1495 UNIVERSITY OF ABERDEEN | CELEBRATING 525 YEARS 1495 – 2020

ABERDEEN 2040

Image Mining – 2

Data Mining & Visualisation

Lecture 30

2025

Today...

- Denoising
- Summary

Denoising

Image Noise

Digital images will often contain some amount of noise or artifacts.

This could be a result of a low quality camera sensor, not enough light in the picture, from lossy data transmission, etc.

It results in images that look grainy and low quality, making further analysis more difficult.

Image Denoising

Image denoising is the process with which we remove unwanted noise from an image.

By doing so, we can re-construct and restore the image to be closer to what it should look like.

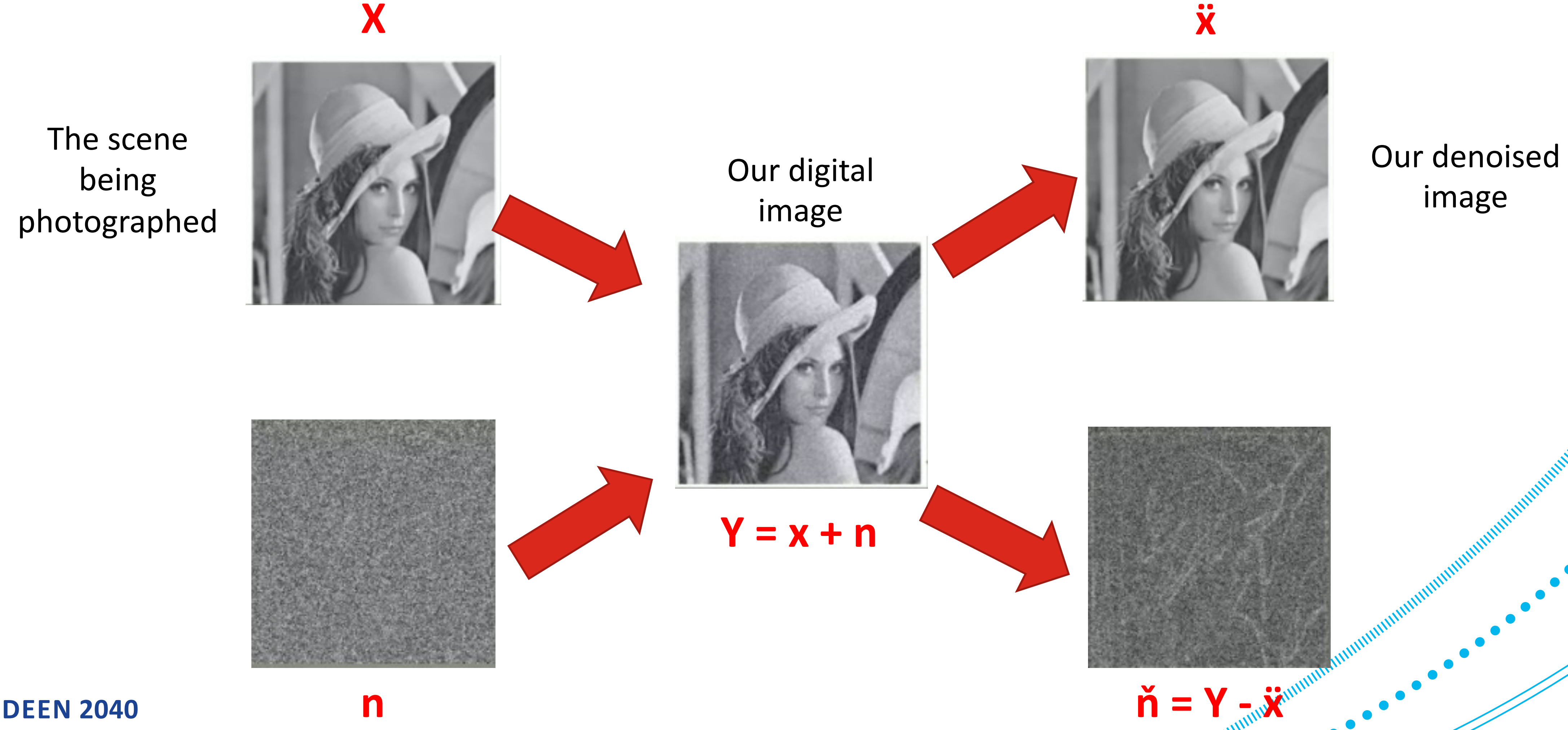
Goal of Denoising

With denoising, we want to estimate the unknown signal from the noisy data that we have.

