Medical Imaging - Measuring Features

Veronika Cheplygina

Medical imaging project - timeline

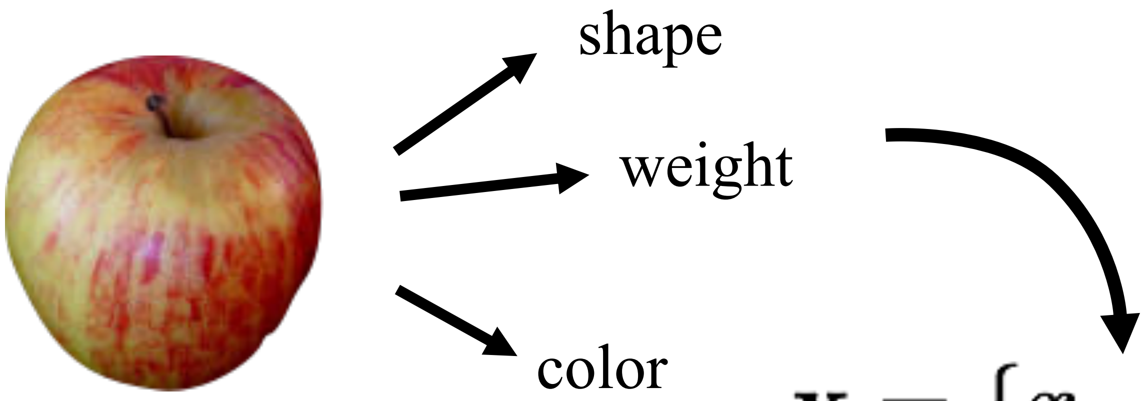
- Tue 6 – Introduction
- Fri 9 – **Extract features from images**
- Tue 13 – Classify image to predict diagnosis
- Fri 16 – Evaluate results
- Tue 20 – TBA / Recent research

Today

- Feature representation, high vs low-level features
- Features for skin cancer
  - Group activity
- General purpose features
- Notebook

- Objects/samples/images → Feature representation (e.g. CSV) / dataset

- Encode each sample by *p* features, represent it in a **feature space**

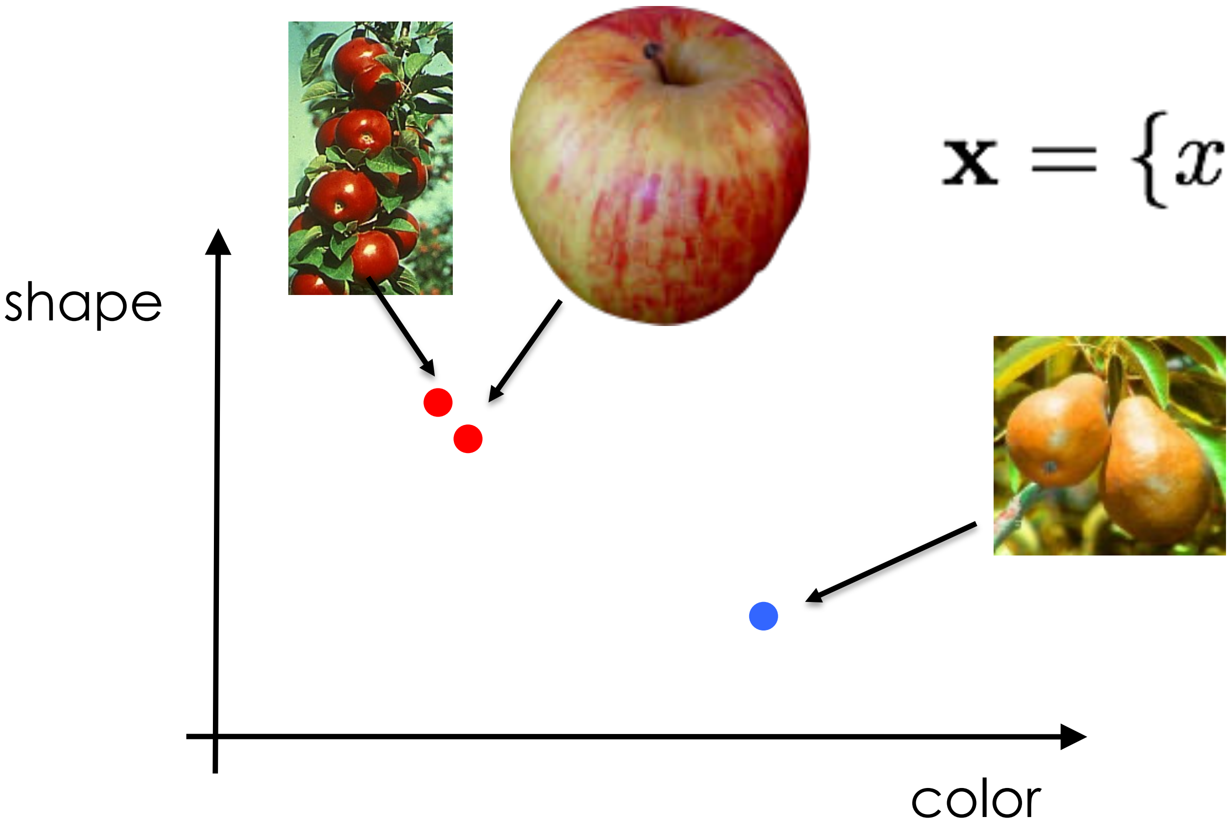


$$\mathbf{x} = \{x_1, x_2, ..., x_p\}$$

- Feature space for fruit

$$\mathbb{R}^p$$

$$\mathbf{x} = \{x_1, x_2, ..., x_p\}$$



dataset

feature

label

object/sample

Object	Weight	Color	Label
Apple 1	25	36	1
Apple 2	20	34	1
Apple 3	35	40	1
Pear 1	35	55	2
Pear 2	37	35	2
Pear 3	40	57	2
Pear 4	36	41	2

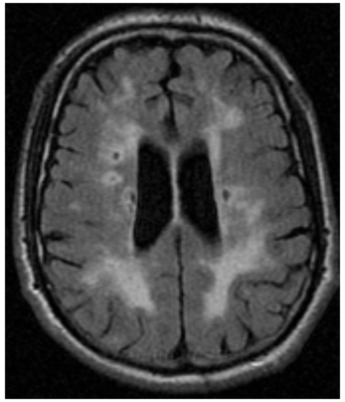
An image can be associated with different representations!

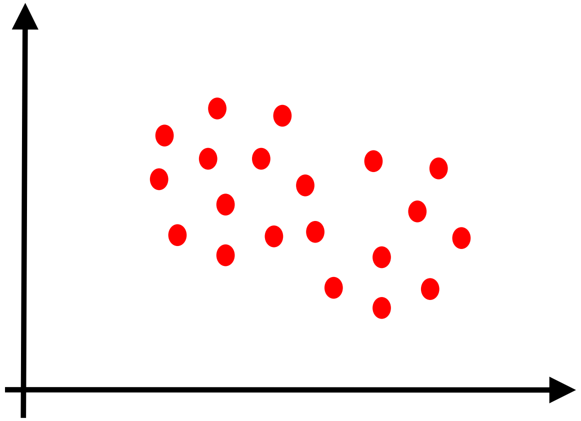
Segmentation

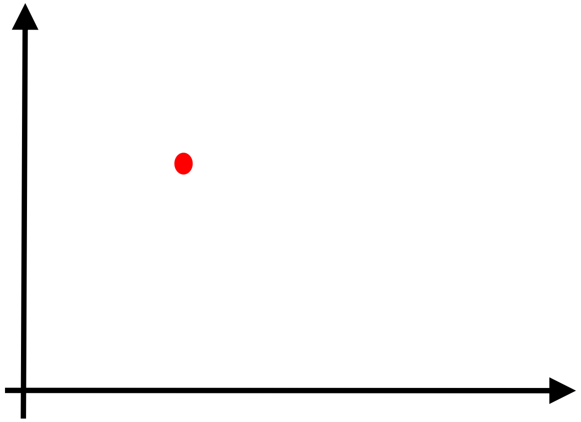
Represent each pixel

Diagnosis

Represent each image

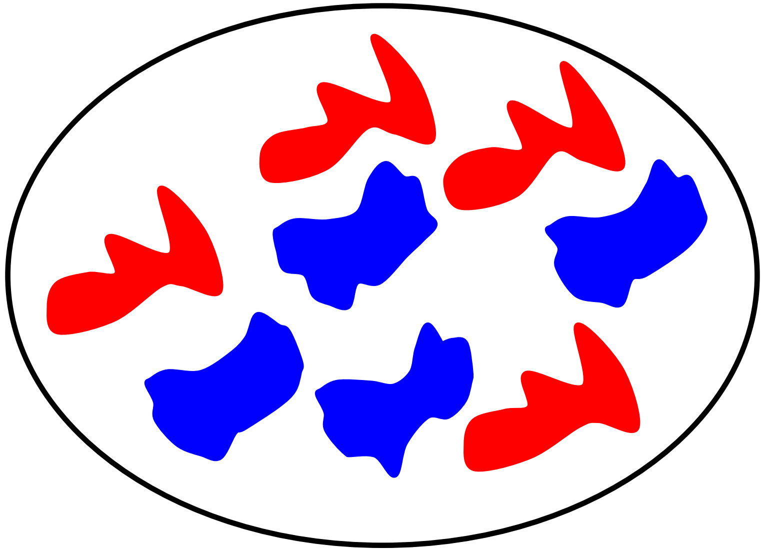




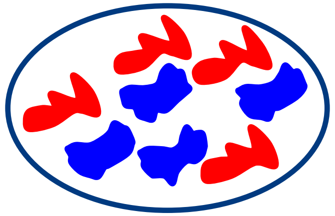


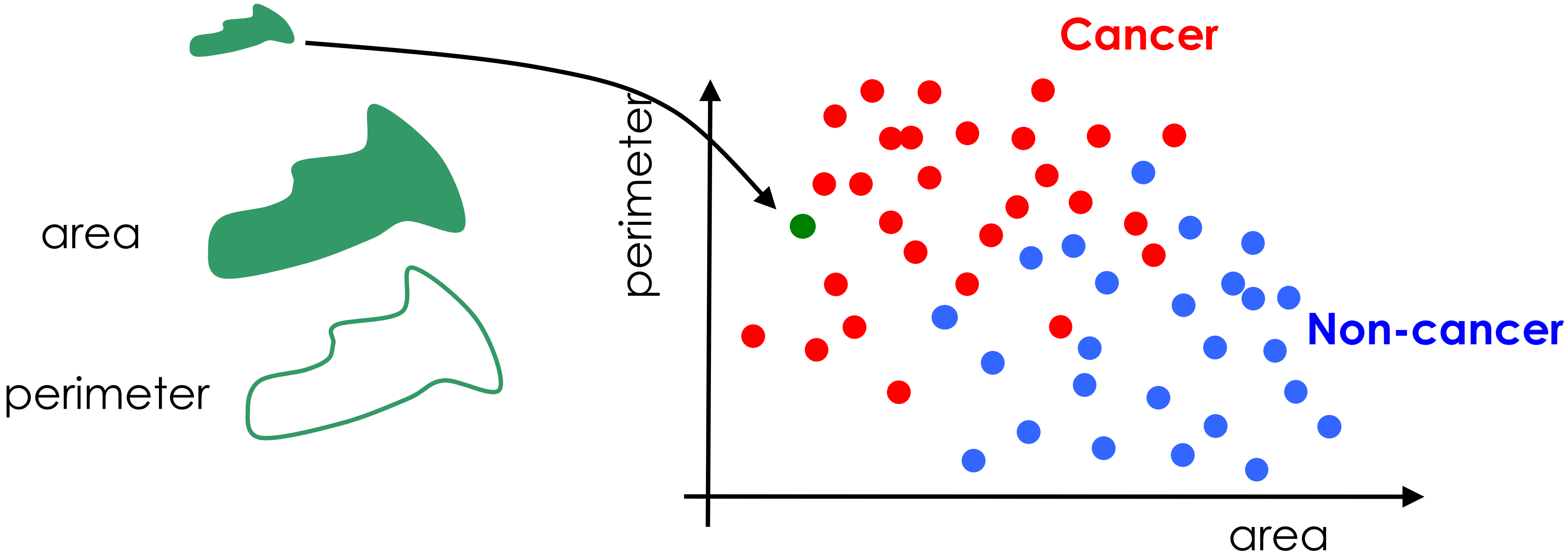
Example: red spots = cancer and blue spots = non-cancer

Blue spots have more smooth shapes than the red → how to measure this?



