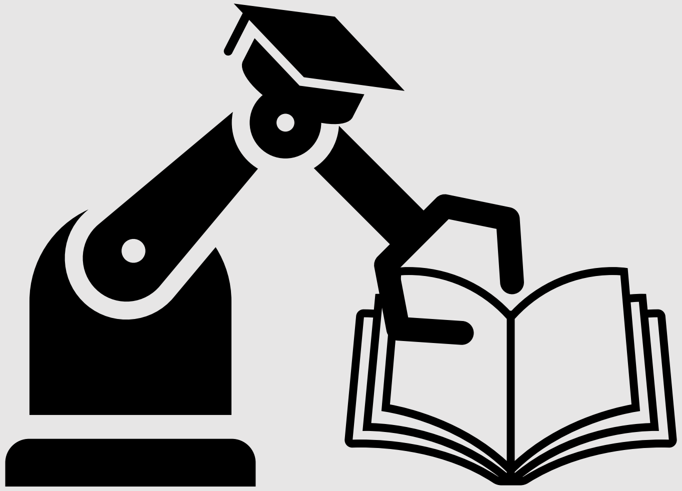
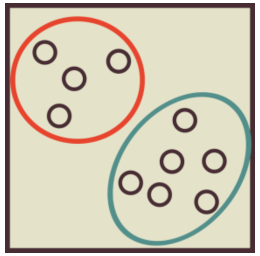
Unsupervised Learning