By doing so, we can reduce the noise, while preserving as much useful information as possible.

This is often an important preprocessing step for image analysis.

Problem Definition



Noise Models

We can categorise image noise in two main ways:

Additive noise model: Noise signal gets **added** to the original signal to produce a corrupted noisy signal.

Multiplicative noise model: Noise signal gets **multiplied** to the original signal to produce a corrupted noisy signal.

Noise Models

The **additive noise model** follows the following rule:

$W(x,y) = s(x,y) + n(x,y)$

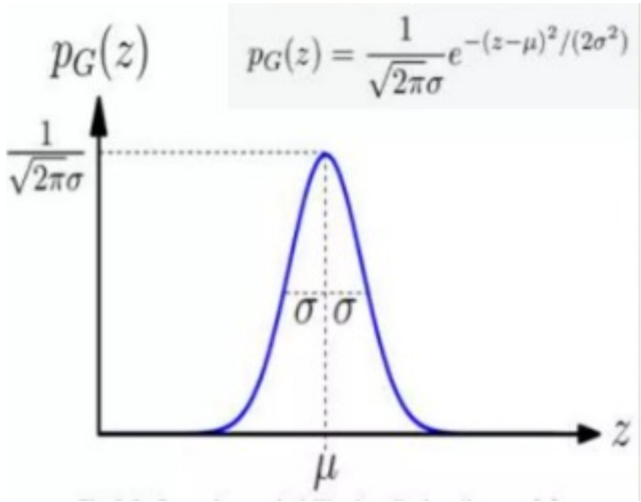
The **multiplicative noise model** follows:

$W(x,y) = s(x,y) \times n(x,y)$

Where, s(x, y) is the original image intensity at pixel location (x, y); w(x, y) is the corrupted signal; and n(x, y) denotes the noise introduced.

Types of Noise

Gaussian Noise is an example of the additive noise model. Each pixel is the sum of the true pixel value and a random noise value drawn from a Gaussian (normal) distribution.



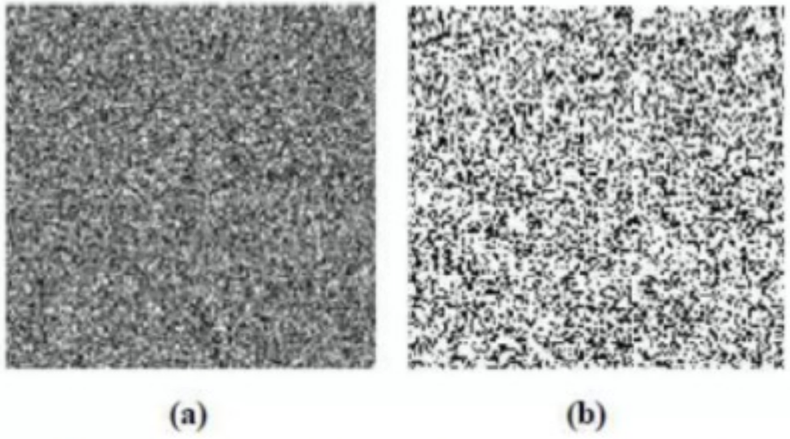
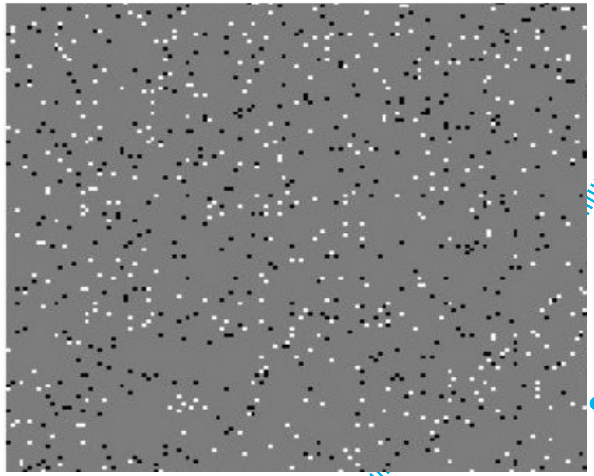


Image 2.1: Gaussian Noise Image (a) mean=0, variance 0.05 (b) mean=1.5, variance 10.