Measuring shape with area and perimeter



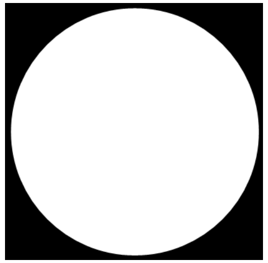


Measuring shape with area and perimeter

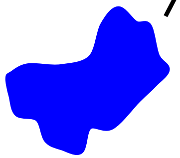
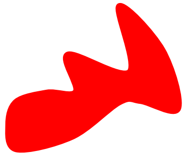
- The relationship of area and perimeter tells us about the shape
- We can combine two features into a single number
- Compactness $c = \frac{l^2}{4\pi A}$ (*l* = length i.e. perimeter, *A* = area)

Measuring shape with area and perimeter

- Compactness $c = \frac{l^2}{4\pi A}$

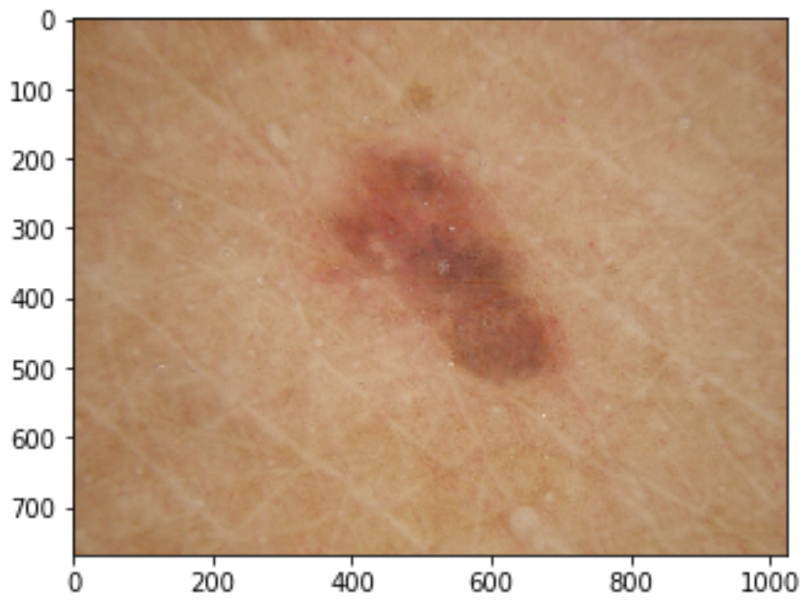


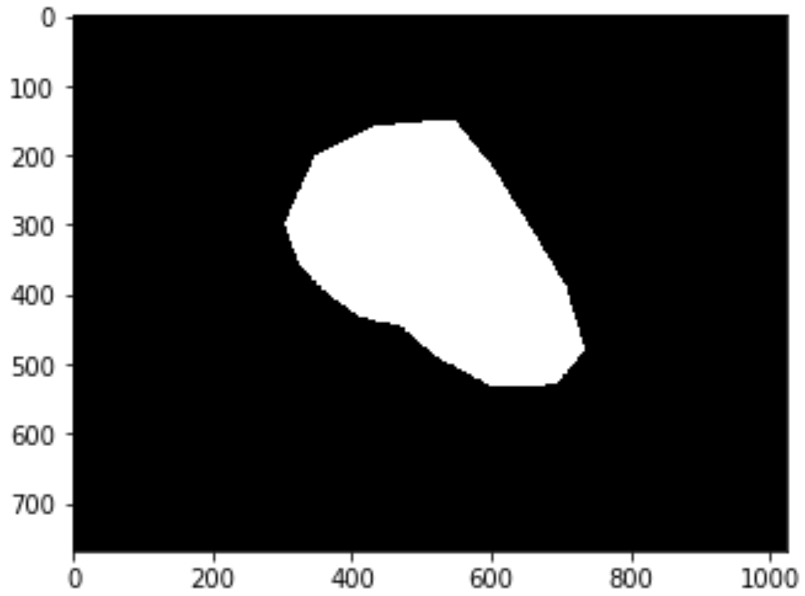

1.3 7.2 79.1 743.2

Measuring shape with area and perimeter - Steps

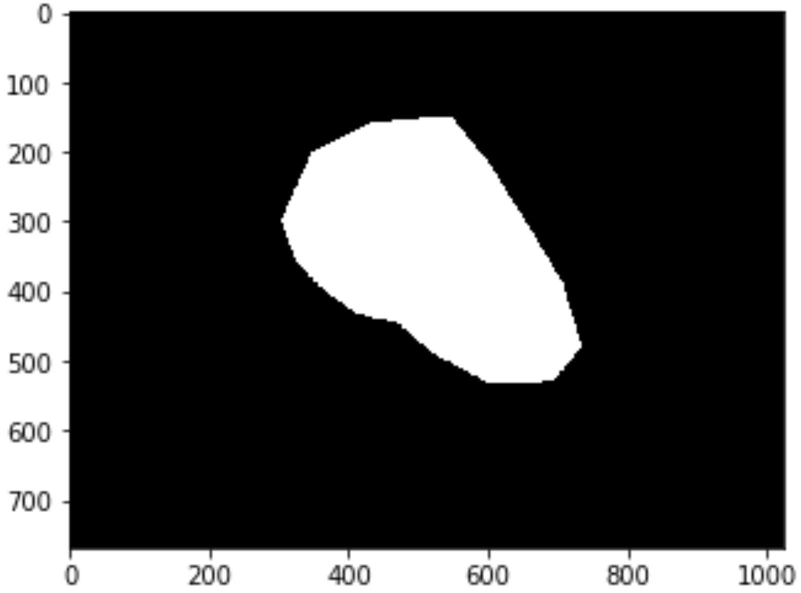
- Need a binary image or **mask** with 1’s inside and 0’s outside the shape
- Area = sum of all pixel values in the mask

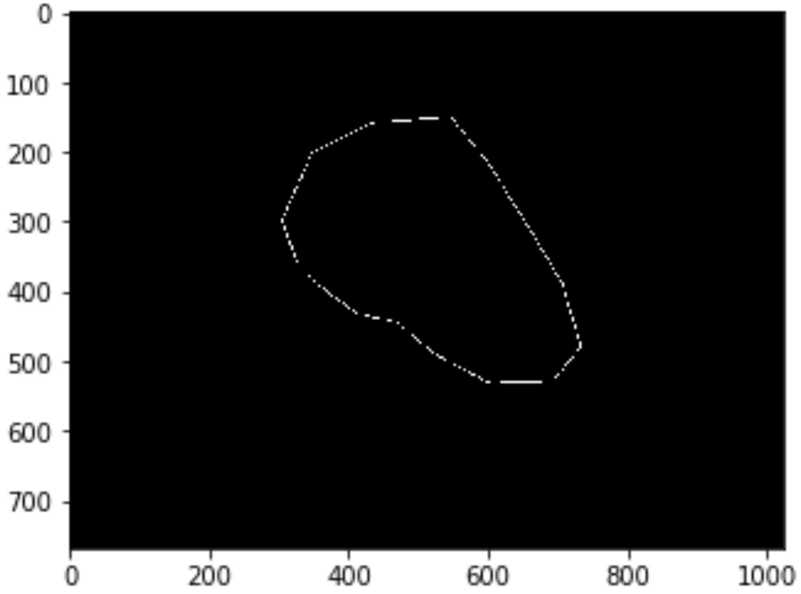




Measuring shape with area and perimeter - Steps

- Perimeter = sum of pixels on the border
  - Resize the image by a few pixels
  - Subtract the smaller image from the larger image
  - Sum the pixel values

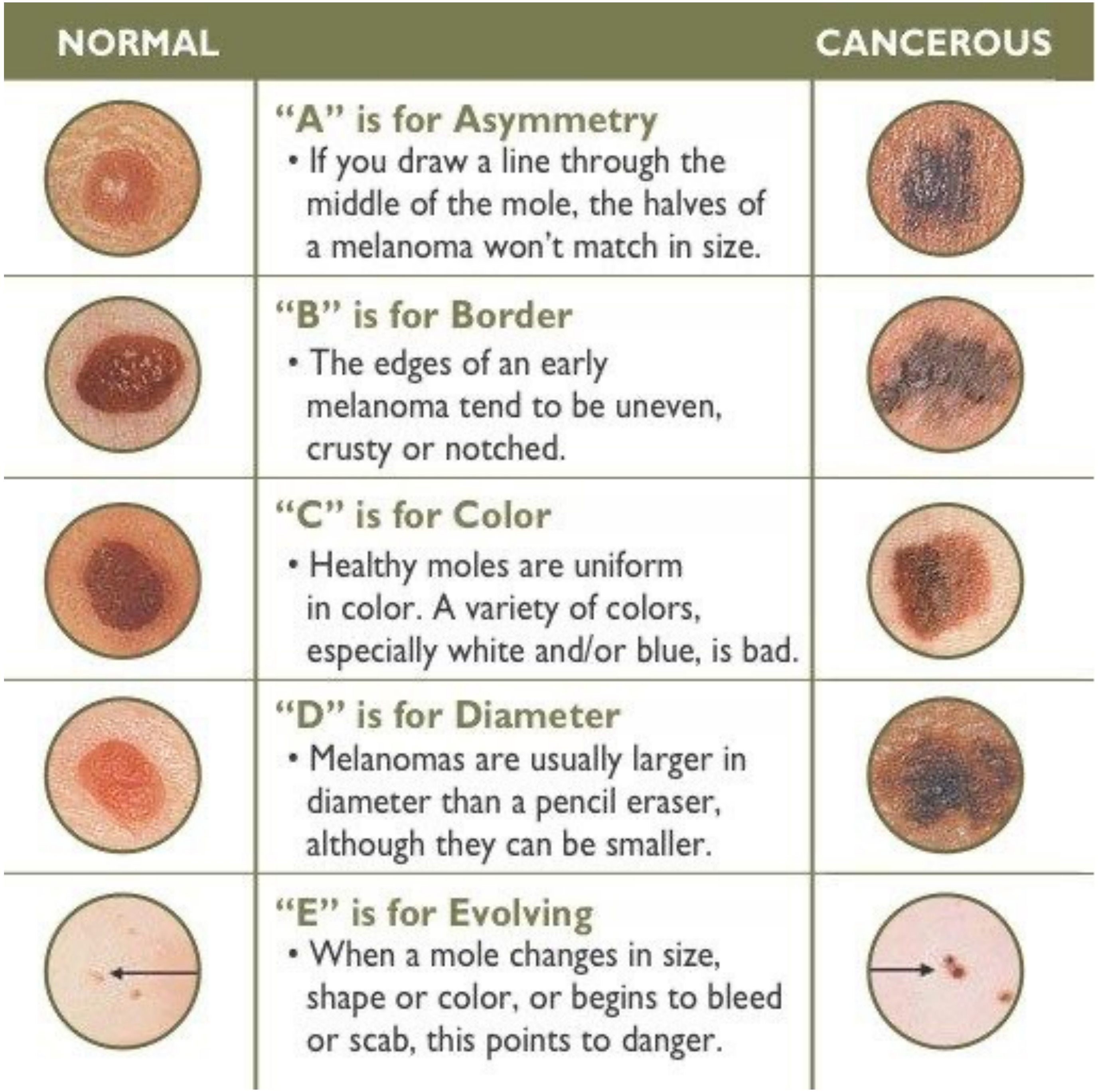




Features for skin cancer

- Experts use “ABCDE” features to recognize melanoma (what about other types?)

