module4 · image analysis

introduction to image analysis

Introduction

Image data can be sourced from various places including:

- Photographs
- Security cameras, smart phone cameras
- Scanned documents (e.g., bank checks)
- Medical images
- Artwork
- Science (e.g., biology)
- Mechanical processes (e.g., flow through a duct)

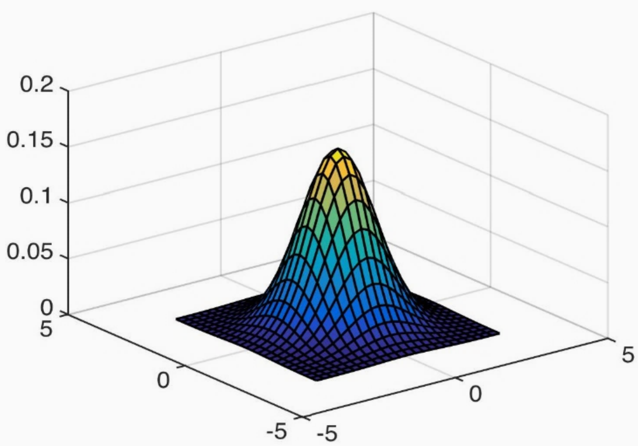
Image data can be transformed and manipulated in many ways including:

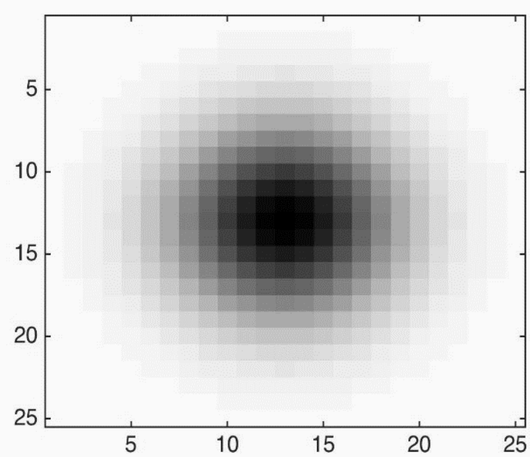
- Reduce noise
  - Not just to make images look nicer visually, but also for detecting what objects are inside the images. Imagine a self-driving car trying to reduce noise from the rain on its camera to identify where the center line of the road and the curb is.
- Find interesting features (edges, corners)
  - The interesting features try to help discern what the essence of the image is; a way to summarize the image in the way needed for some other purpose.
- Use the features for classifying images
  - Finding out what is inside the image

Blurring and Denoising

Image processing is a well-studied and highly mature field. **Blurring** for the purpose of **Denoising** is one of the most popular techniques applied to images. The latter involves convolution with what is called a **filter**; the **filter** is typically **Gaussian**.

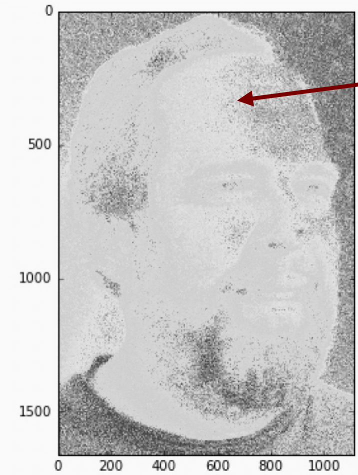
The following is a 2-dimensional plotted Gaussian (left) and a top view of the Gaussian (right). The right image displays larger values in the middle and smaller values on the outside:

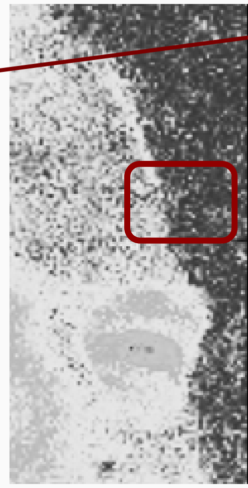




The process will take the latter Gaussian and **'convolve'** it with an image.

The following illustrates **convolution** of an image with a **filter** (typically Gaussian):



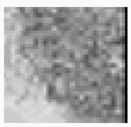
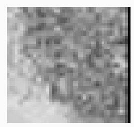
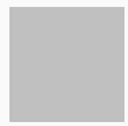


The pixel intensity of the original image (left) is between 0 and 255.

In order to **denoise** the original image, a chosen area of noise is zoomed in on the image (right).

The simplest way to smooth out the original image would be to replace the center of each pane with the average of pixel intensities in the entire square:

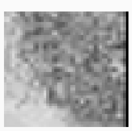
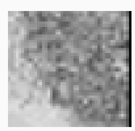
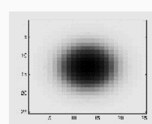
The following measure takes the simple average of the pixel intensities:

Average() $= \sum_{pixels}$  $\bullet$ 

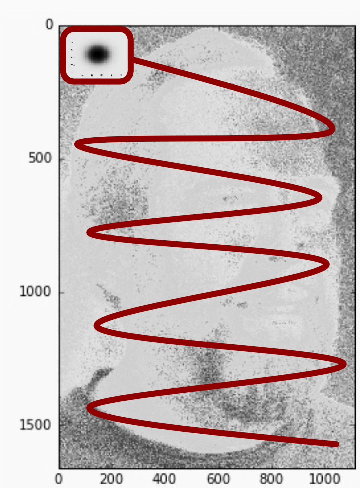
The average pixel intensity of the pane can be computed by multiplying each pixel intensity by $\frac{\mathbf{1}}{\textbf{the number of pixels}}$ and sum all of the resulting values. The average value computed in the pane can replace the middle pixel in the image by its average.

