Neural Nets

Schwartz

September 5, 2016

Making machines that think

The perceptron – a single layer feedforward neural network – was invented in 1957 at the Cornell Aeronautical Laboratory by Frank Rosenblatt with funding from the United States Office of Naval Research. In a 1958 press conference organized by the US Navy, based on Rosenblatt's statements, The New York Times reported the perceptron to be "the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence."

Upon his death, the titular Reverend Thomas Bayes (1701–1761) of the renowned theorem left an unpublished manuscript deriving the distribution for the parameter of a binomial distribution. This found its way to Richard Price, who prior to posthumously reading the manuscript at the Royal Society, dutifully edited the manuscript and added an introduction that forms much of the philosophical basis for Bayesian analysis. Mathematical historians have argued that "by modern standards, we should refer to the Bayes-Price rule. Price discovered Bayes' work, recognized its importance, corrected it, contributed to the article, and found a use for it. The modern convention of employing Bayes' name alone is unfair but so entrenched that anything else makes little sense." Quite unaware of Bayes' work, the French mathematician Pierre-Simon Laplace reproduced and extended Bayes' results in 1774. It was also suggested that the blind English mathematician Nicholas Saunderson discovered the theorem some time before Bayes, but this is disputed.

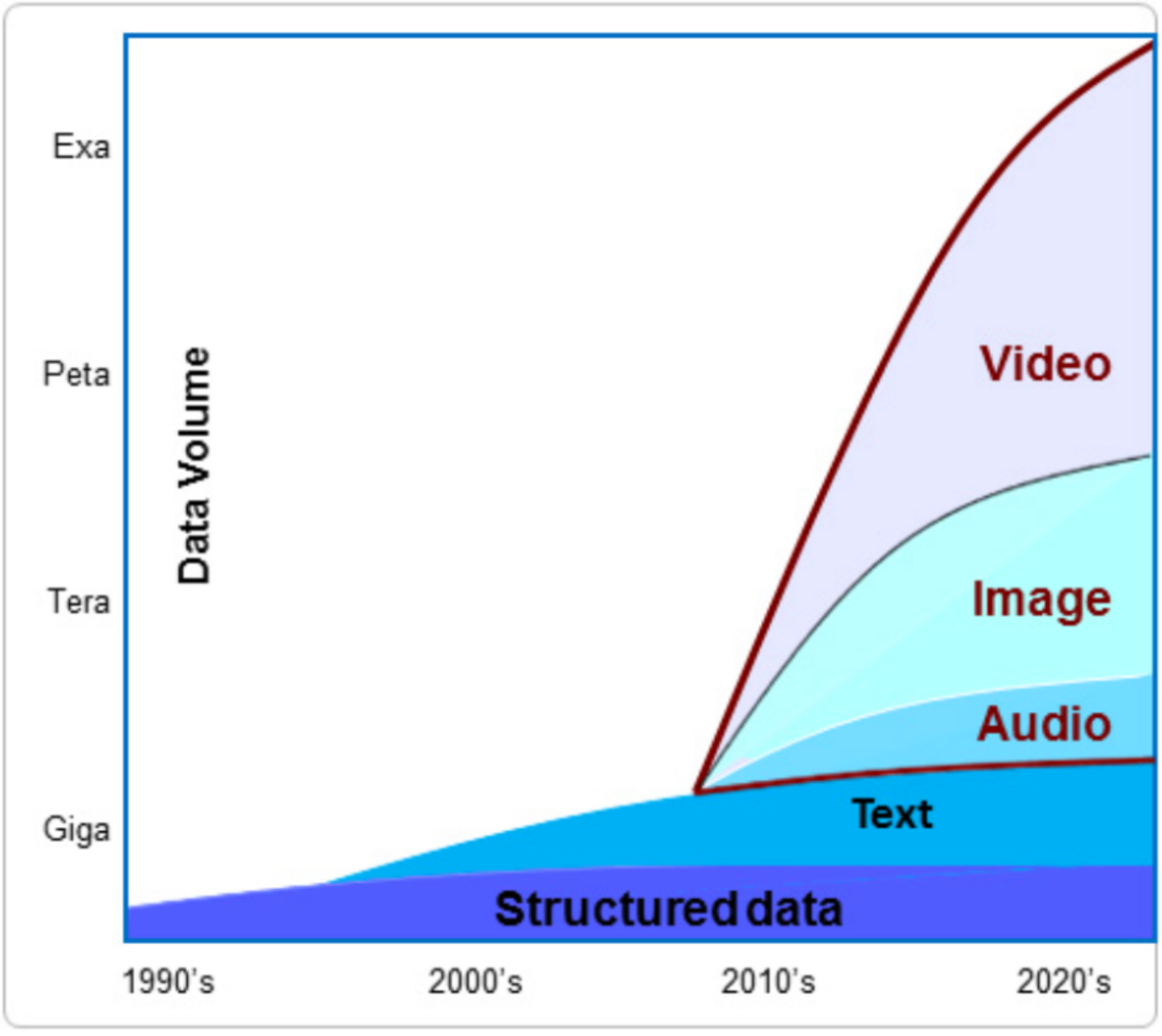
The elegant simplicity and effectiveness of Bayes' theorem for "learning" has prompted some psychologists to ask if the human brain itself might be a Bayesian-reasoning machine. They suggest that the Bayesian capacity to draw strong inferences from sparse data could be crucial to the way the mind perceives the world, plans actions, comprehends and learns language, reasons from correlation to causation, and even understands the goals and beliefs of other minds. The key to successful Bayesian reasoning is not in having an extensive, unbiased sample, which is the eternal worry of frequentists, but rather in having an appropriate "prior". This prior is an assumption about the way the world works. With the correct prior, even a single piece of data can be used to make meaningful Bayesian predictions. By contrast, frequentism is perhaps not well suited to making decisions on the basis of limited information – which is something that people have to do all the time.