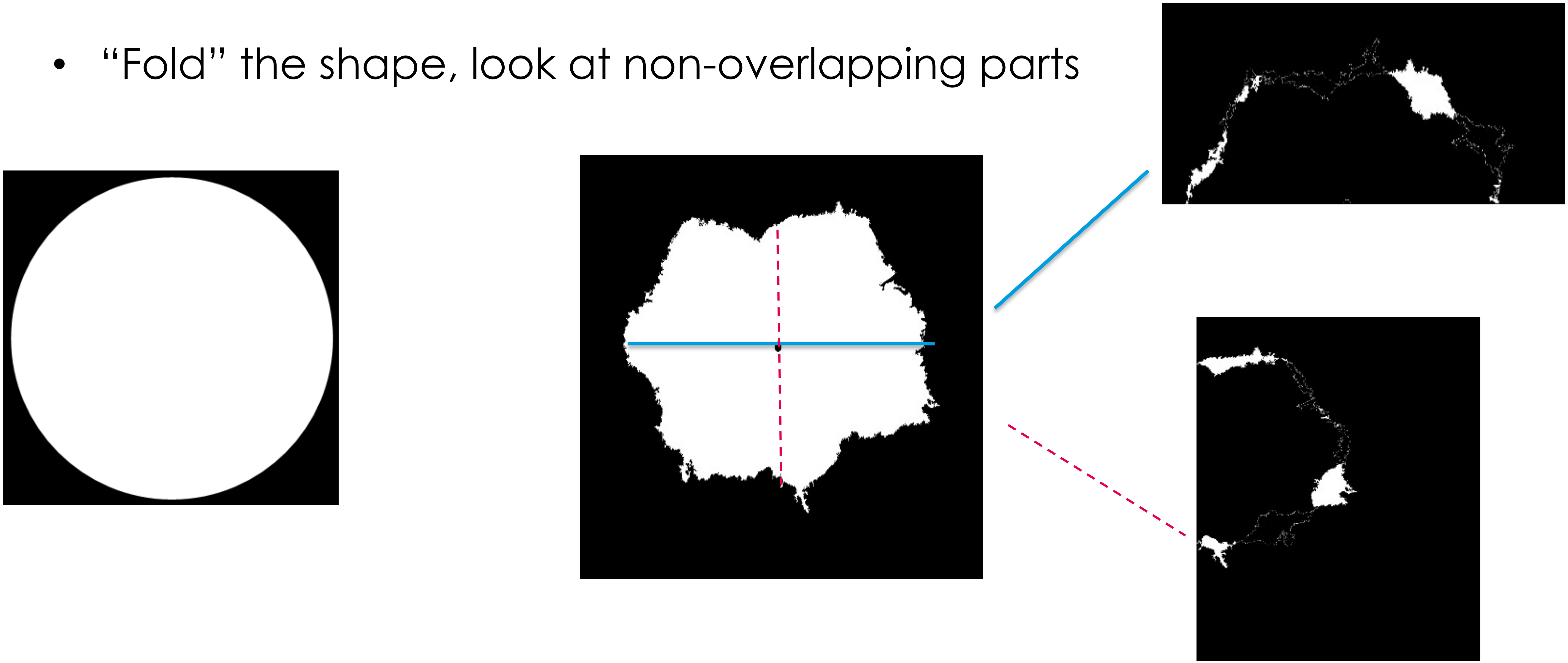
- A – Asymmetry
- B - Border
- C – Color
- D – Diameter
- E - Evolving



https://www.pinterest.com/grotonderm/

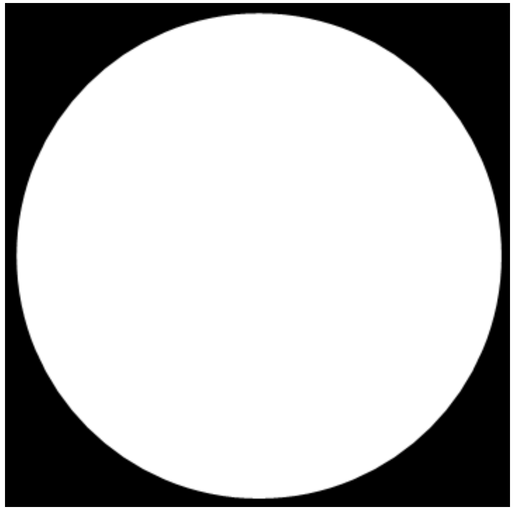
Features for skin cancer - Asymmetry

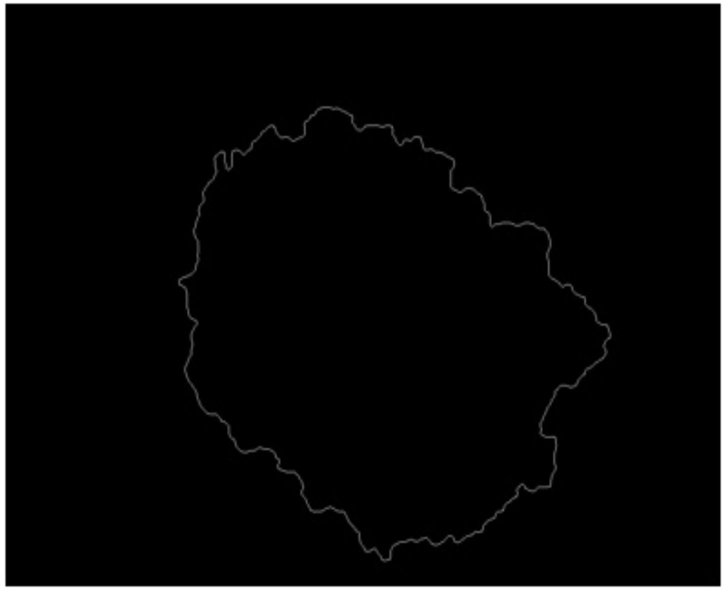
- “Fold” the shape, look at non-overlapping parts

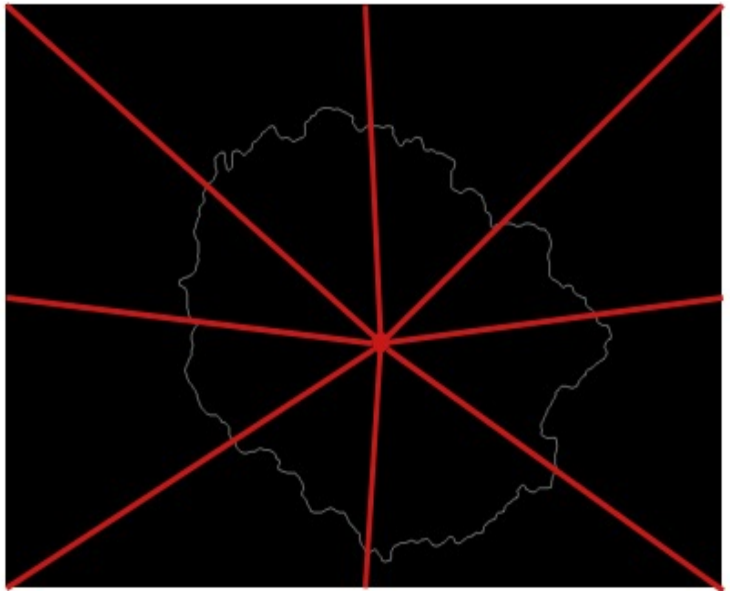


Features for skin cancer - Border

- Variation of radius in sections of shape







Activity 2 - Measuring color

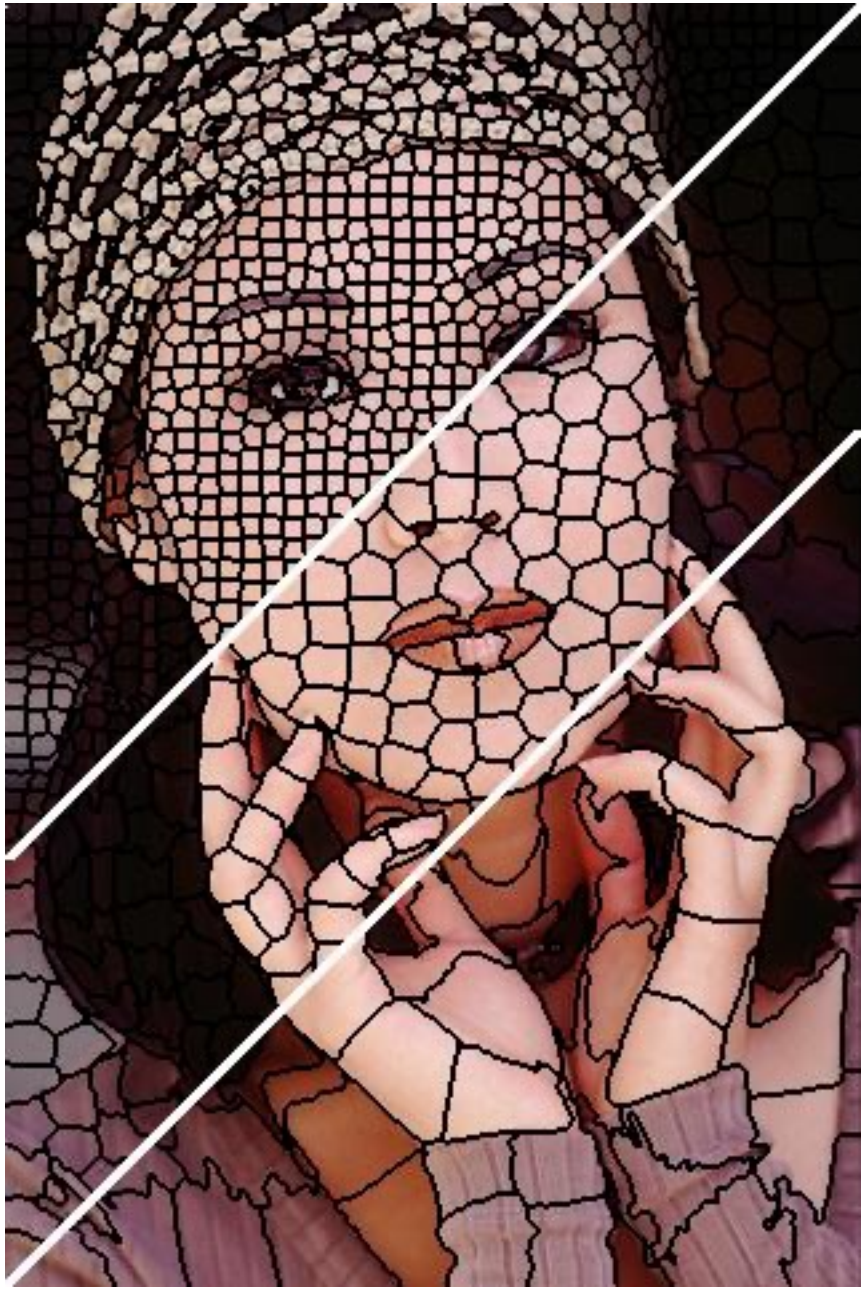
You can add this template to the Google doc from last time:

https://tinyurl.com/2f4u3uew

Measuring color

- Variation in RGB not always intuitive
- How many pixels are enough for a color to be "there"?
- "Average" color?