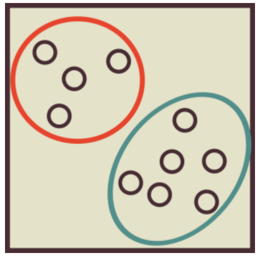


Clustering Algorithms

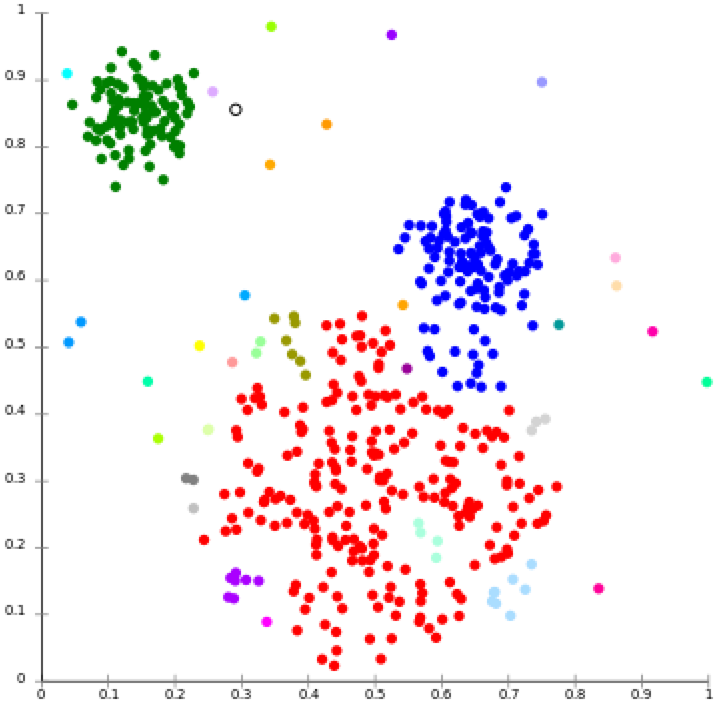


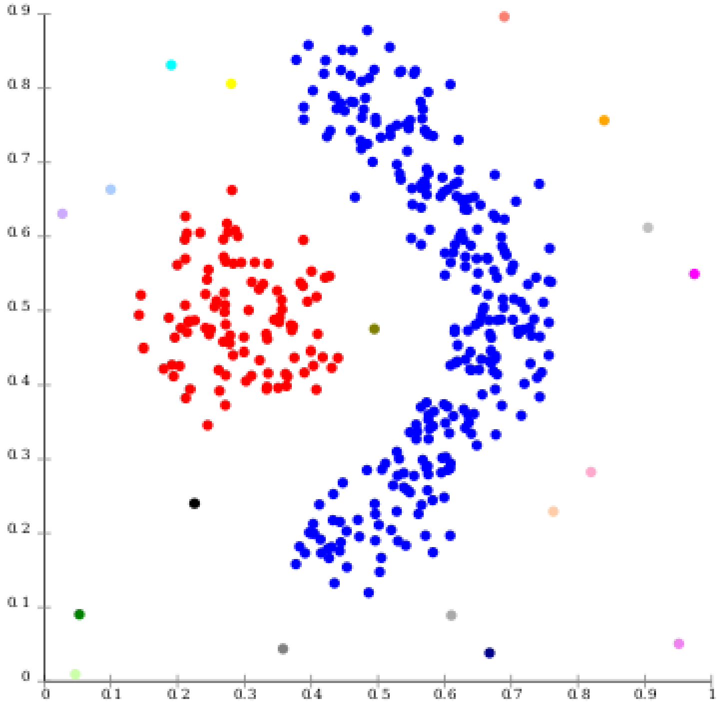
- Groups objects that are similar
- Typically organized by modeling approaches
- Two classes
  - Hard clustering
  - Fuzzy clustering
- Examples:
  - Connectivity based clustering
  - K-means
  - Distribution based clustering
  - Density based clustering

Connectivity-Based Clustering



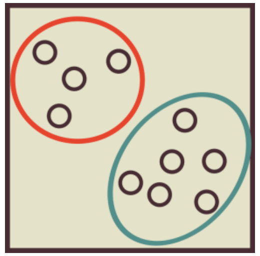
- Objects are more related to the nearby objects rather then those fare away
- Similarity measure – Euclidian distance or anything else