It is thought by cognitive neuroscientists that neocortical development occurs in sequential layers, driven by waves of nerve growth factors which result in a self-organizating system. Modern so-called *deep learning* successors of the perceptron use Bayesian model fitting techniques to analogously sequentially train layer upon layer of neurons to produce unsupervised (self-organizing) classification networks.

Objectives

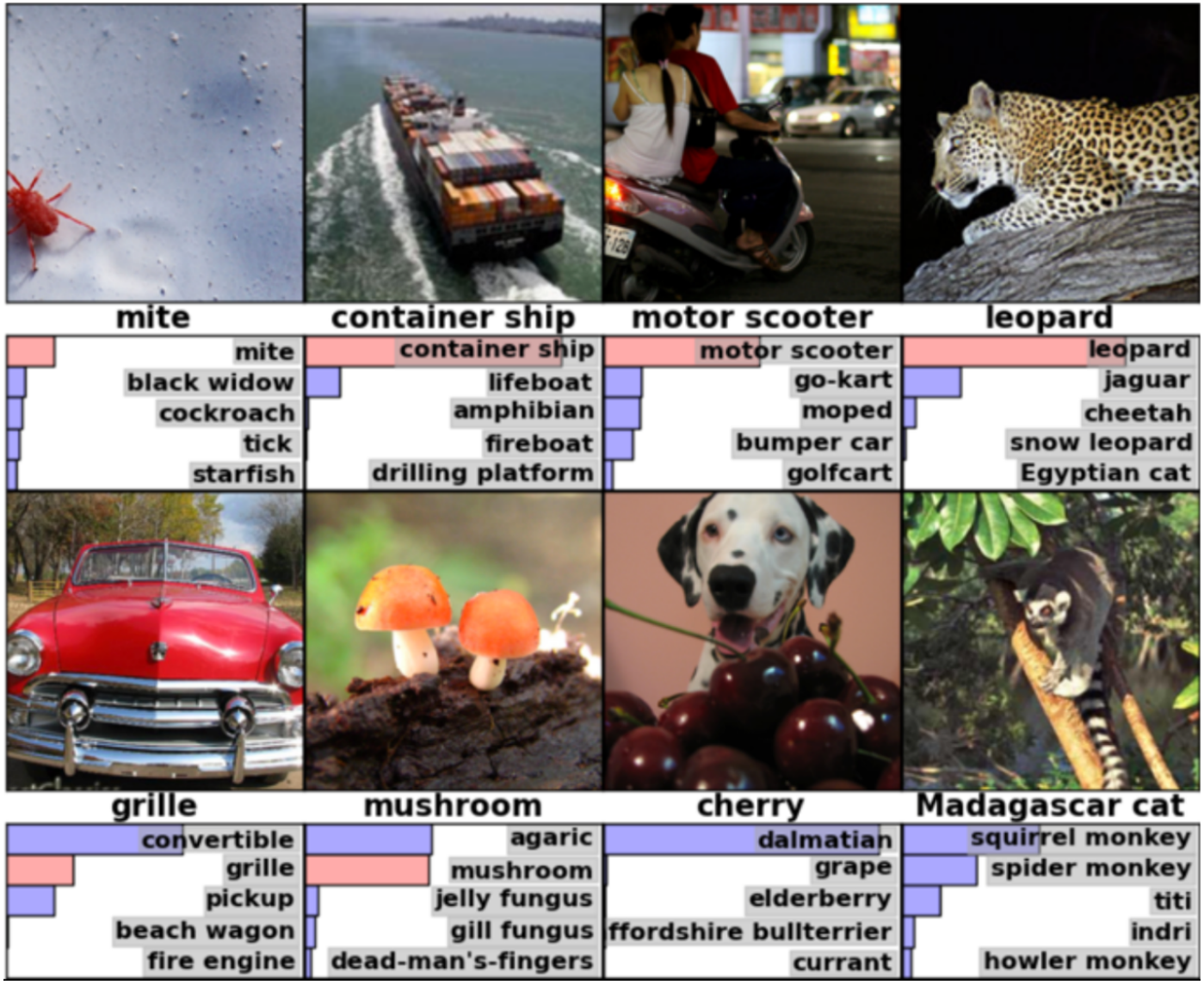
- understand image data
- understand image tools and pipelines
- understand convolutions
- understand neural networks
- understand how they can recapitulate other methodologies
- understand how activating functions provides power
- understand how layers provide abstract feature encoding
- understand how neural networks are trained, i.e., "backpropegation is gradient decent is the chain rule"
- understand that backpropegation doesn't always work well
- understand convolutional networks structure and features
- understand there is new thing called deep learning that can be viewed as an extension of neural networks

Well that's interesting...