More robust: look at areas with continuous color – group pixels into "superpixels" and find

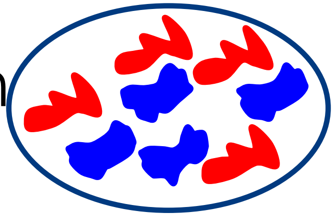


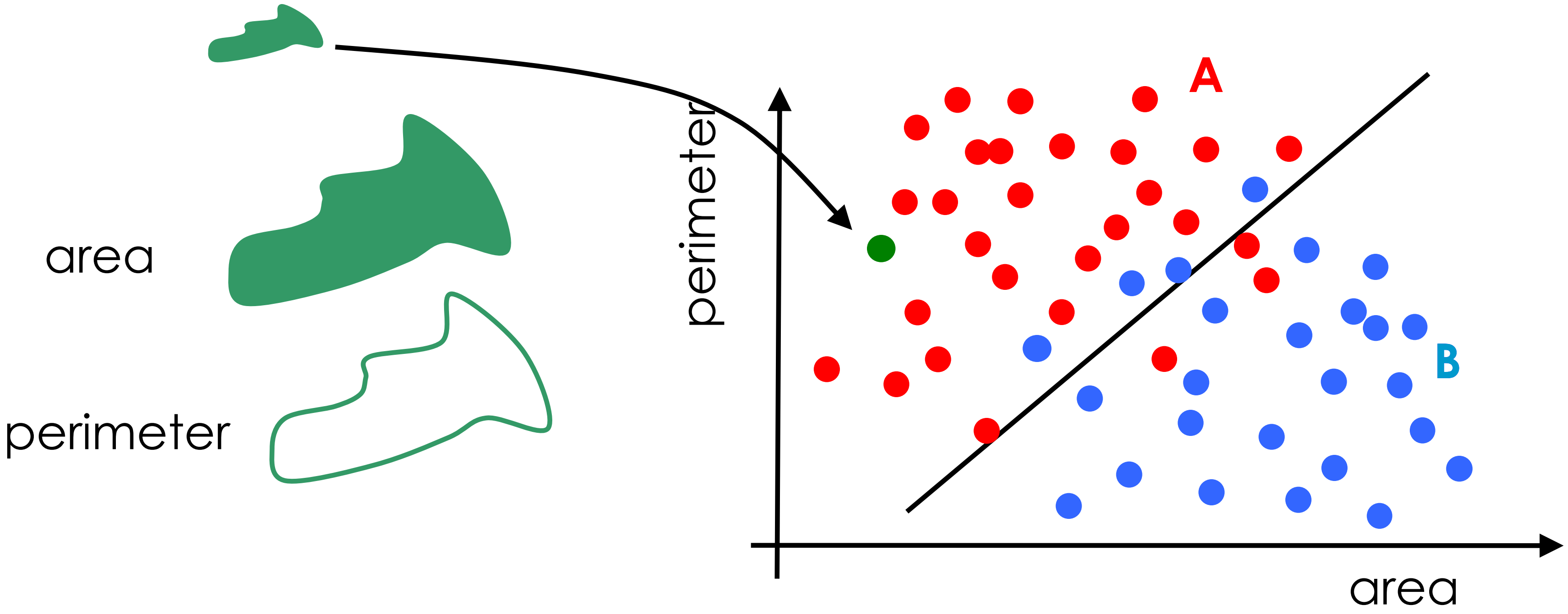
https://www.epfl.ch/labs/ivrl/research/slic-superpixels/

https://scikit-image.org/docs/dev/api/skimage.segmentation.html

High-level vs low-level features

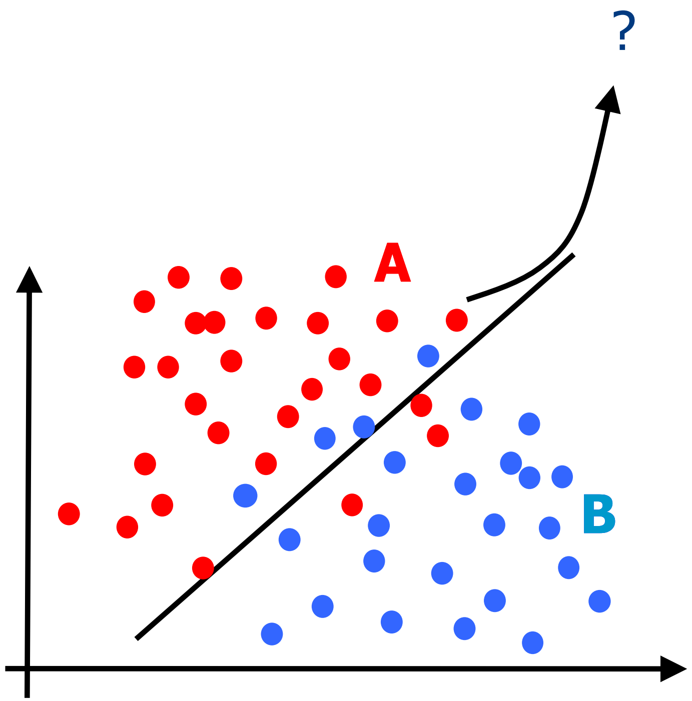
Knowledge-driven, high-level feature representation
a.k.a. “handcrafted”





Representation removes information

- Is it possible to create two different images, for which representation is identical?
- Given an area and a perimeter, can we reconstruct a unique shape?
- Representation removes information!



High-level features: advantages

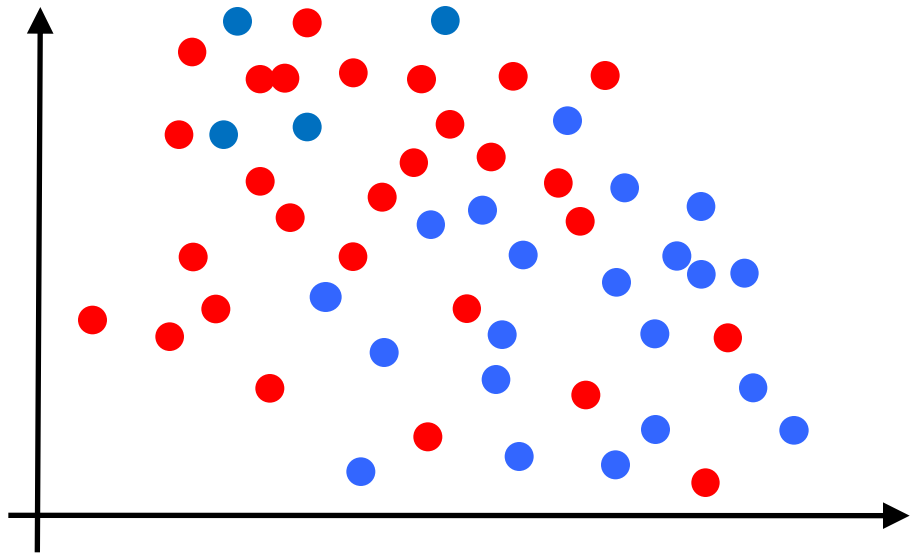
- Condense lots of information into few “high-level” features
- If features are informative, only a few objects/samples are needed

High-level features: advantages

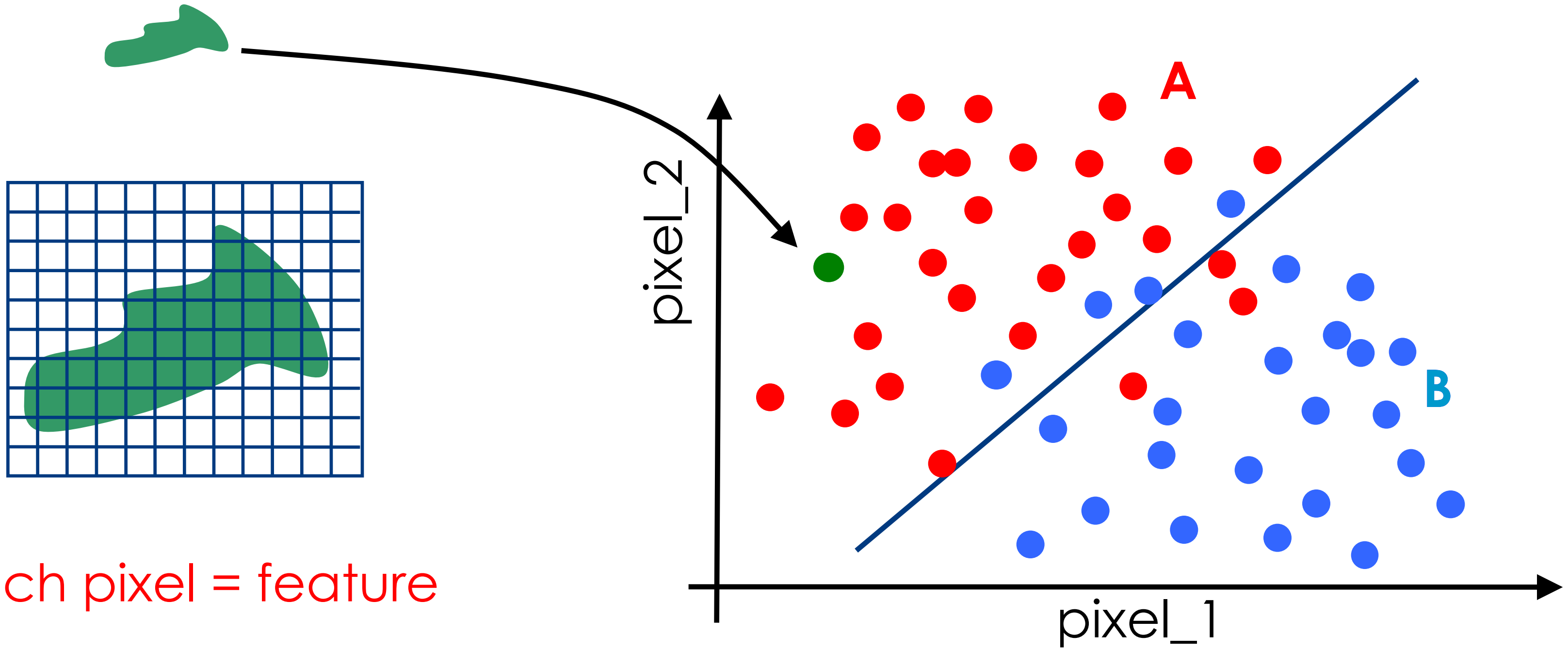
- Remove irrelevant variations in the data
- “Area” and “perimeter” are invariant to rotation. Rotation is not important in this task