Types of Noise

Impulse Noise comes in two types; fixed value and random value.

Fixed value (salt and pepper noise) is generally caused by errors in transmission. It has only two possible values for an 8-bit image; 0 for pepper noise, and 255 for salt noise.



Random value

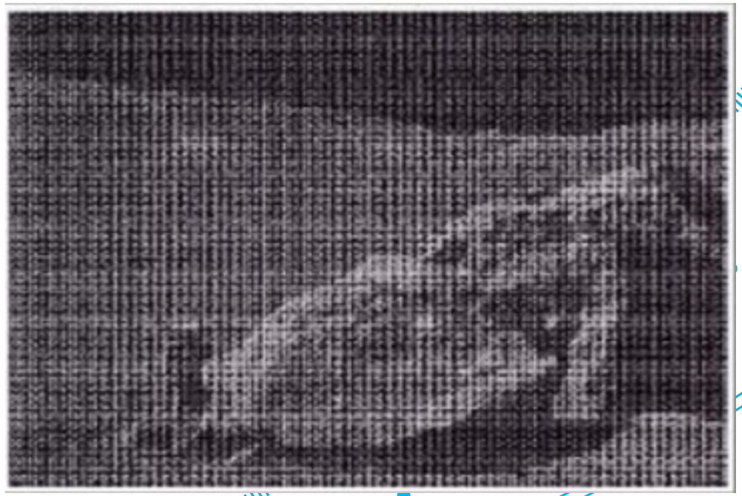
The random valued impulse noise can have any value between 0 and 255 hence its removal is quite difficult.

Types of Noise

Uniform Noise is uniformly distributed noise, where each pixel is the sum of the true pixel value and a random noise value drawn from a uniform distribution.

Periodic noise comes from electrical or electromechanical interference during image acquisition.

Image with periodic noise



Denoising Methods

Spatial domain

- Non-linear filter
- Linear filter
- Median filter
- Mean filter
- Weighted mean filter

Frequency domain

- LPF(Low pass filter) on Fourier Transform

Spatial Domain

In **spatial domain**, we deal with images as it is (simplymwe directly deal with the image matrix).

Linear Filter

Linear filters are filters where each pixel value is calculated as a weighted sum of the values within its neighbouring pixels.