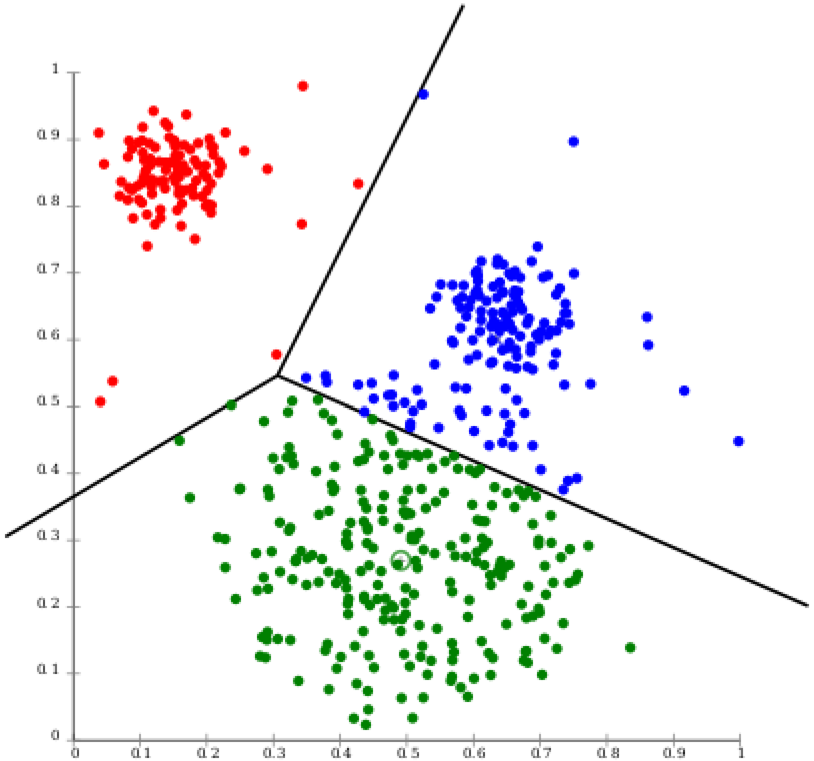


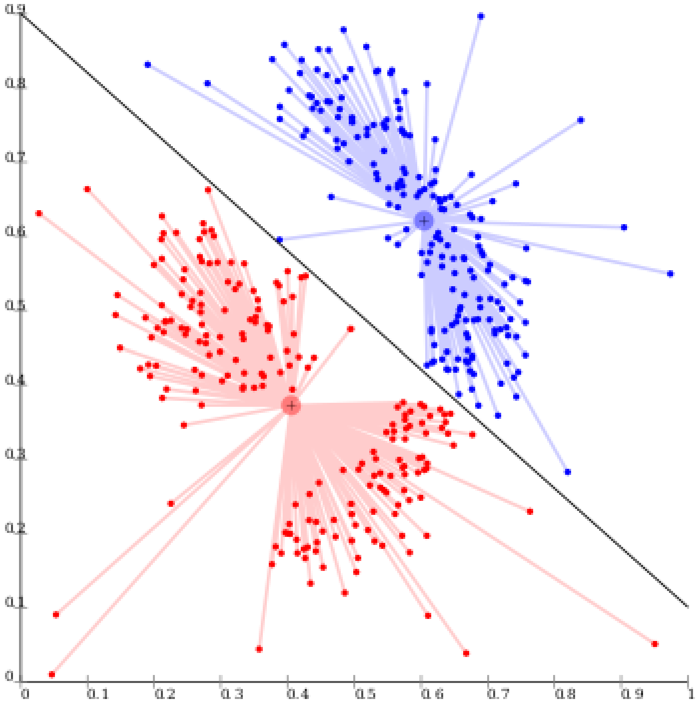
https://en.wikipedia.org/wiki/Cluster_analysis

Centroid-Based Clustering



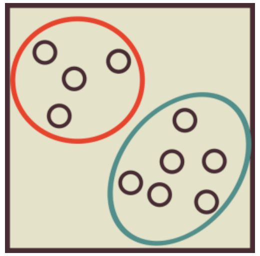
- Represented by center vector
- Center does not have to be necessary one of the data points



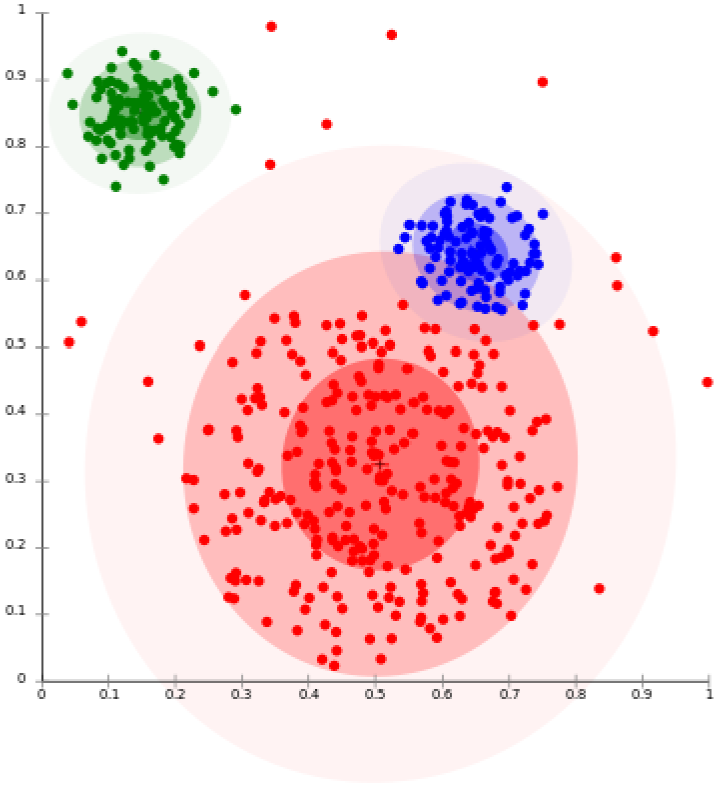


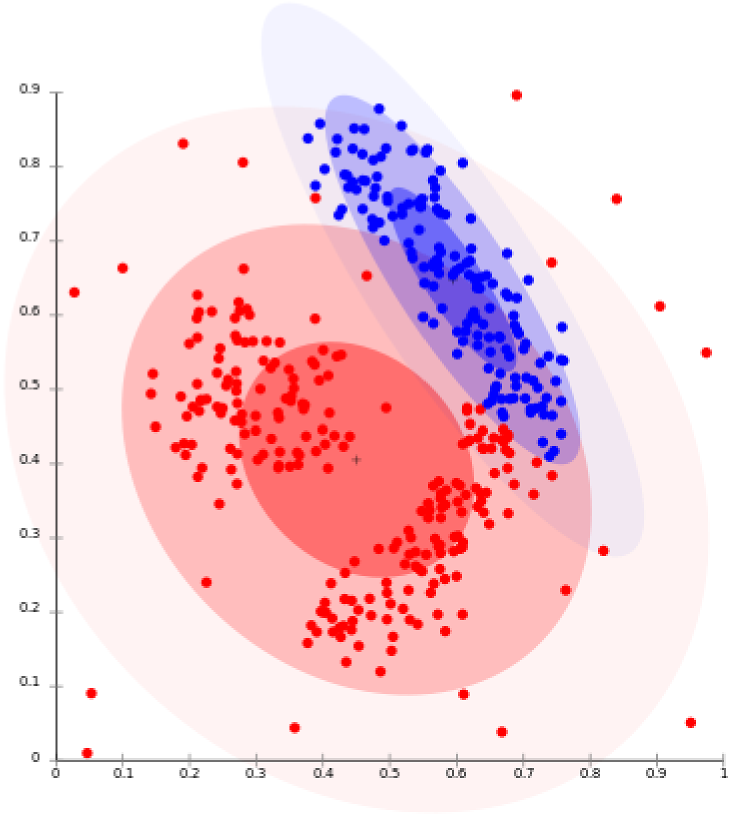
https://en.wikipedia.org/wiki/Cluster_analysis