High-level features: disadvantages

- Prior knowledge is difficult to encode, often leads to <u>class overlap</u>



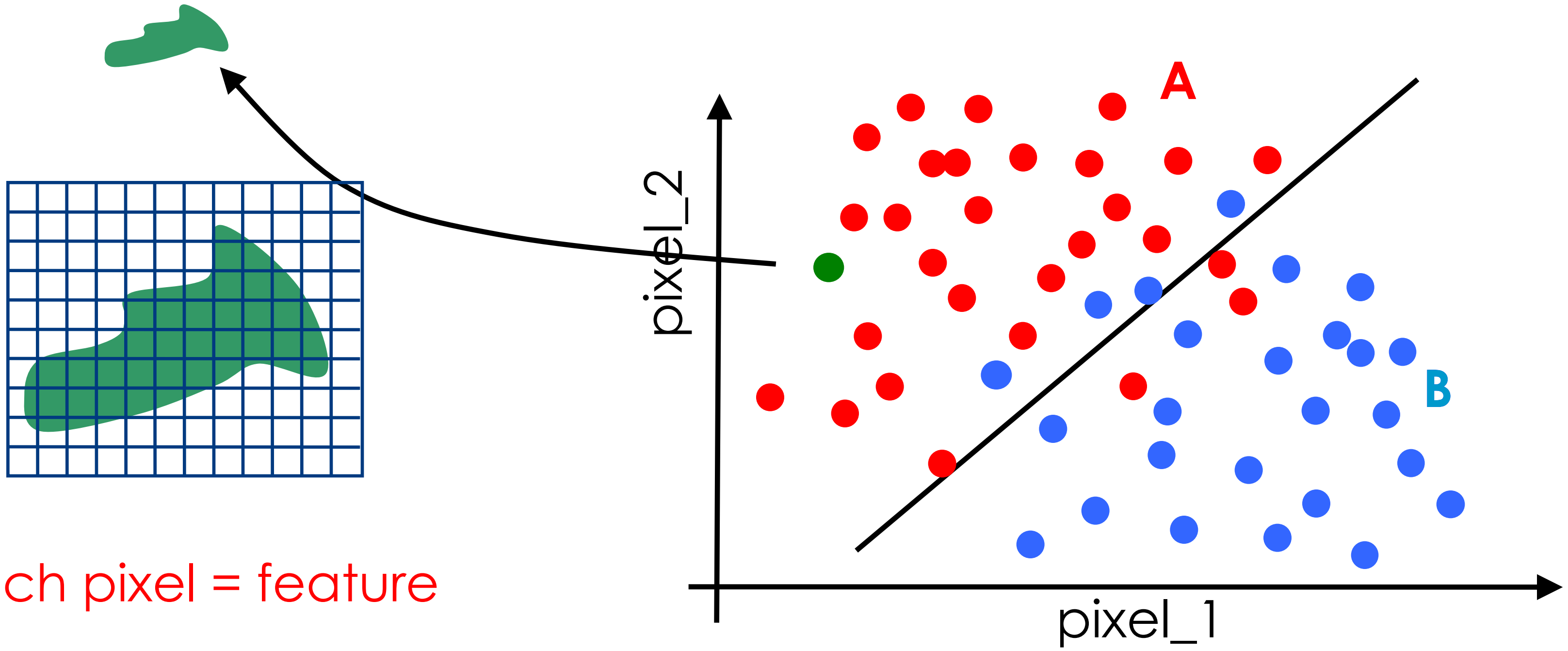
Data-driven / low-level feature representation



Each pixel = feature

Slide David Tax

(Less) information is removed: given a pixel representation, we can (almost) reconstruct the shape

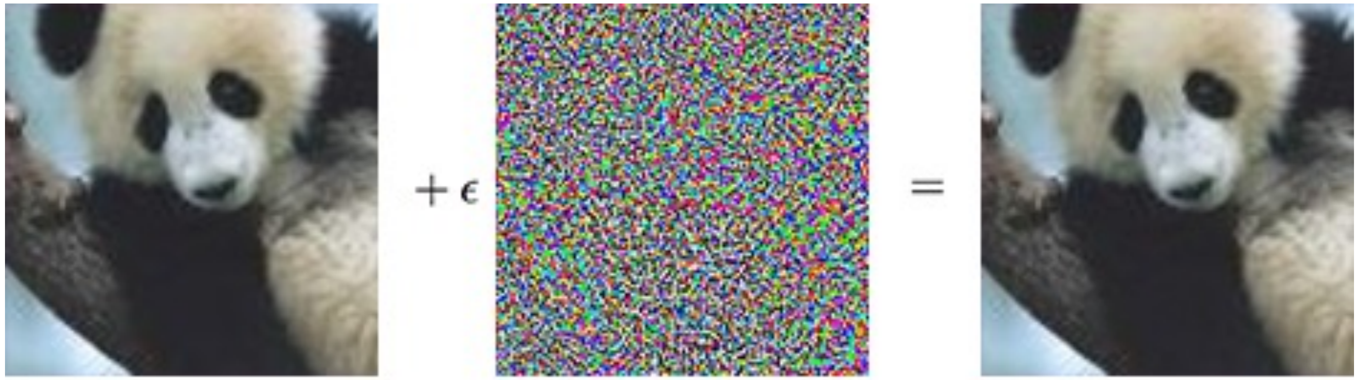


Each pixel = feature

Slide David Tax

- High-dimensional feature representation
- More objects/samples are needed to figure out which features are important
- May not be invariant to irrelevant variation (such as rotation)

- Disadvantages: not invariant to irrelevant variation in data – can also lead to class overlap



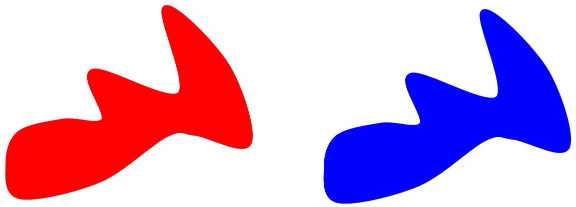
- [Same image, but different representations and classes according to network]

Source: https://blog.openai.com/adversarial-example-research/

Reasons for class overlap

- Features remove relevant information
  - Already happens during acquisition!

- Features capture noise (anything irrelevant to the label)
  - Natural variation within each class
  - Noise introduced during image acquisition



- Label noise

There is a trade-off / choice depends on application

Object

Many low-level features

Many examples

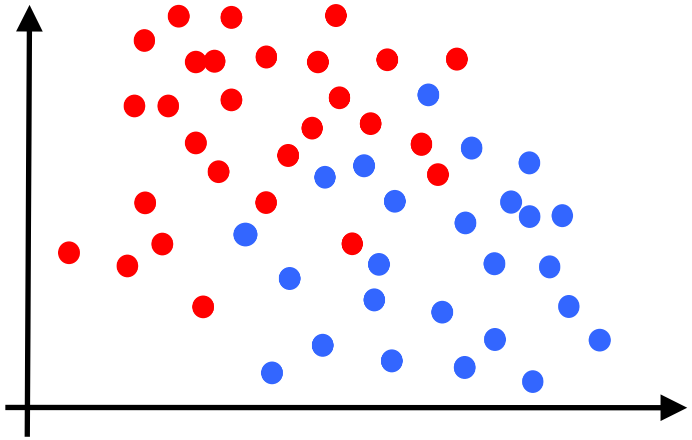
Few high-level features

Few examples

Label

What is a good feature?

- Variation of values for different images
  - "Value at location (0,0) in mask image" will probably always be zero
- Not too correlated with other features
  - "Value at (x,y)" and "value at (x+1,y)" will be similar

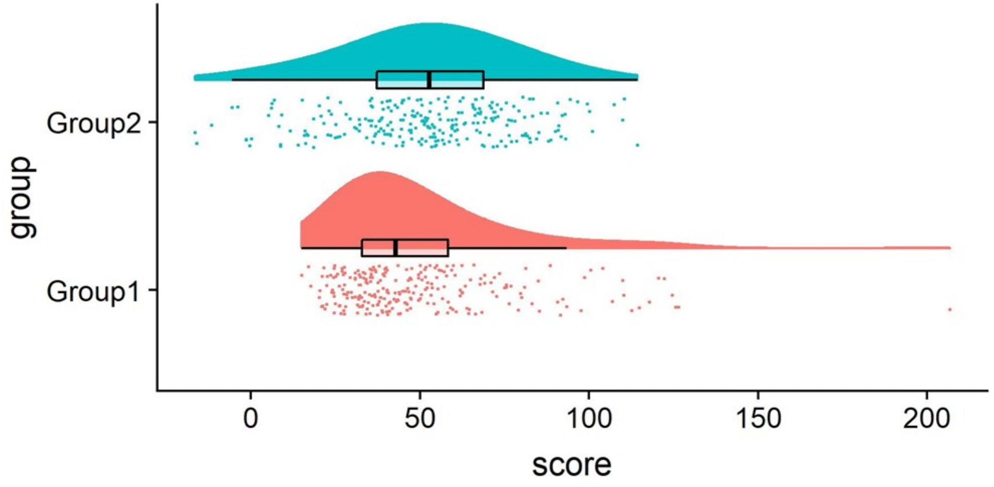


What is a good feature?

- The feature provides (some) information about the class of the image

- Combining features can help to separate the class better

- Use scatterplots or “rain cloud plots” to examine distributions of your features per class



https://wellcomeopenresearch.org/articles/4-63

General purpose features

General purpose features

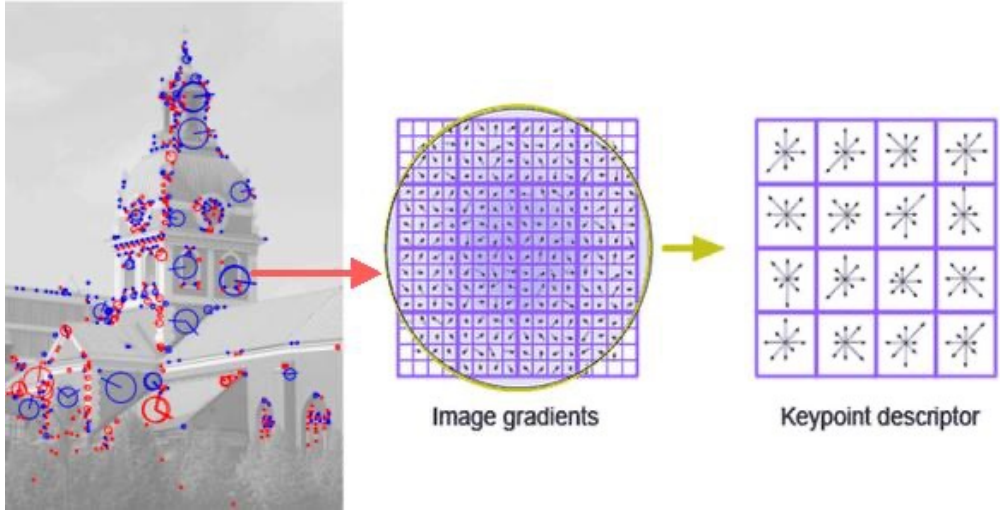
- Describe local structure (“what is where”) in an image

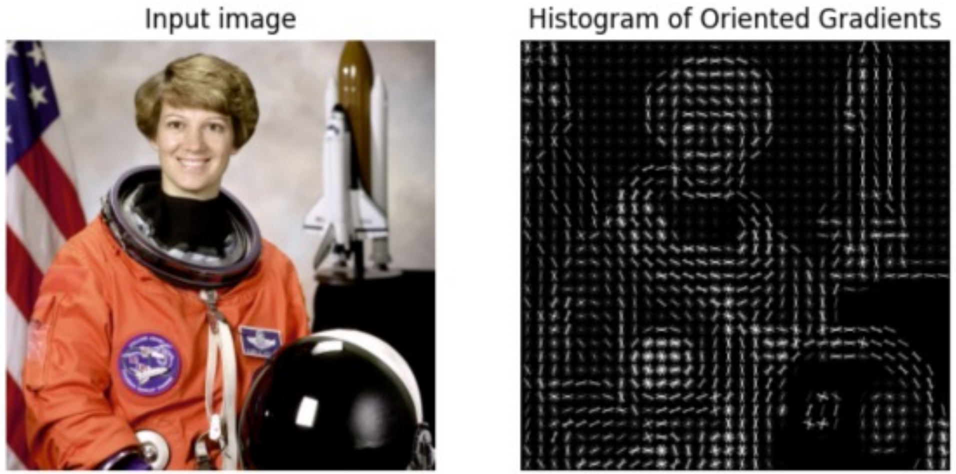
- Idea:
  - **Filter** image to reveal some kind of patterns (e.g. edges)
  - Create a histogram of the intensities in filter image
  - Do this for multiple types of filters

General purpose features

- SIFT = Scale Invariant Feature Transform
- HOG = Histogram of Gradients
- LBP = Local Binary Patterns
- (examples in skimage)