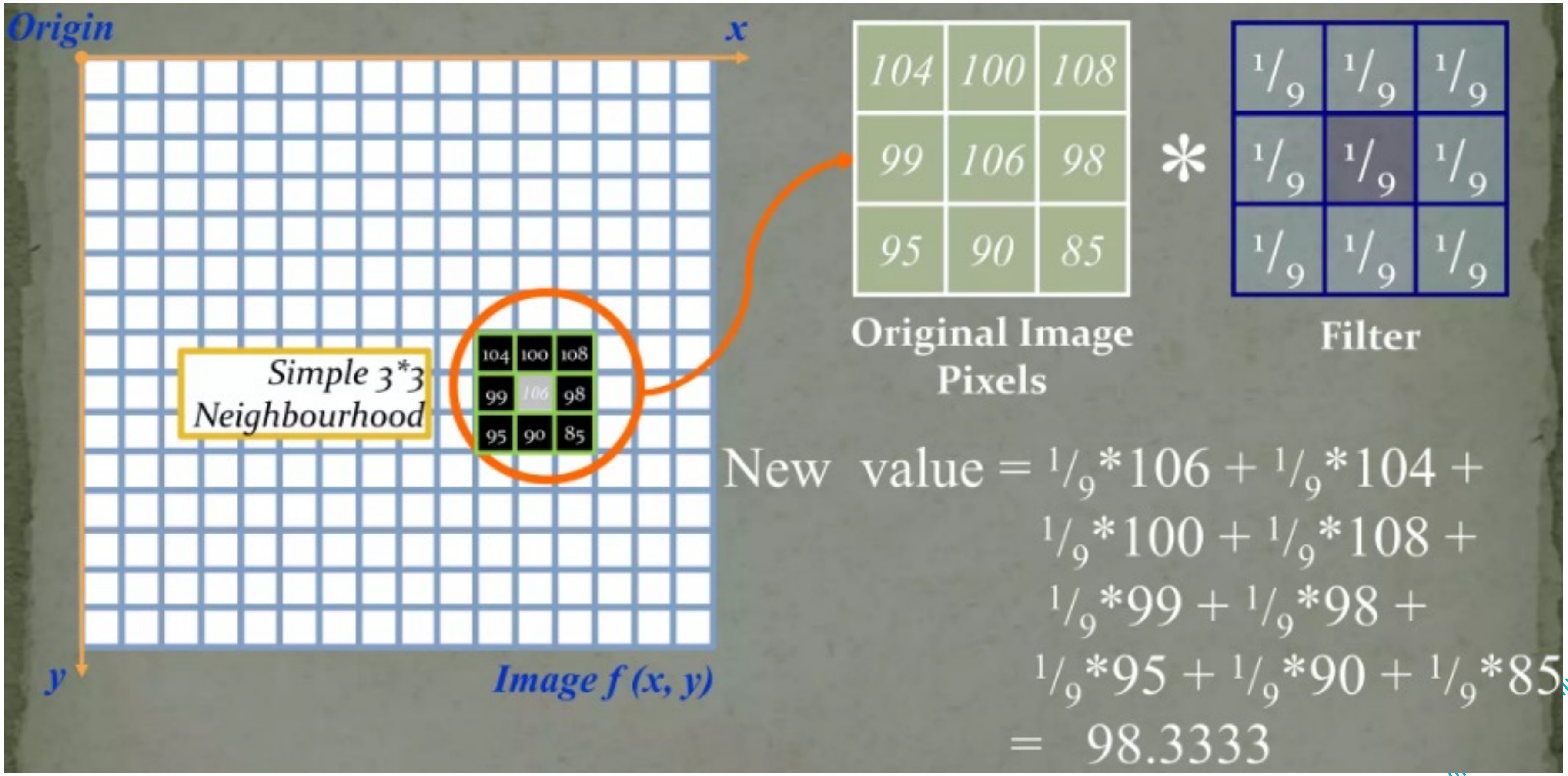
They are used to remove certain types of noise, and tend to work well with Gaussian noise.

Mean Filter

Mean filters replace the centre value of the window with the average values of its all nearest pixels values together with itself.

They are simple to design, and implement, but they tend to blur sharp edges, and can destroy lines and other fine details within an image.