Distribution-Based Clustering



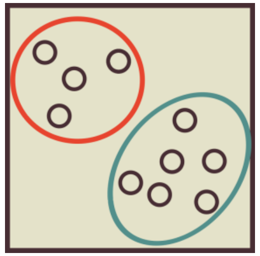
- Clusters defined as object belongings to the same distribution
- Convenient for artificial datasets, but suffer from overfitting in practice



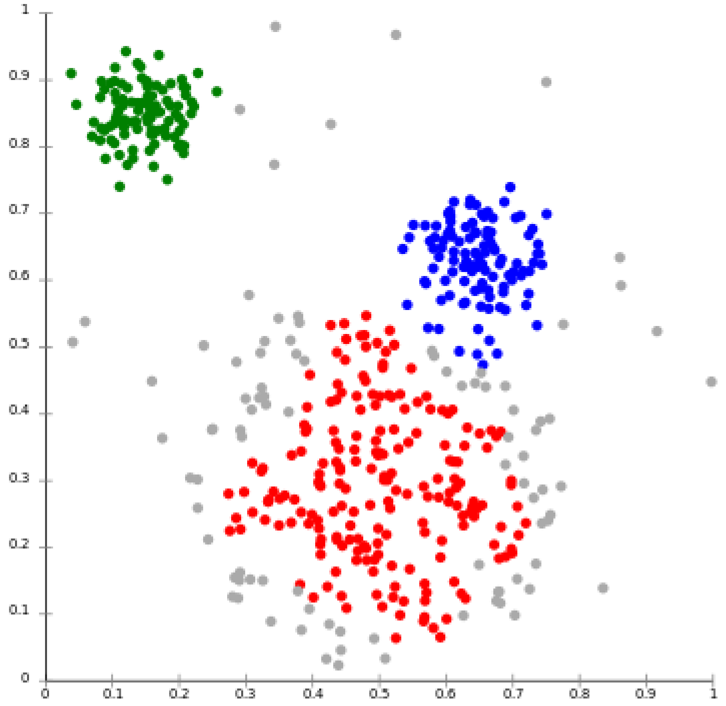


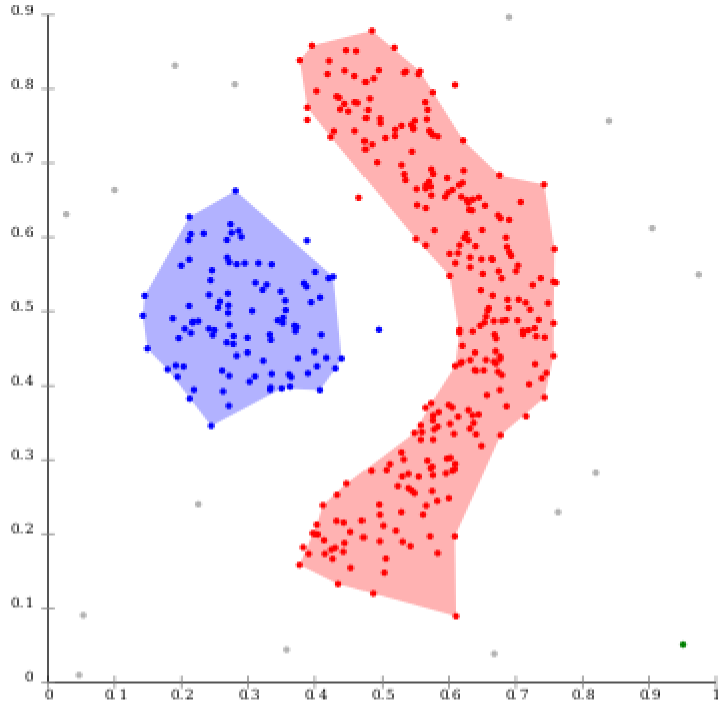
https://en.wikipedia.org/wiki/Cluster_analysis