- We look at Gaussian filter banks





Filter banks – Image filtering

- Filtering an image = **convolution** of an image and another function

- Replace each pixel by a **linear combination of its neighbors**

- For equal weights, the center pixel becomes:

(10+1+1+1+1+1+1+1+1) / 9 = 2

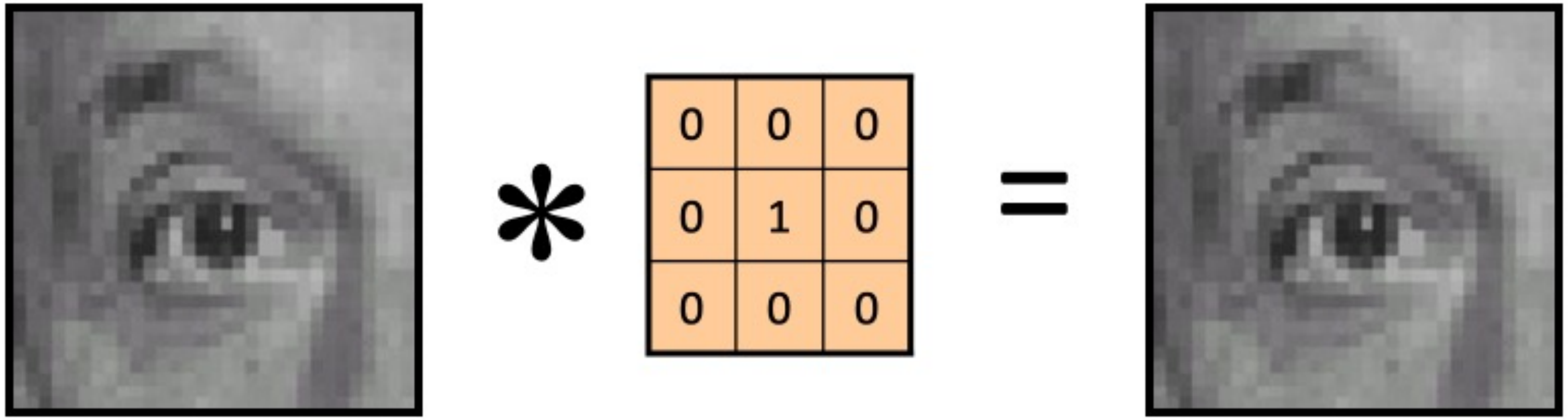
- Repeat for all pixels

1	1	1
1	10	1
1	1	1

	2	

Image filtering

- The weights can be also viewed as a matrix, often called the kernel
- Different kernels have different effects on the image
- Source: D. Lowe / N. Snavely

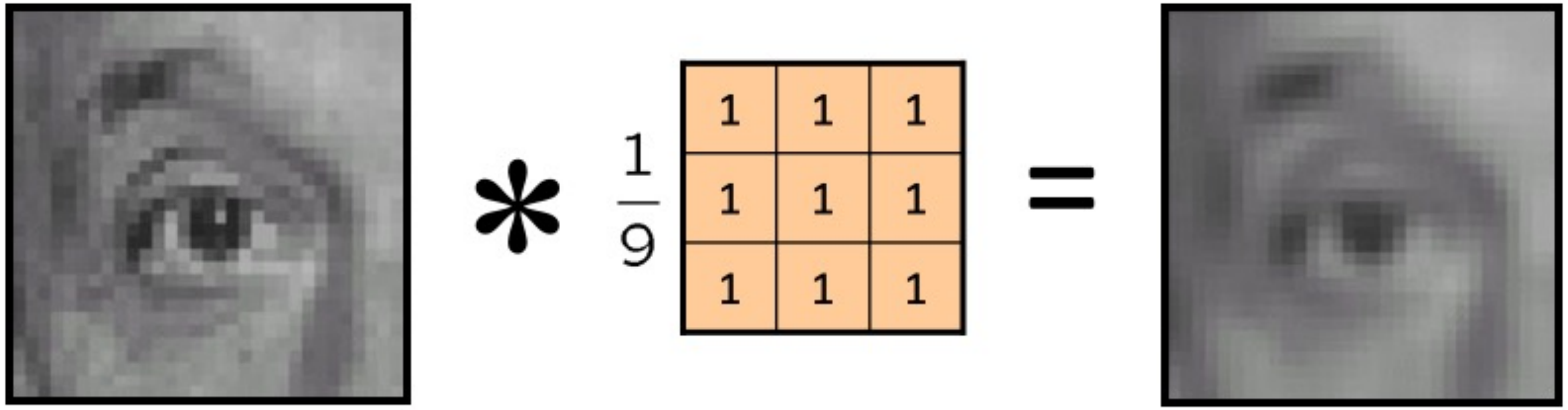


Original

Identical image

Image filtering

- Source: D. Lowe / N. Snavely



Original

Blur (with a mean filter)

Image filtering

- Source: D. Lowe / N. Snavely

Original

$$* \left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right) =$$

Image filtering

- Source: D. Lowe / N. Snavely

Original

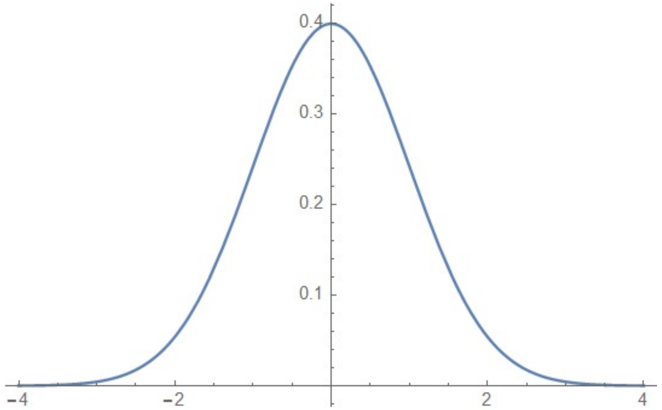
$$
* \left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right) =
$$

Sharpening filter
(accentuates edges)

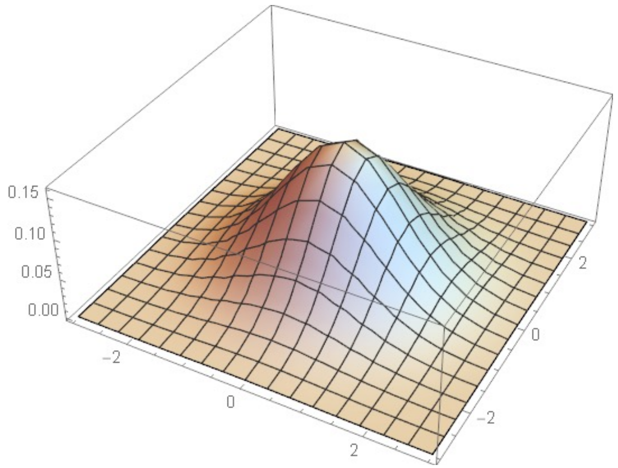
Gaussian filter banks

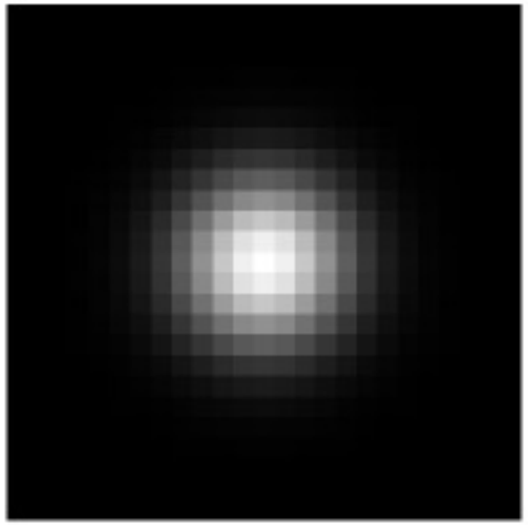
- Filtering with equal weights can create “blocky” patterns
- Instead, Gaussian kernels are suitable
- The weights are given by a (family of) Gaussian functions

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



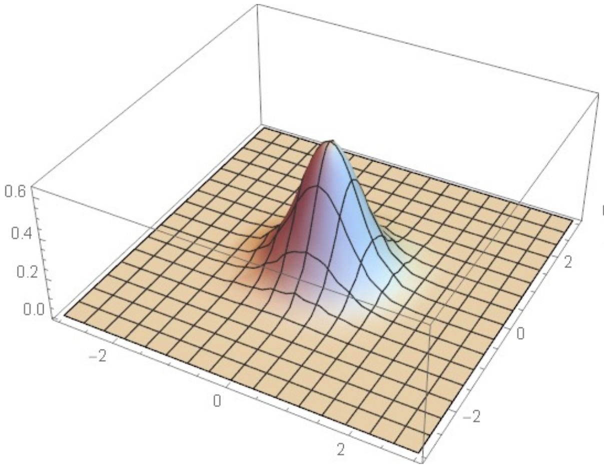
$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