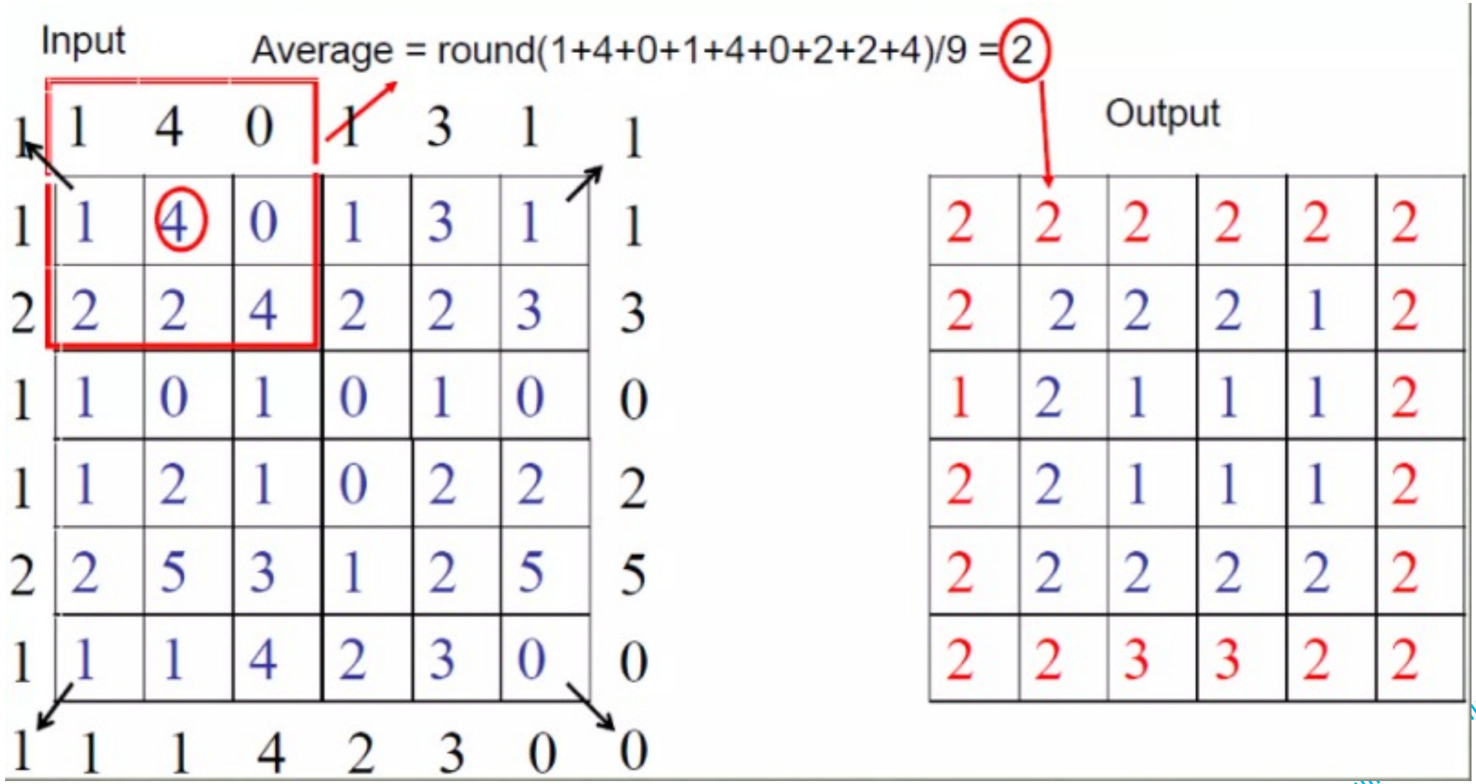
Mean Filter



Origin
x
y
Image f (x, y)
Simple 3*3
Neighbourhood
104 100 108
99 106 98
95 90 85
*
1/9 1/9 1/9
1/9 1/9 1/9
1/9 1/9 1/9
Original Image
Pixels
Filter
New value = 1/9*106 + 1/9*104 +
1/9*100 + 1/9*108 +
1/9*99 + 1/9*98 +
1/9*95 + 1/9*90 + 1/9*85
= 98.3333

Mean Filter

We can extend the values outside the border with the values at the boundary.



Mean Filter

- Original image

Image corrupted by Gaussian noise

De-noising by mean filter

Weighted Mean Filter

Mean filters replace the centre value of the window with the average values of its all nearest pixels values together with itself.

They are simple to design, and implement, but they tend to blur sharp edges, and can destroy lines and other fine details within an image.

Weighted Mean Filter

- Simple averaging (mean filter) of neighbouring pixels lead to over-smoothing
- Instead if weighting all neighbouring pixels equally, assign higher weights to pixels that closer to the pixel being convolved.