Density-Based Clustering



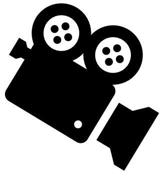
- Cluster defined as areas with higher density (require density drops)
- Objects in sparse areas considered to be noise



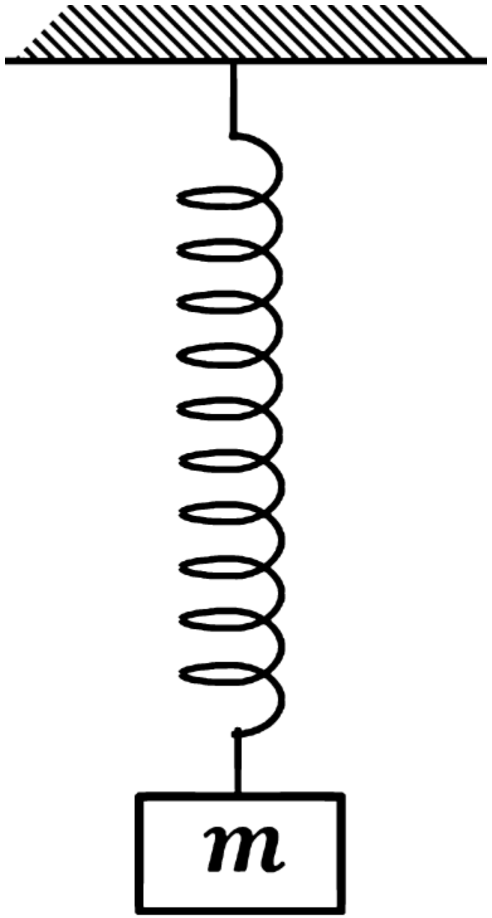


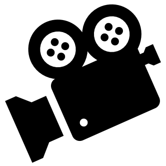
https://en.wikipedia.org/wiki/Cluster_analysis