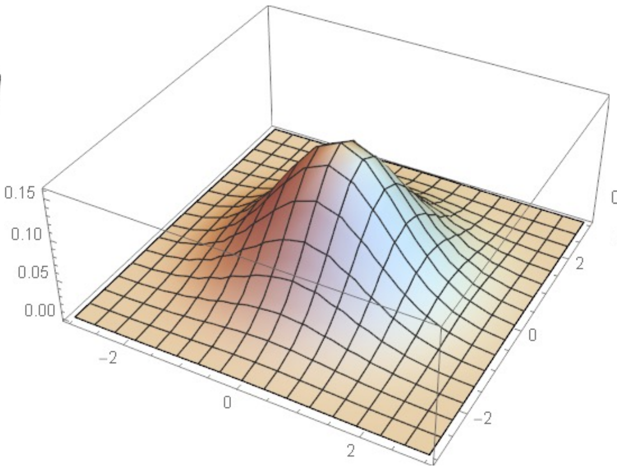


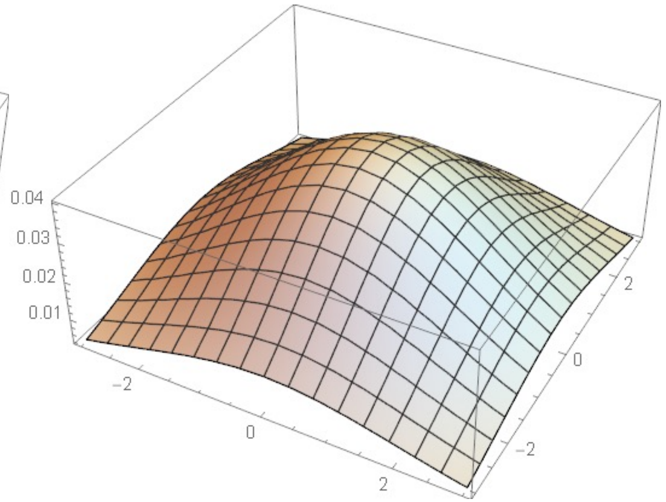


- Gaussians have a standard deviation, or scale

- $\sigma = 0.5$ $\sigma = 1$ $\sigma = 2$

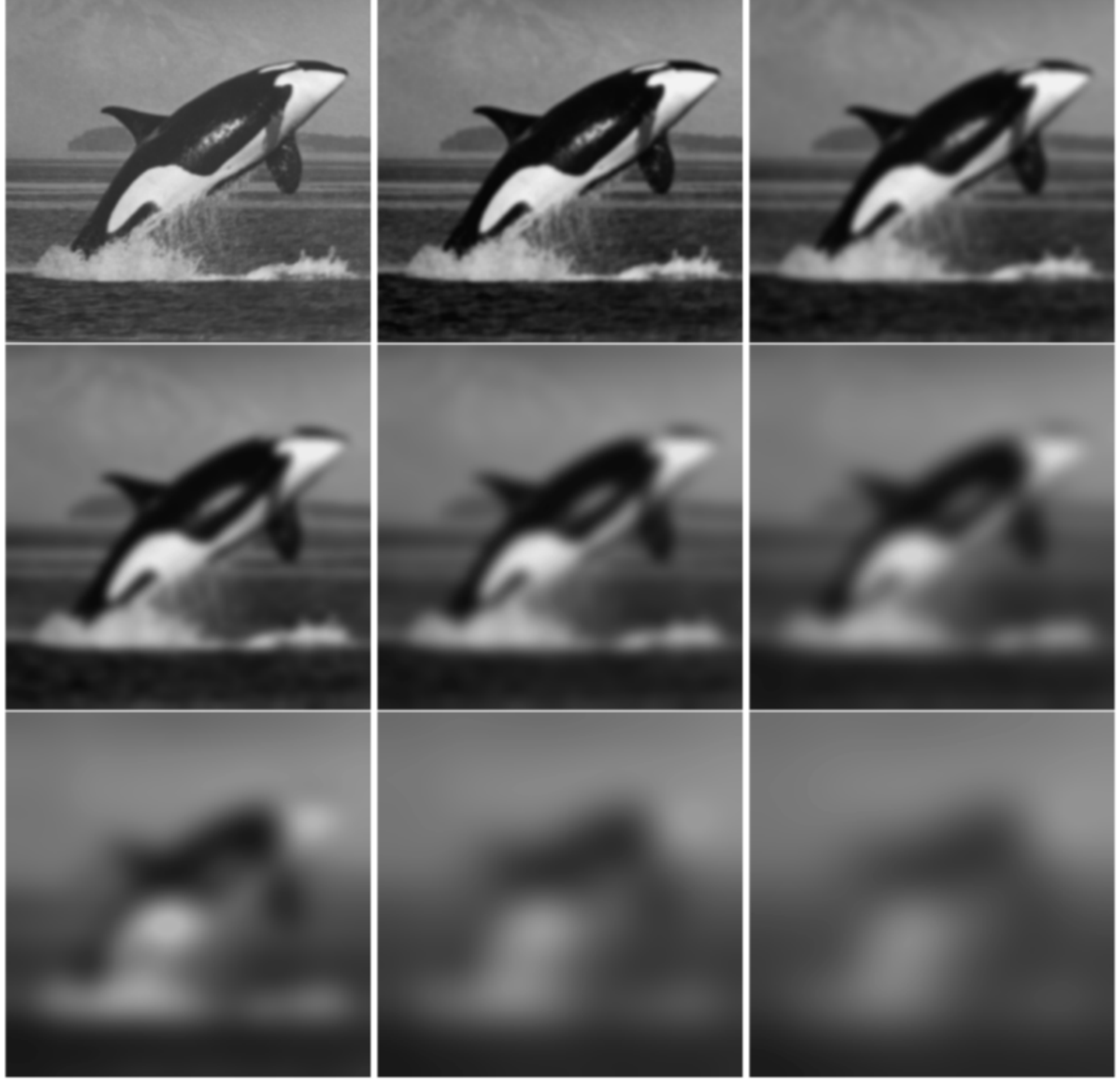




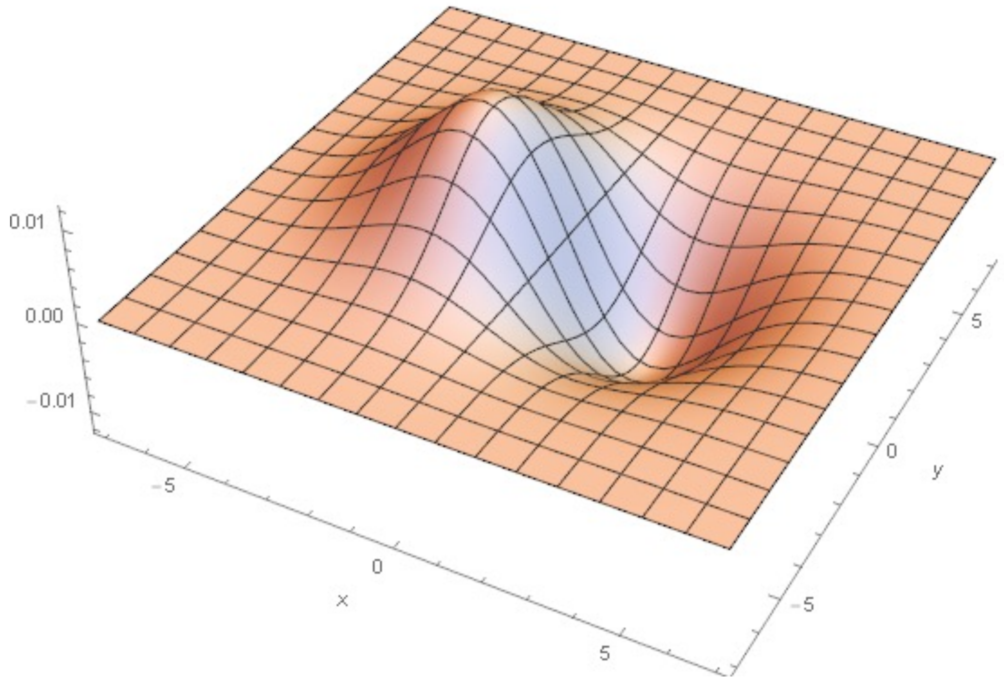


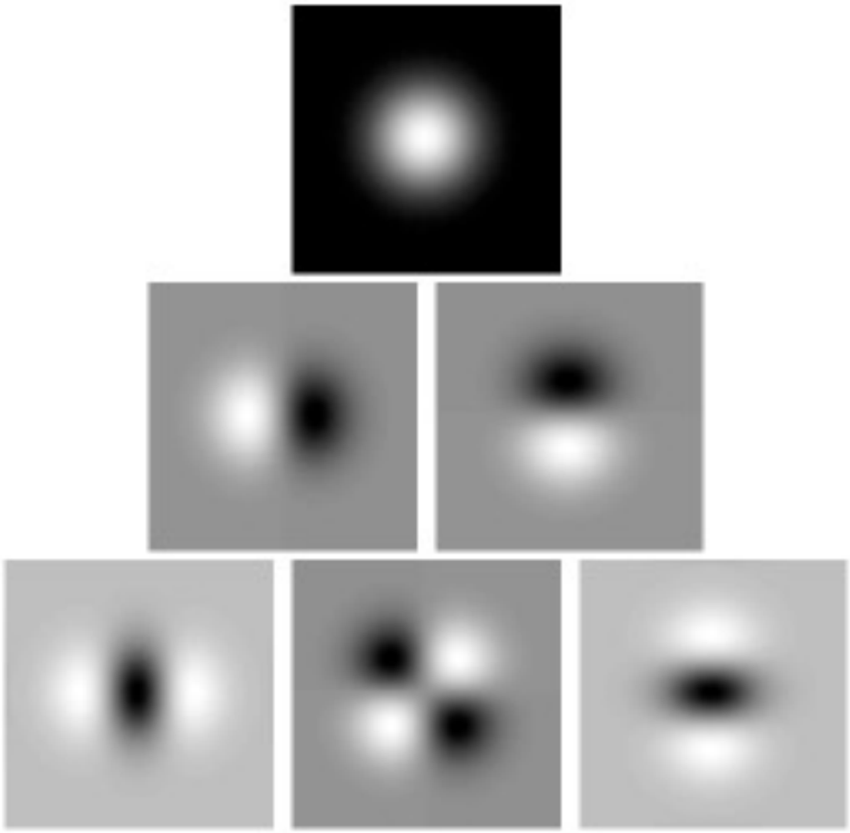
- Gaussians have a standard deviation, or scale

- Larger scale = pixels far away have higher weights = more blurring



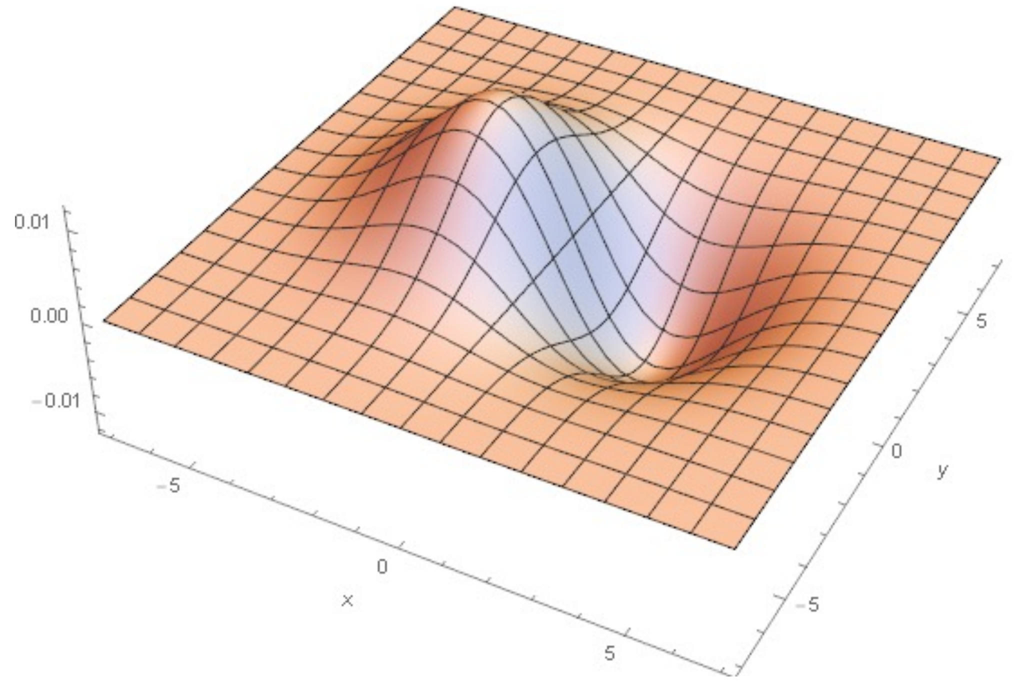
- Not only Gaussians, but also derivatives of Gaussians

