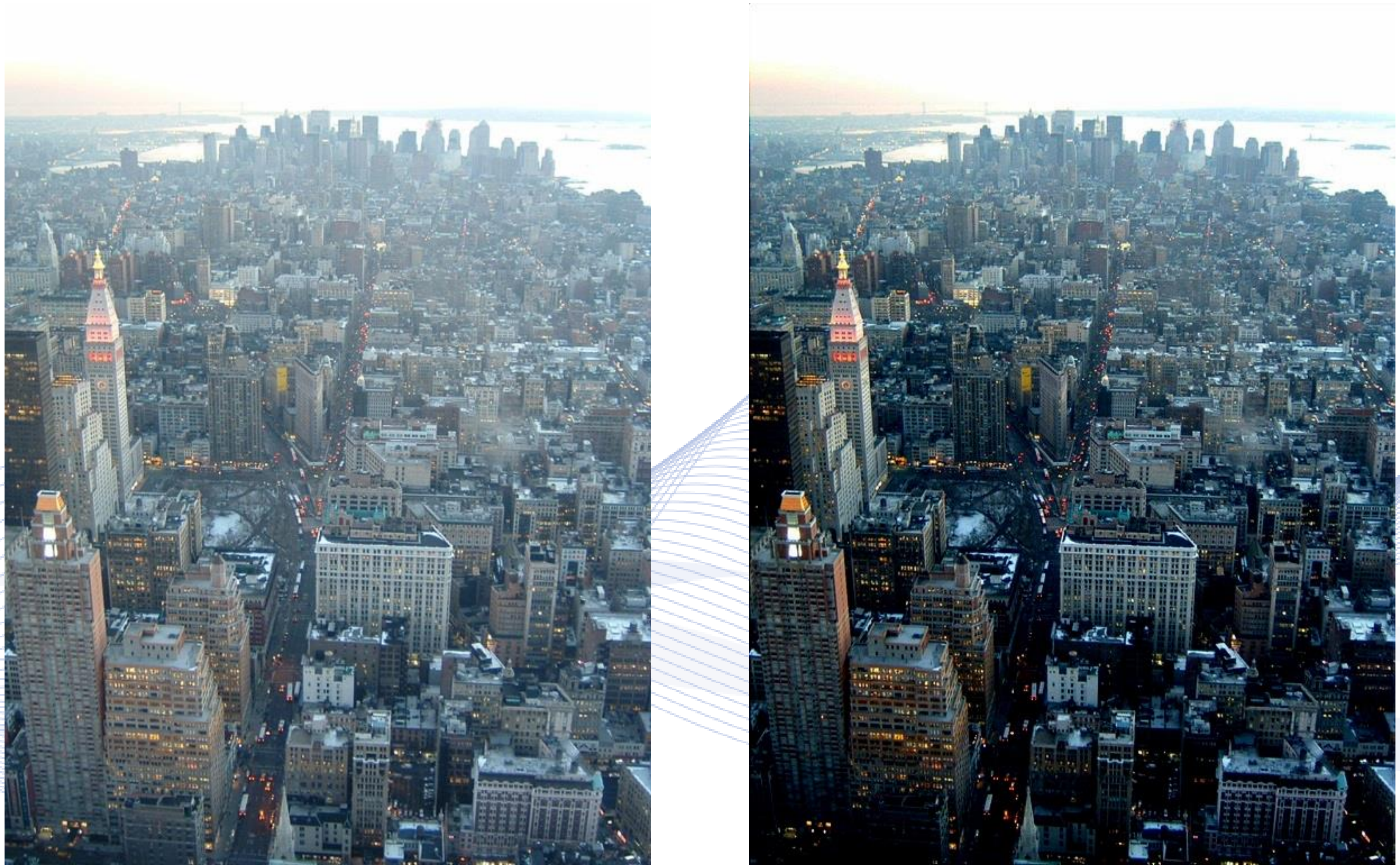


Image de-fogging.

References

[PIT2000] I. Pitas, Digital Image Processing Algorithms and Applications, J. Wiley, 2000.
[PIT2021] I. Pitas, “Computer vision”, Createspace/Amazon, in press.
[PIT2017] I. Pitas, “Digital video processing and analysis ” , China Machine Press, 2017 (in Chinese).
[PIT2013] I. Pitas, “Digital Video and Television ” , Createspace/Amazon, 2013.
[NIK2000] N. Nikolaidis and I. Pitas, 3D Image Processing Algorithms, J. Wiley, 2000.
[PIT2000] I. Pitas, Digital Image Processing Algorithms, J. Wiley 2000.
[FIS2003] R. Fisher, S. Perkins, A. Walker, E. Wolfart 2003 homepages.inf.ed.ac.uk/rbf/HIPR2/filtops.htm
[Weickert 1998] J. Weickert, “Anisotropic Diffusion in Image Processing”, Teubner, 1998.
[MIN2017] Min Chen et al. “Deep features learning for medical image analysis with convolutional autoencoder neural network". IEEE Transactions on Big Data (2017).
[DER2023] https://github.com/TheLethargicOwl/Single-Image-De-Raining-Keras

Q & A

Thank you very much for your attention!

More material in
http://icarus.csd.auth.gr/cvml-web-lecture-series/

Contact: Prof. I. Pitas
pitas@csd.auth.gr

Artificial Intelligence & Information Analysis Lab