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Mean filter

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Weighted mean filter

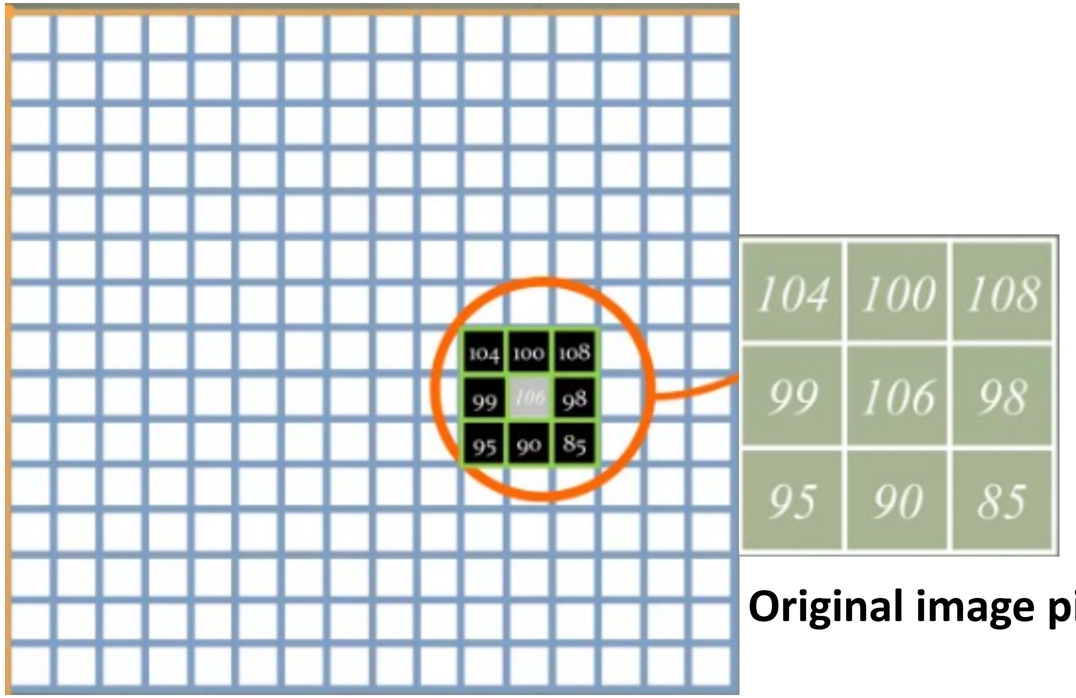
Non-Linear Filters

- Can preserve edges.
- Very effective at removing impulsive noise.
- They are more powerful than linear filters because they are able to reduce noise levels without blurring edges.
- Difficult to design than linear filters.
- Median Filter.

Median Filter

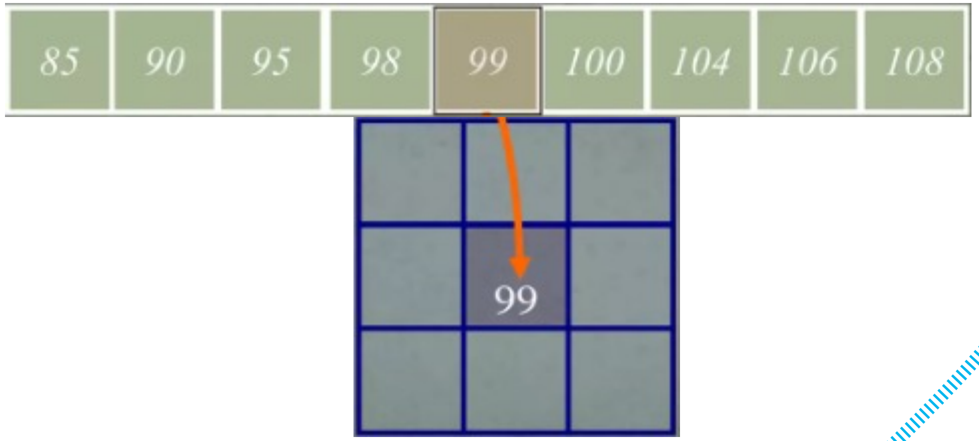
- Median filter is one of the most important filters to remove random valued impulse noise.
- In this filter the value of corrupted pixel in noisy image is replaced by median value if corresponding window.
- The ***median*** is calculated by first sorting all the pixel values into ***ascending*** order and then replace the pixel being calculated with the ***middle*** pixel value.
- **Salt and pepper** noise.