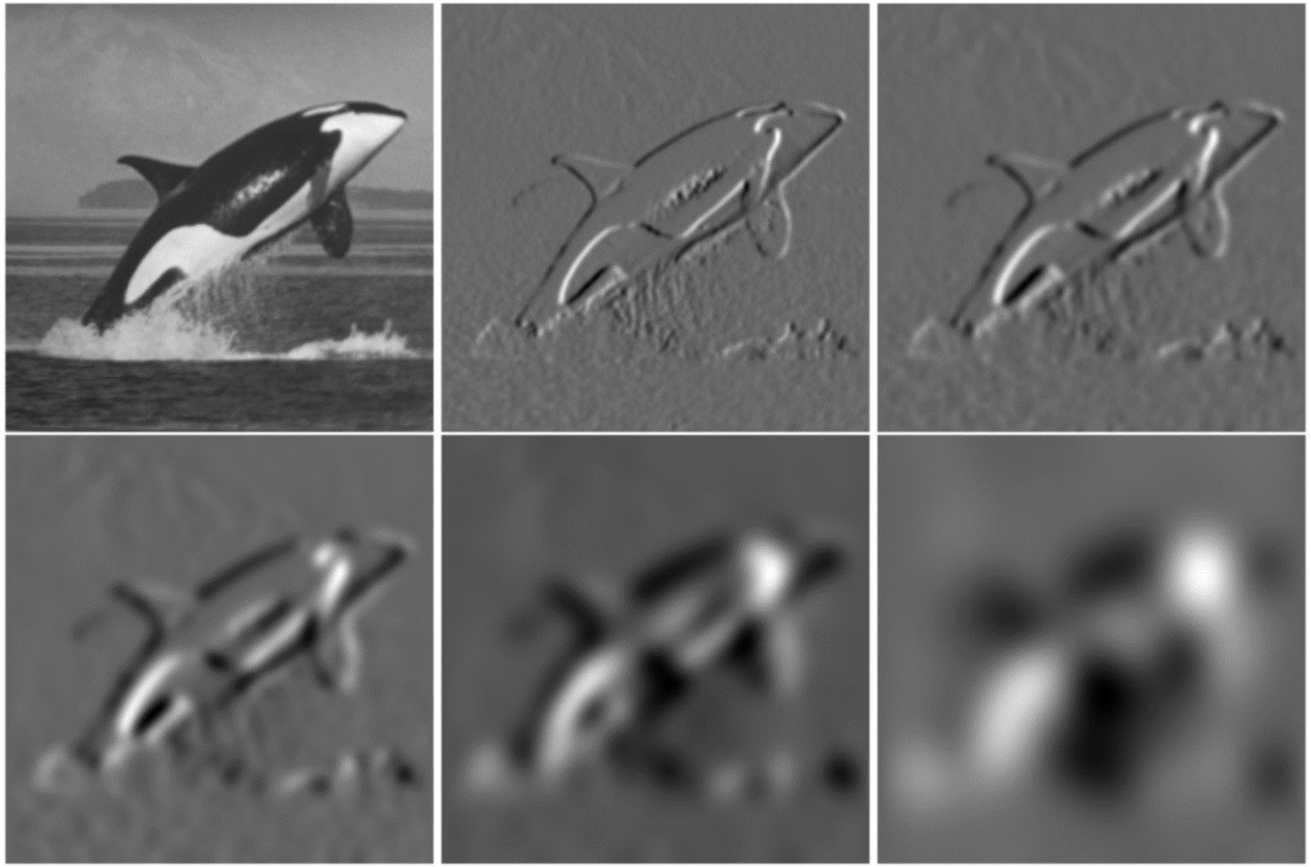




Source: Lindbergh

- Not only Gaussians, but also derivatives of Gaussians

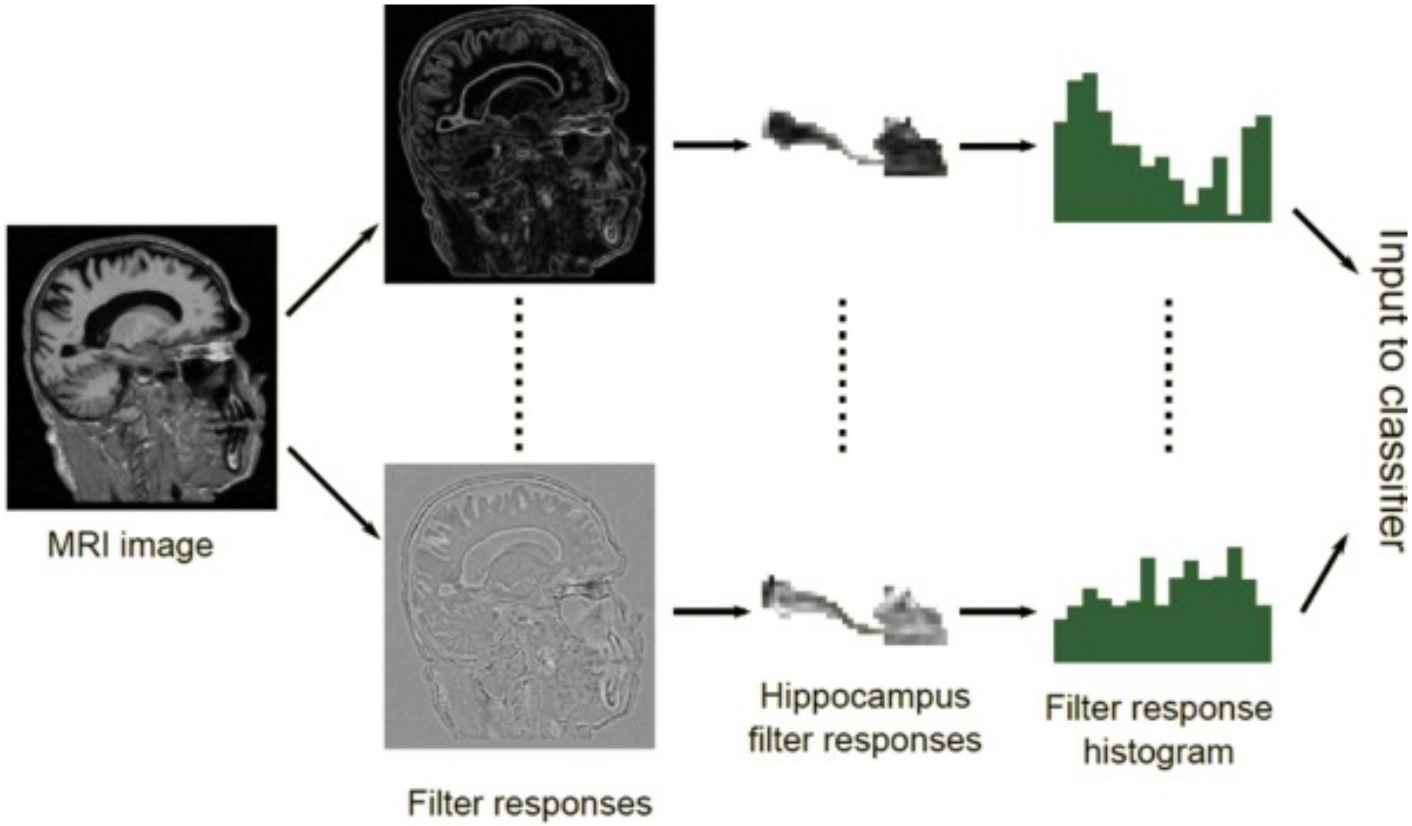




Histograms of filtered images

- How to compare images – we need a fixed-length representation

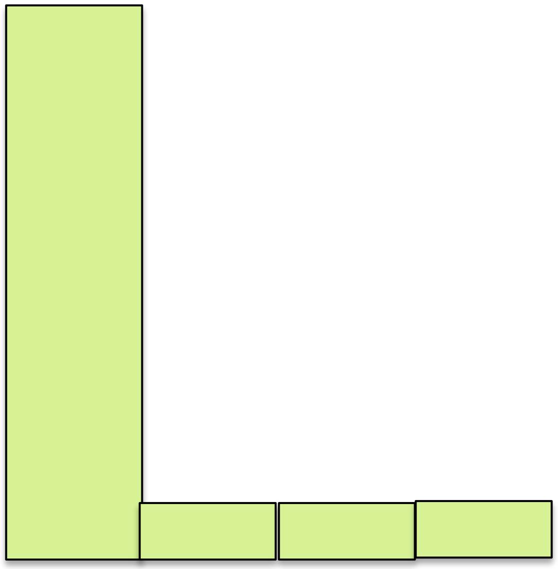
- We can create histograms (how often are patterns visible?)

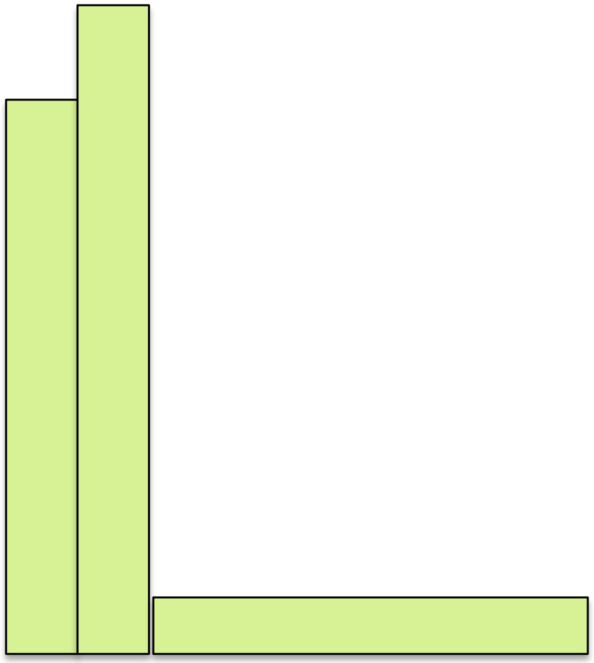


Source: Sorensen et al, Human Brain Mapping 2016

Histograms of filtered images

- This is a high-dimensional representation
- E.g. 6 filter types x 4 scales x 20 bins per histogram = 480 features
- Choosing the bins is important
  – Avoid “most data in few bins”
  – Adaptive binning instead
  – Determine bins on “representative” image beforehand



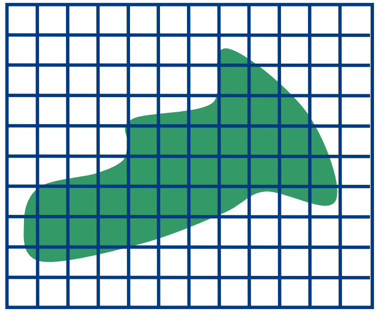


Dimensionality reduction

- Adding features can reduce class overlap
- But more features is not is necessarily better
  - Less intuitive / cannot visualize in 2D or 3D
  - Features can be redundant (highly correlated or not informative)
  - Larger dataset needed

Dimensionality reduction - example

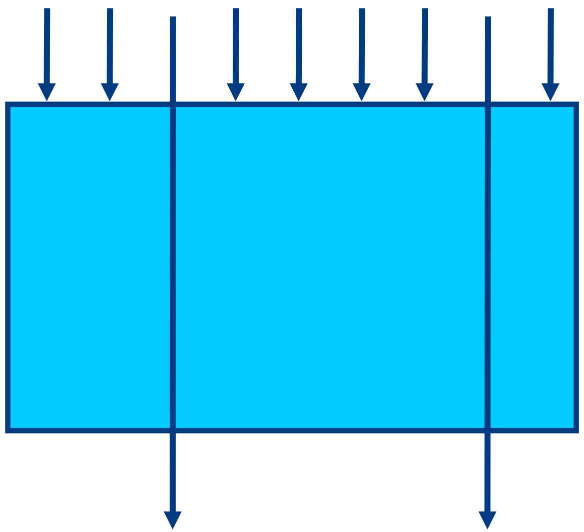
- In many applications, a lot of features will be redundant
  - Highly correlated
  - Uniformative
- Examples
  - Neighboring pixels or histogram bins
  - Top left pixel

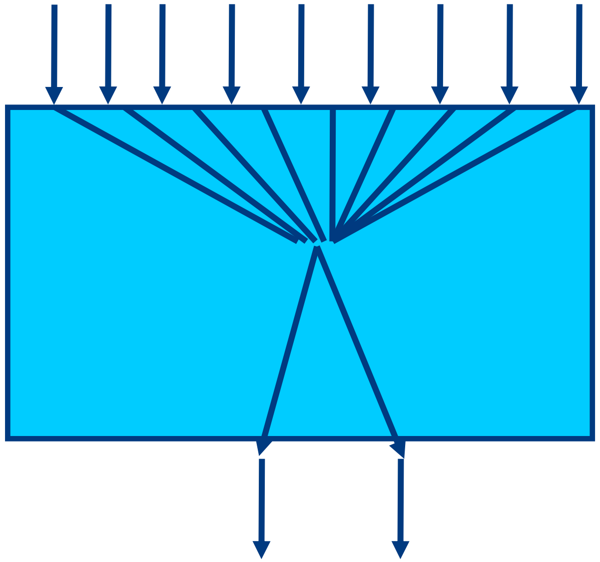


Each pixel = feature

Dimensionality reduction

- Start with p features
- Feature selection: select $k < p$ features
- Feature extraction: combine up to p features, into $k < p$ features
- [More on this after classifiers]





- Notebook

https://tinyurl.com/pwt7rjwp