What are images?

How can computers do this??



How *might* computers be able to do this?

How *might* computers be able to do this?

- Compress data
- Keep the search simple
- Segment image into objects

Image processing *tasks*

- Read

Image processing *tasks*

- Read

  ./dog_cat

  ./dog_cat/dog ./dog_cat/cat

Image processing *tasks*

- Read

  ./dog_cat

  ./dog_cat/dog ./dog_cat/cat

- image.shape = (width, height) or (width, height,3)

Image processing *tasks*

- Read
  ./dog_cat
  ./dog_cat/dog ./dog_cat/cat

- image.shape = (width, height) or (width, height,3)
  A *tensor* (e.g., a color image) is a matrix of vectors

$$\begin{bmatrix} RGB & RGB & \cdots & RGB \\ RGB & RGB & \cdots & RGB \\ \vdots & \vdots & \ddots & \vdots \\ RGB & RGB & \cdots & RGB \end{bmatrix}$$

Image processing *tasks*

- Resize

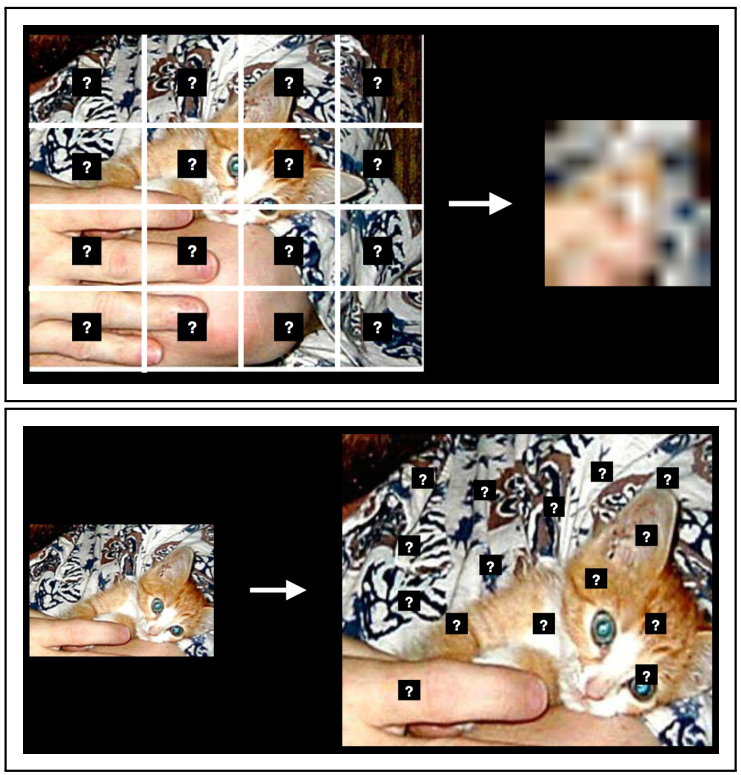
Image processing *tasks*

- Resize to create a uniform feature set

Image processing *tasks*

- Resize to create a uniform feature set

  Down/Upsampling – *not cropping* – bi-linear interpolation



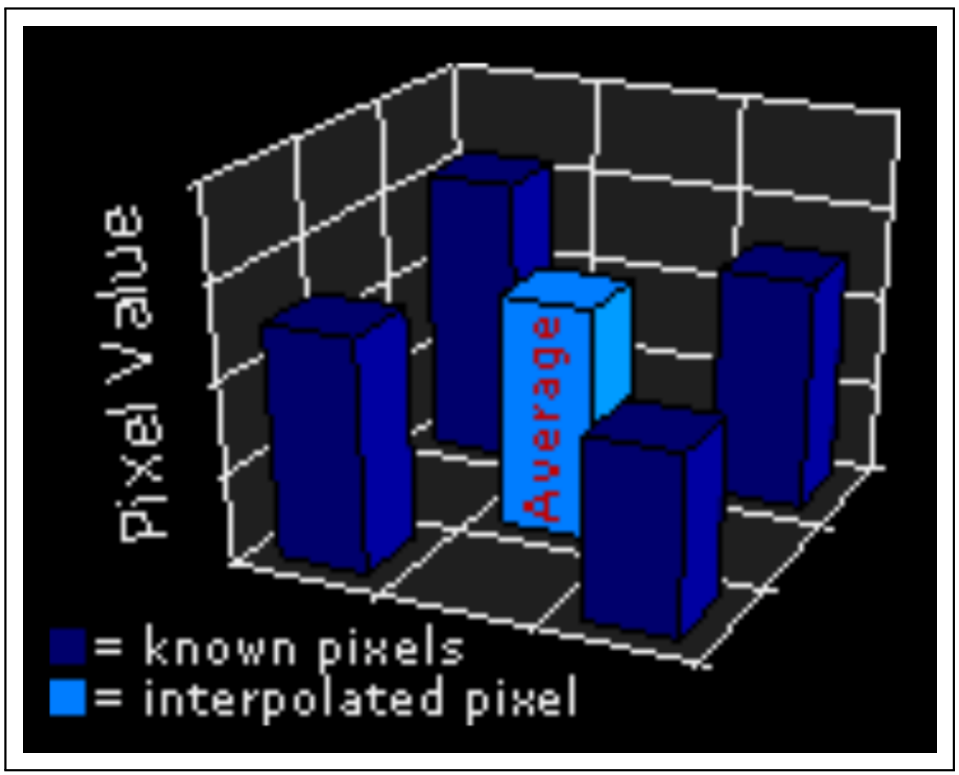
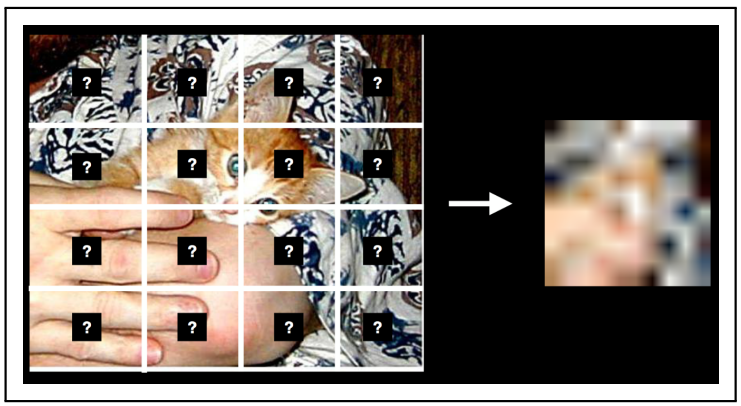
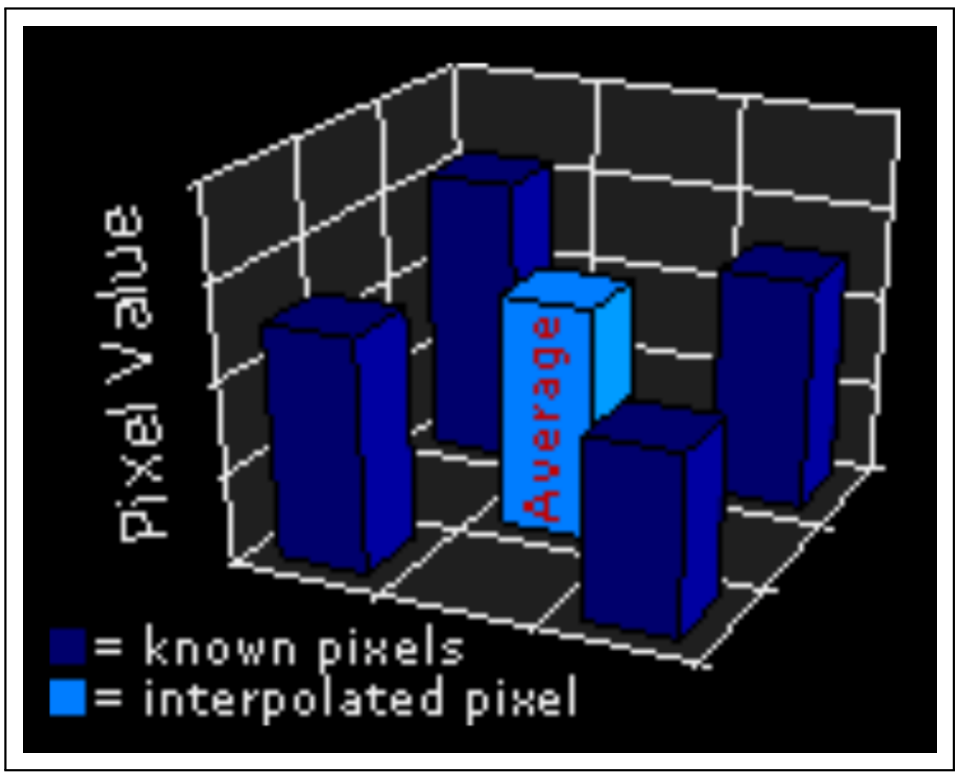