Median Filter



Original image pixels

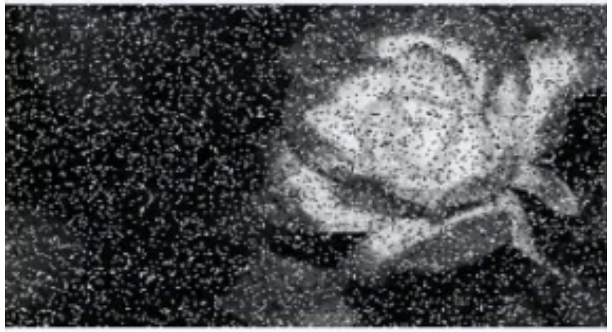
Sort the pixel values



- The above is repeated for every pixel in the original image

Median Filter

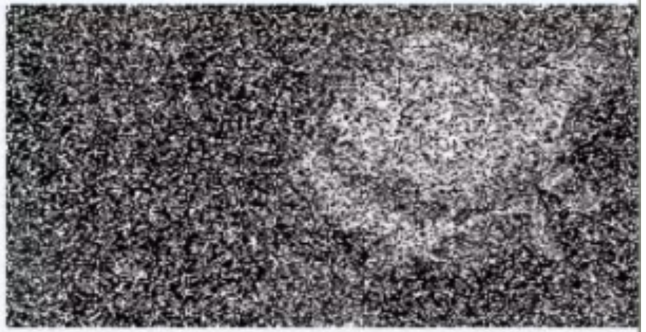
Original



Salt & pepper
%20

Denoising by
median filter

Original



Salt & pepper
%60

Denoising by
median filter

Summary

Bringing it all Together

So far, we’ve outlined some basic concepts about how digital images can be created and preprocessed.

There are several contexts in which we might want to mine and analyse such image data.

Bringing it all Together

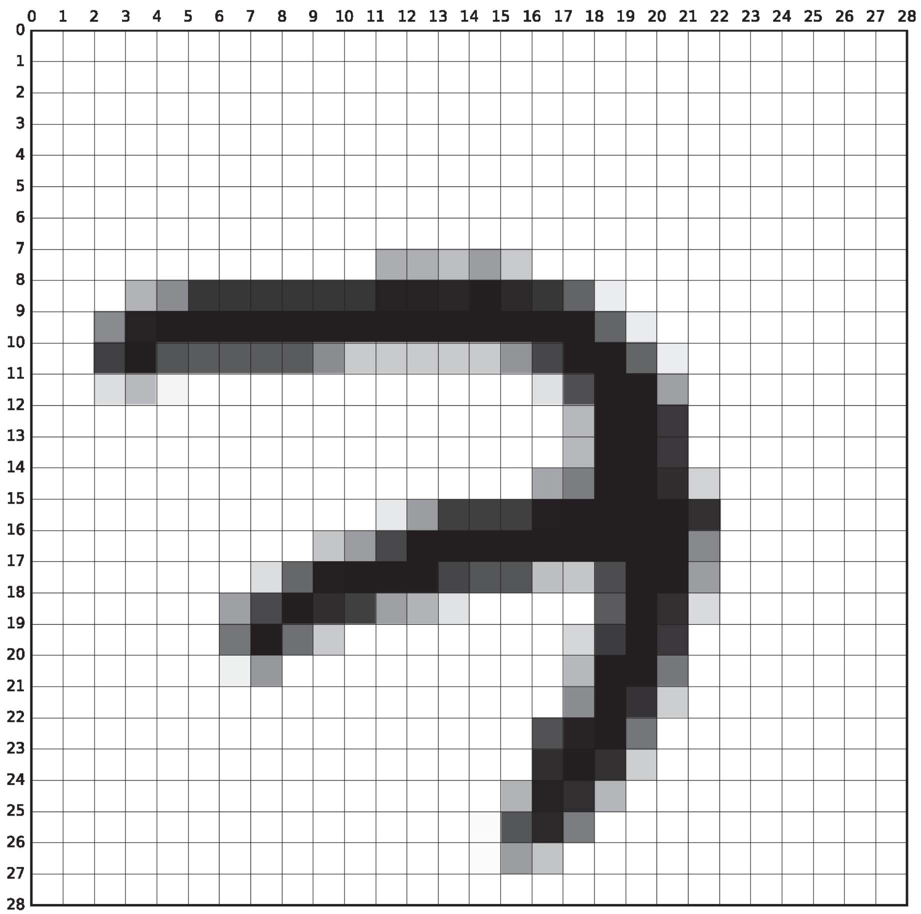
By analysing and mining images, we can leverage the information contained within, and use that information for some downstream analysis.