Intrinsic Dimensions

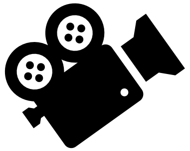


	1	2	3
X	…	…	…
Y	…	…	…
Z	…	…	…



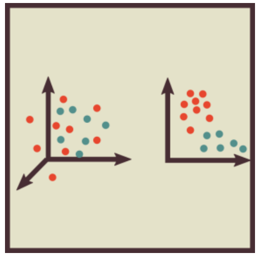


	1	2	3
X	…	…	…
Y	…	…	…
Z	…	…	…



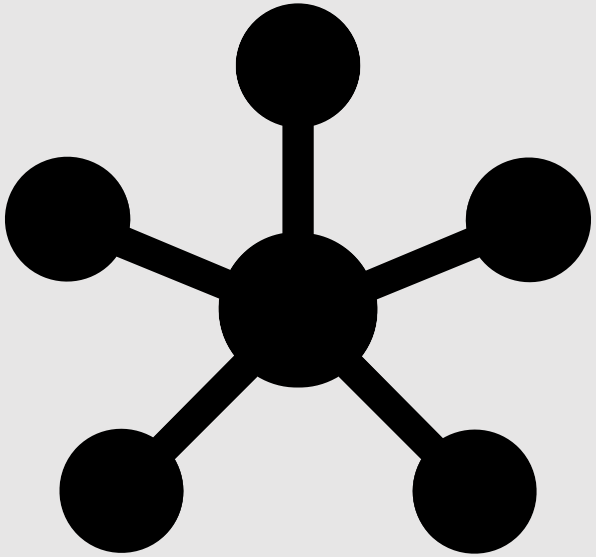
	1	2	3
X	…	…	…
Y	…	…	…
Z	…	…	…

Dimensionality Reduction Algorithms



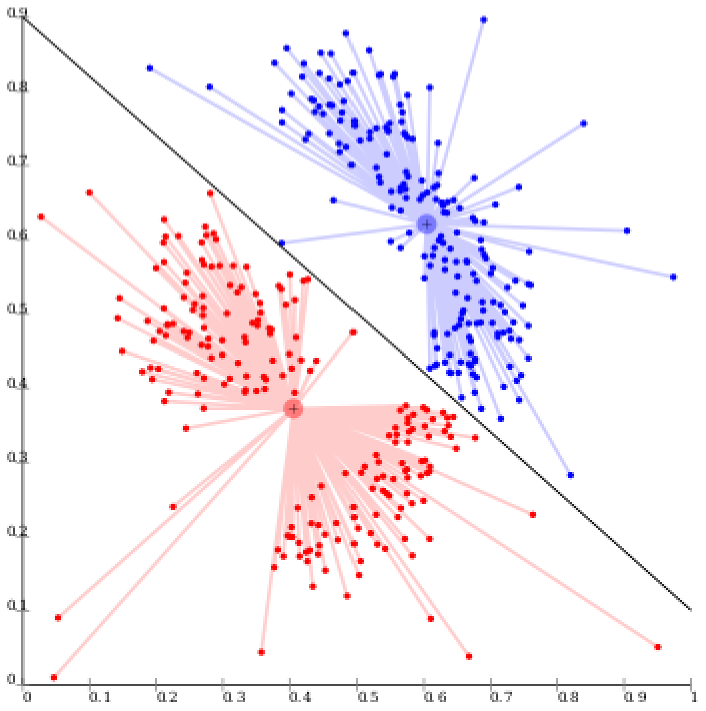
- Reducing number of random variables to a set a principal variables
- Finds structures in data to reduce dimensionality unsupervised
- Lower dimensional variables often visualized for labeling and further supervised learning
- Examples:
    - Principal component analysis (PCA)
    - Linear discriminant analysis (LDA)
    - t-distributed stochastic neighbor embedding (t-SNE)
    - Uniform manifold approximation and projection (UMAP)

K-means



K-means Introduction

- **Centroid-based clustering**
- Assumes Euclidean space/distance
- Advantage
  - Suitable for large datasets
  - Can be applied to non-well separated clusters
- Disadvantage
  - Requires to select the number of clusters ***k***



K-means Algorithm

- Input:
  - K (number of clusters)
  - Data set $\{x_1, x_2 \ldots x_m\}$
- Algorithm

1. Select randomly k centroids
2. Assign cluster indices to each point based on the distance to centroids
3. Update centroid locations
4. Repeat 2-3 until convergence (i.e., no change)

Selecting k Value

- Try different values and look for the average distance to centroid as k increases
- Alternatively use silhouette

Selecting Starting Points

- Naïve Approach
  - Select points randomly
  - Possible problems when selecting points in same place
- Approach 1: Sampling
  - Cluster a smaller subset of data using different clustering algorithm
  - Pick representatives from each cluster
- Approach 2: Dispersed Set
  - Select first point randomly
  - Next points select such they have a largest possible distance from already selected points

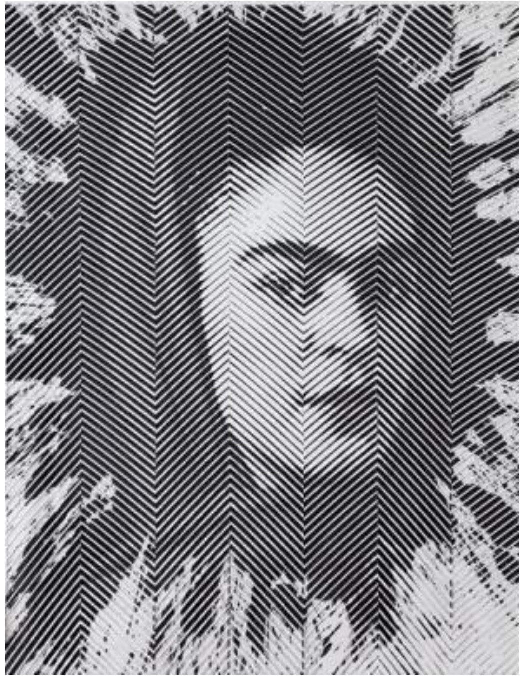
Complexity

- In each round we examine each input points once
  - O(kn) for n points and k clusters
  - The problem is the number of rounds to converge