Image processing *tasks*

- Resize to create a uniform feature set

  Down/Upsampling – *not cropping* – bi-linear interpolation



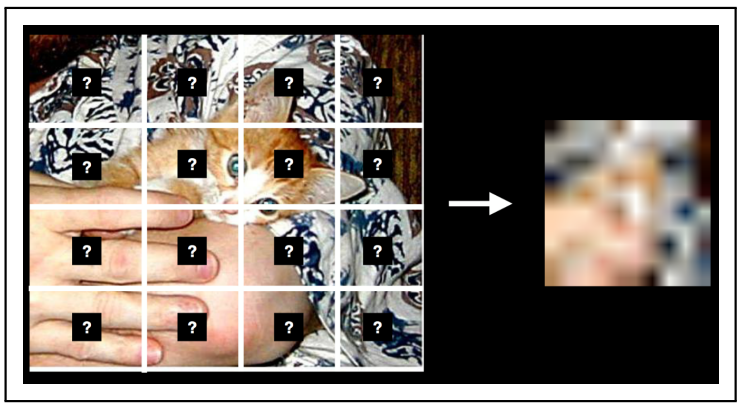


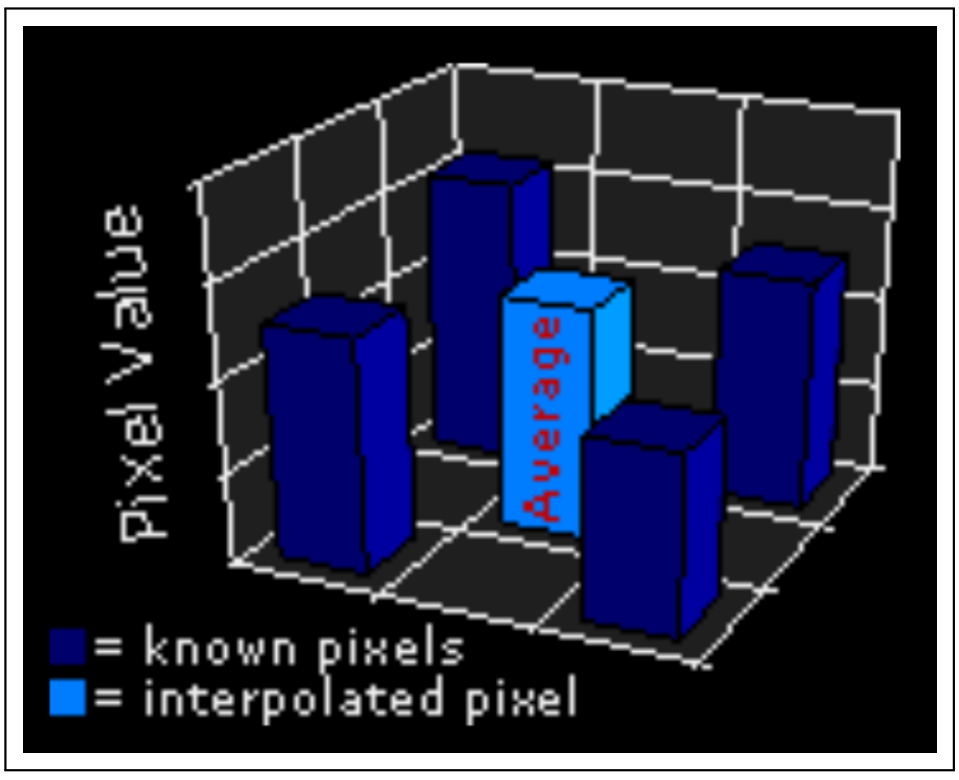
Reduce size/processing time

Image processing *tasks*

- Resize to create a uniform feature set

  Down/Upsampling – *not cropping* – bi-linear interpolation





Reduce size/processing time

Resolution visually checkable

Image processing *tasks*

- Denoise: remove unnecessary details to allow for & better generalization of images to image classes

Image processing *tasks* – more on denoising

- Spatial proximity *and pixel agreement (bi-lateral)*

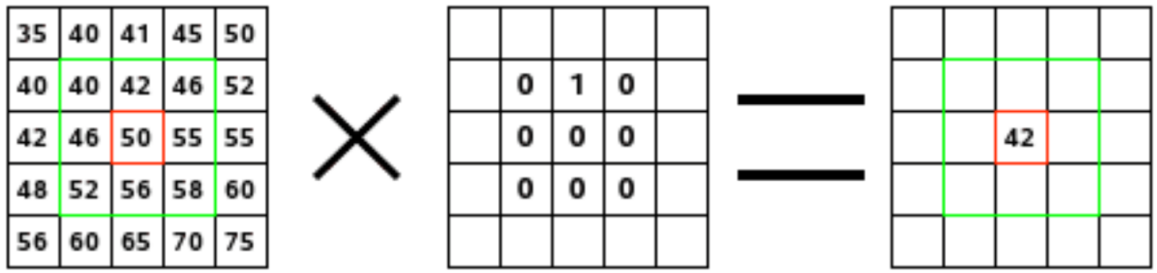
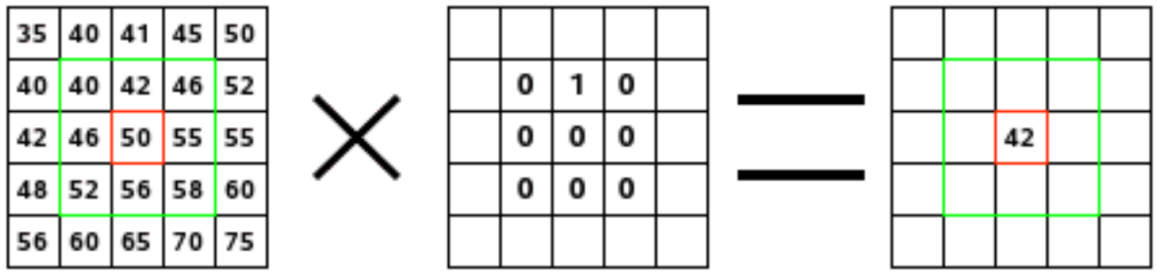