We've already seen one way in which we might approach analysing an image, during the dimensionality reduction lecture.

Image Representation

During that lecture, we outlined how a 28x28 grayscale image could be represented as a 784-dimensional vector.

Each column could then represent a particular pixel in the 28x28 array, containing a value representing that pixel's grayscale intensity rating.



Optical Character Recognition

With enough images representing different characters, we could use this approach to train an ML model for optical character recognition (OCR).

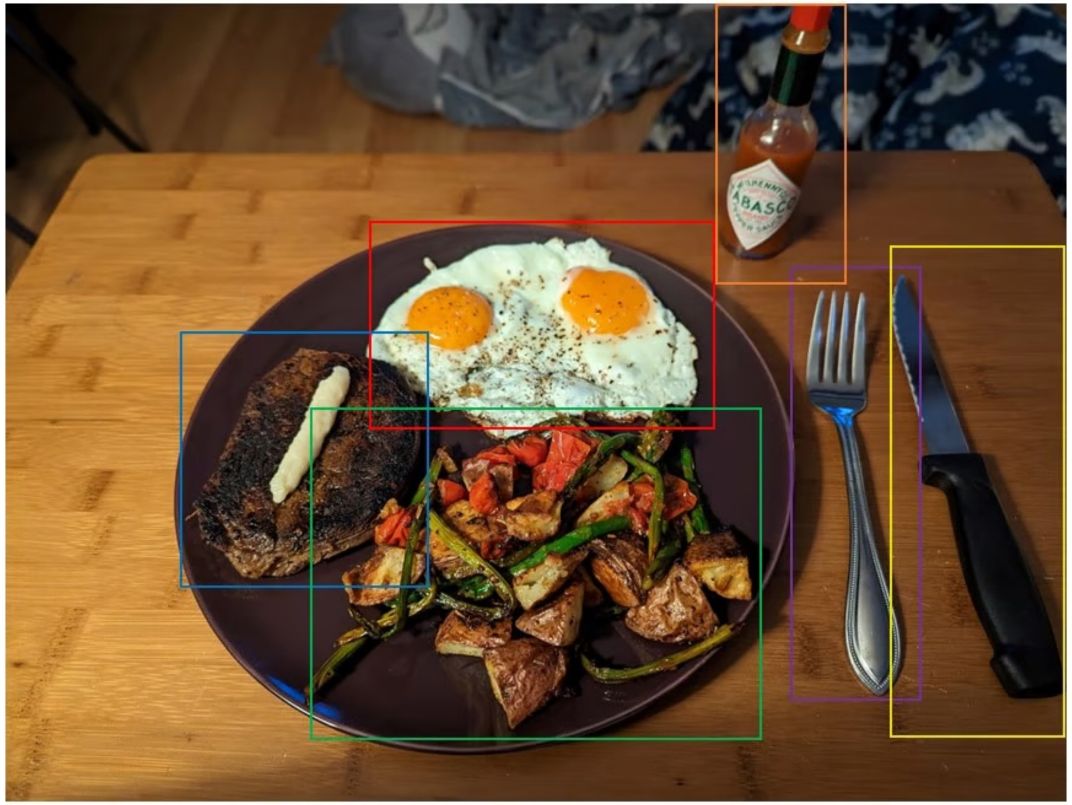
From there, we could look to extract text from the raw image data.

Object Recognition

Similarly, we could train, or use pre-existing, ML models to provide object recognition, to identify objects within images.

Or, facial recognition models, to identify individuals within images.

Object Detection



Eggs, Steak, Vegetables, Fork, Knife, Bottle

Image source: https://www.datacamp.com/tutorial/seeing-like-a-machine-a-beginners-guide-to-image-analysis-in-machine-learning

Image Mining

Implementing these types of computer vision techniques is beyond the scope of this data mining course.

However, hopefully you can see how the ability to extract information from images could open various avenues for downstream data analysis.