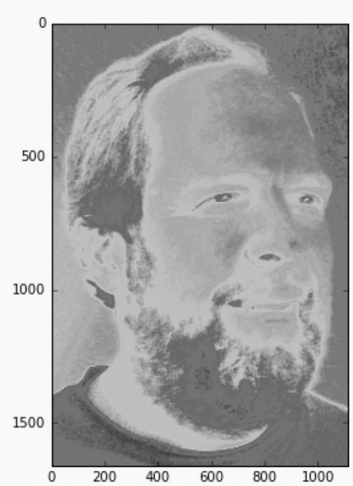
A disadvantage of the above process is trying to estimate what belongs in the middle of the square; it should depend more on the pixels near the middle opposed to the edges.

Rather than an average weight for each individual square, the center pixels are more weighted than the outer. The latter is where Gaussian is applied as a weighting function:

Weighted Average () $= \sum_{pixels}$  $\bullet$ 

The process is repeated over every spot in the image, taking a weighted average of the pixels. The pixels in the nearby area are each weighted according to the height of the Gaussian; the new intensity for that point in the center. The process of computing the latter all over the image is **convolution**:



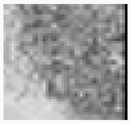


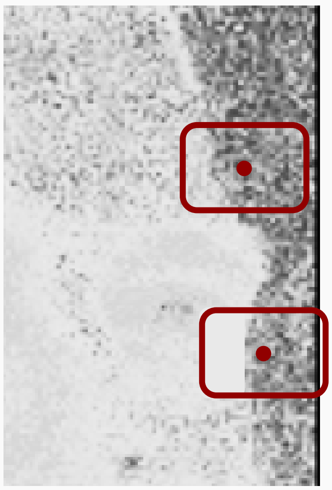
The resulting image has been **denoised**. However, if the Gaussian was too wide, the image would have become too blurred. If the bandwidth was too narrow, the image would remain noisy.

It is not uncommon for this process to blur edges and lines.

Denoising with a Median Filter

A **Median Filter** computes the median of the intensities as opposed to the weighted average of the Gaussian Filter. The reason for applying a **Median Filter** is due to **preserving edges**. Gaussian Filters are prone to blurring edges in an image.

Median()

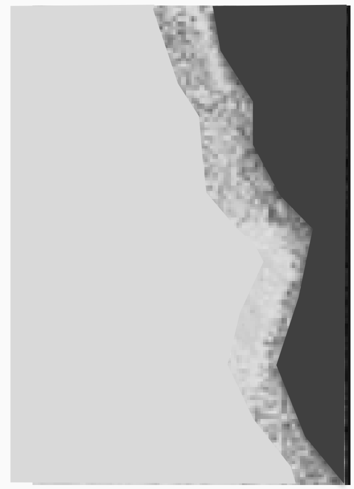


Median filters tend to be better at preserving edges than Gaussian filters.

In the illustrated pane (right), a Gaussian Filter would be influenced by the lighter portion to the left side of the pane. With a **Median Filter**, the pixel in the center will maintain its intensity measurement regardless.

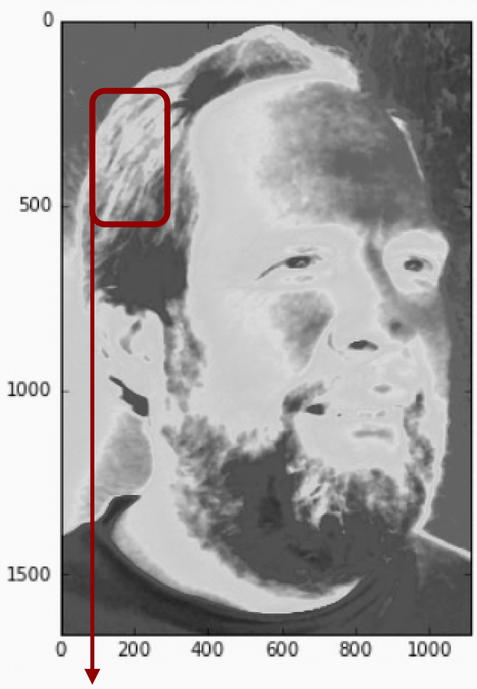
Even if a portion was extremely contrasted within the pain, the median itself would not be affected as seen in the lower pane in the left image.

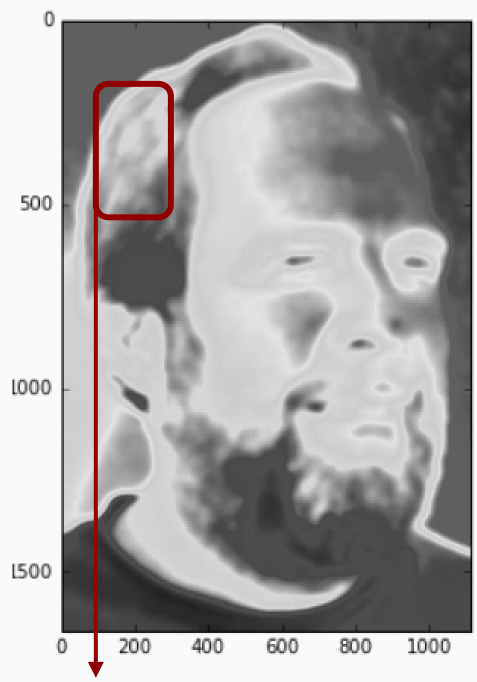
Therefore, the edges will mostly be preserved in an image and smoothing achieved in the appropriate areas.



The following images display an image with a Gaussian Filter (left image) and a Median Filter (right image) using the same sized filter.

A panel of the image is zoomed in below to illustrate the preservation of edges that the Median Filter is able to maintain.





Median Filter

Gaussian Filter