Image processing *tasks* – more on denoising

- Spatial proximity *and pixel agreement (bi-lateral)*



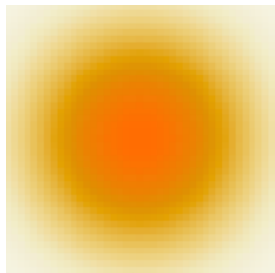
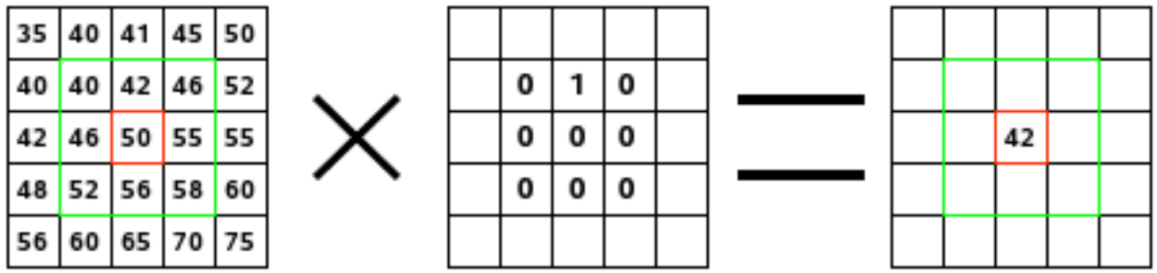


Image processing *tasks* – more on denoising

- Spatial proximity *and pixel agreement (bi-lateral)*



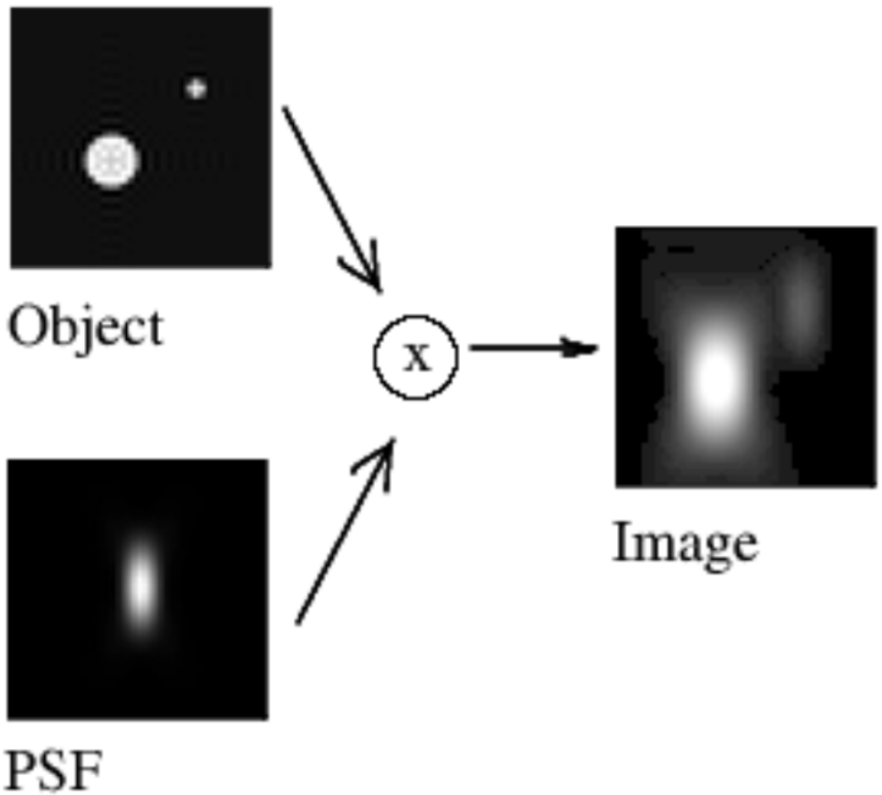
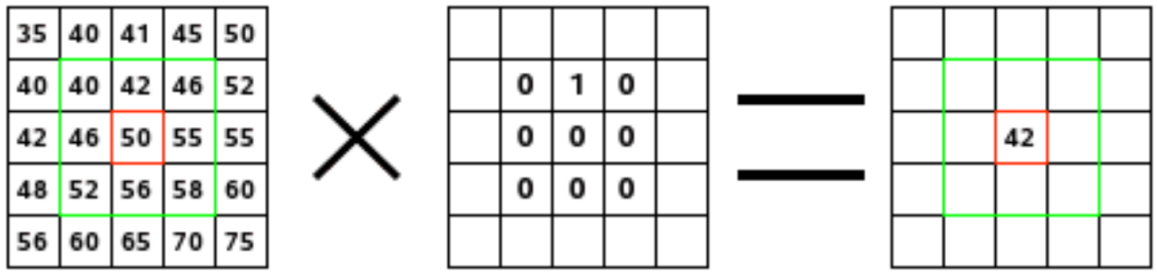


Image processing *tasks* – more on denoising

- Spatial proximity *and pixel agreement (bi-lateral)*



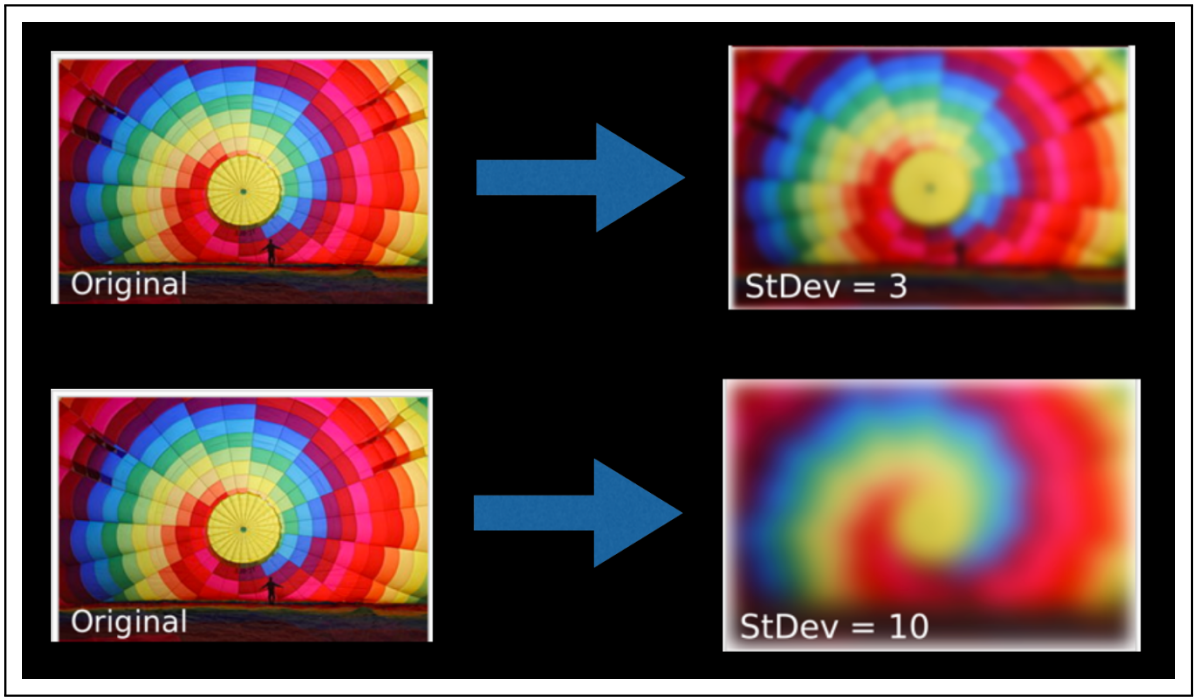


Image processing *tasks* – more on denoising

- Variation Minimization: Variation Fidelity tradeoff

$$x_i \to y_i : \min_{\boldsymbol{y}} \quad \frac{1}{2} \underbrace{\sum (x_i - y_i)^2}_{\textit{Fidelity}} + \lambda \underbrace{\sum |y_{i+1} - y_i|}_{\textit{Variation}}$$

Image processing *tasks* – more on denoising

- Variation Minimization: Variation Fidelity tradeoff

$$x_i \to y_i : \min_{\boldsymbol{y}} \quad \frac{1}{2} \underbrace{\sum (x_i - y_i)^2}_{\textit{Fidelity}} + \lambda \underbrace{\sum |y_{i+1} - y_i|}_{\textit{Variation}}$$

[What is a "hard" image for this procedure?]

Image processing *tasks*

- Grayscale (luminescence): Tensor $\rightarrow$ Matrix

  Normalize $0.2125R + 0.7154G + 0.0721B$

Image processing *tasks*

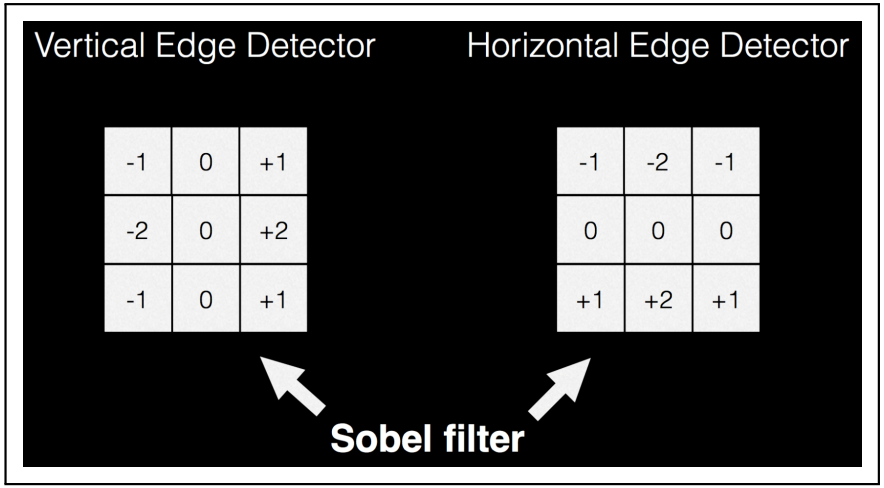
- Grayscale (luminescence): Tensor $\rightarrow$ Matrix

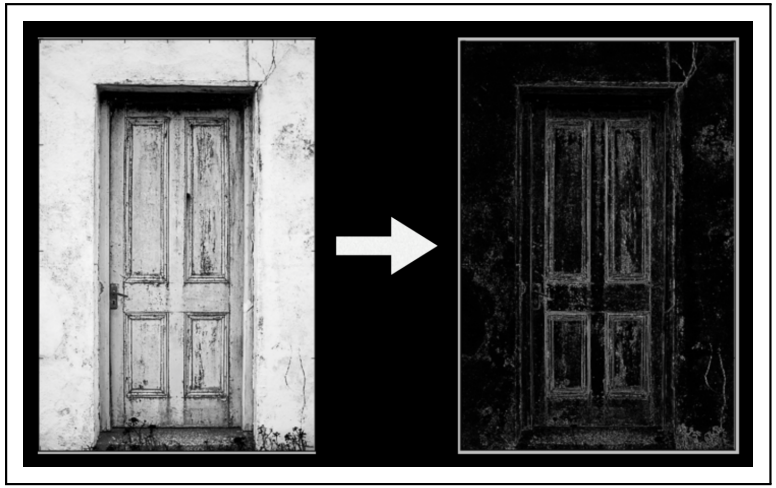
$$\text{Normalize } 0.2125R + 0.7154G + 0.0721B$$

- Edge detection (grayscale): detects large pixel transitions

Image processing *tasks* – more on edge-detection

It may be easier to identify objects if we just consider the edges





We may be able to use intensity gradients for feature ascertainment

Image processing *tasks*

- Image processing libraries
  - Scikit-image (skimage)
  - OpenCV (and dependencies)
  - Python Imaging Library
  - Pillow (a fork of the above)
  - etc.

Image *Color Structure*

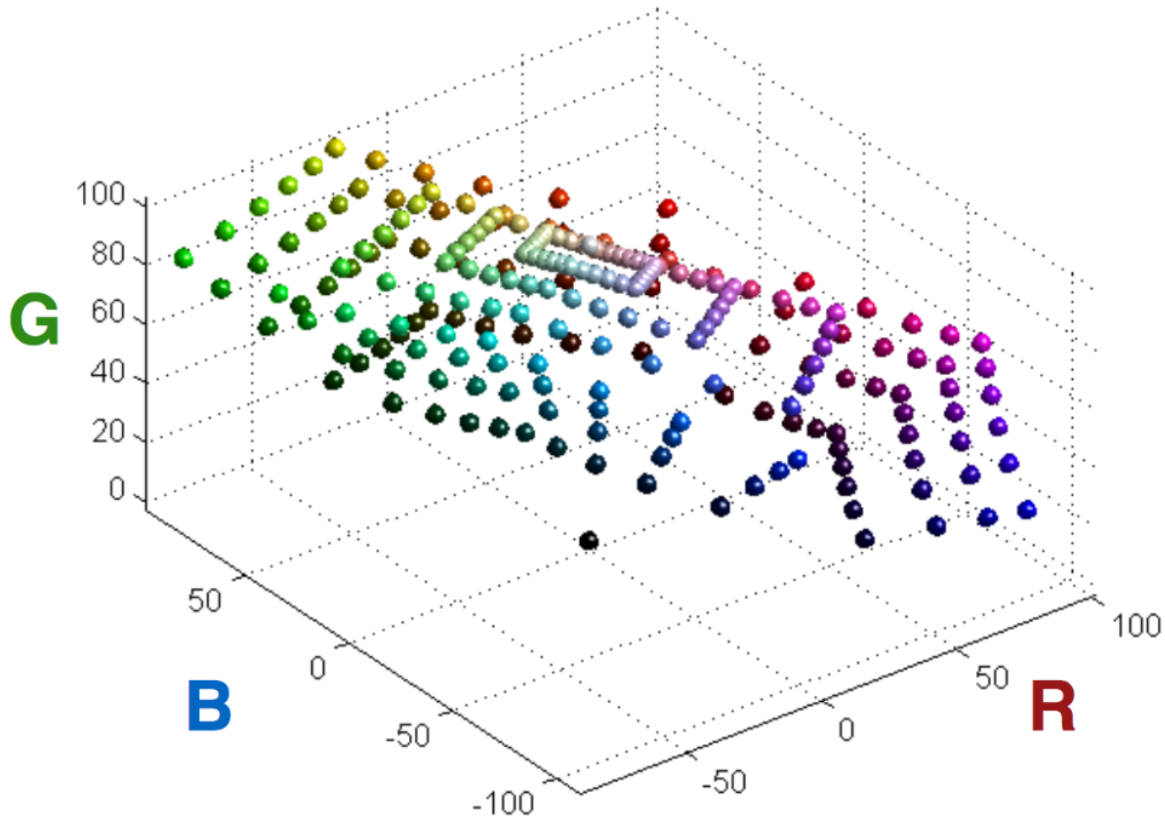


Image *Color Structure*

Image *Featurization*