Image Processing

Short Turing Test

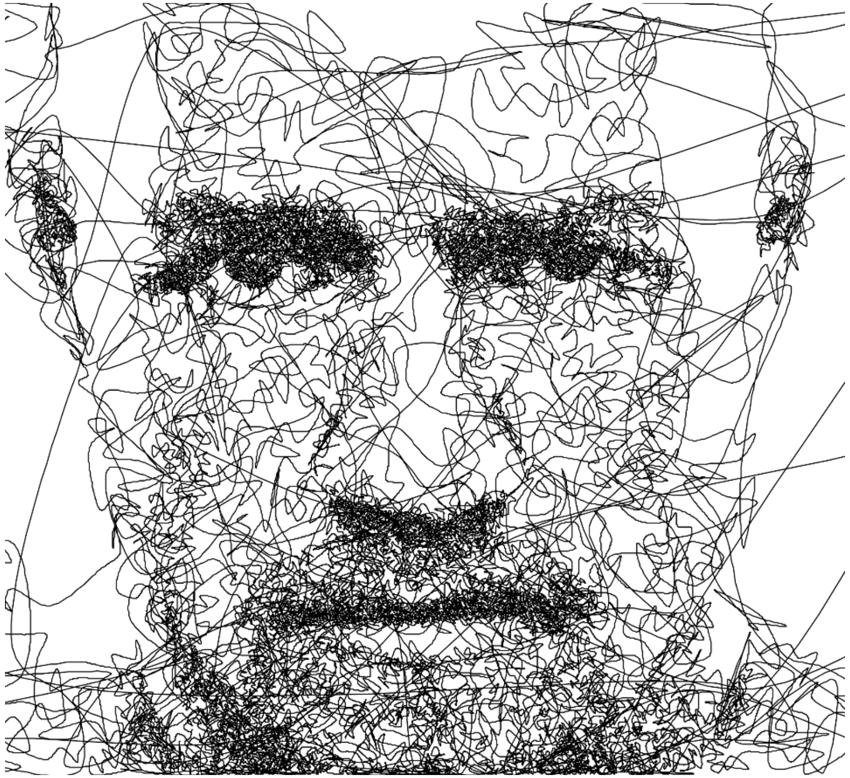
Machine



Human

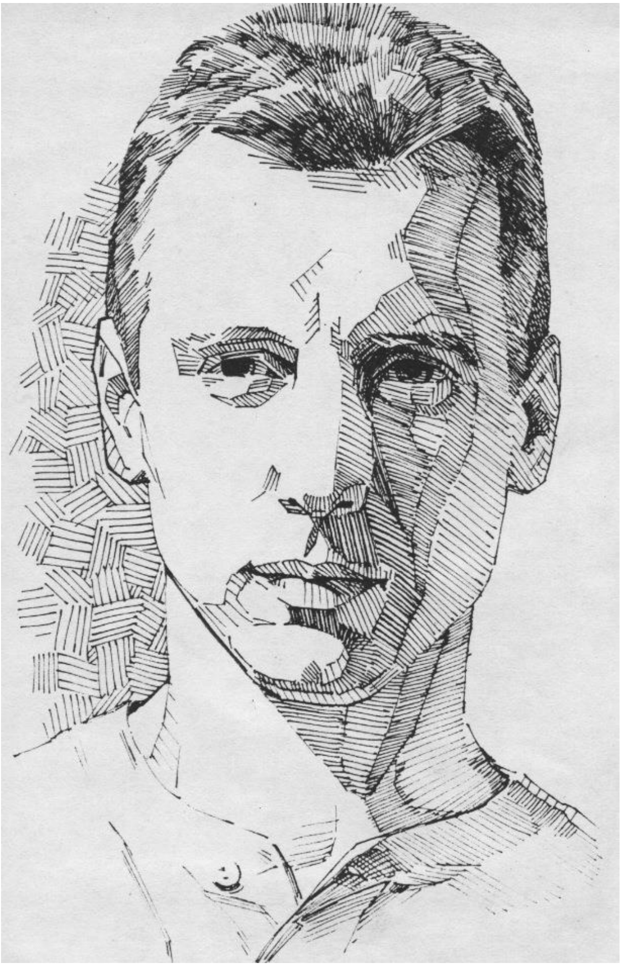
Short Turing Test

Human

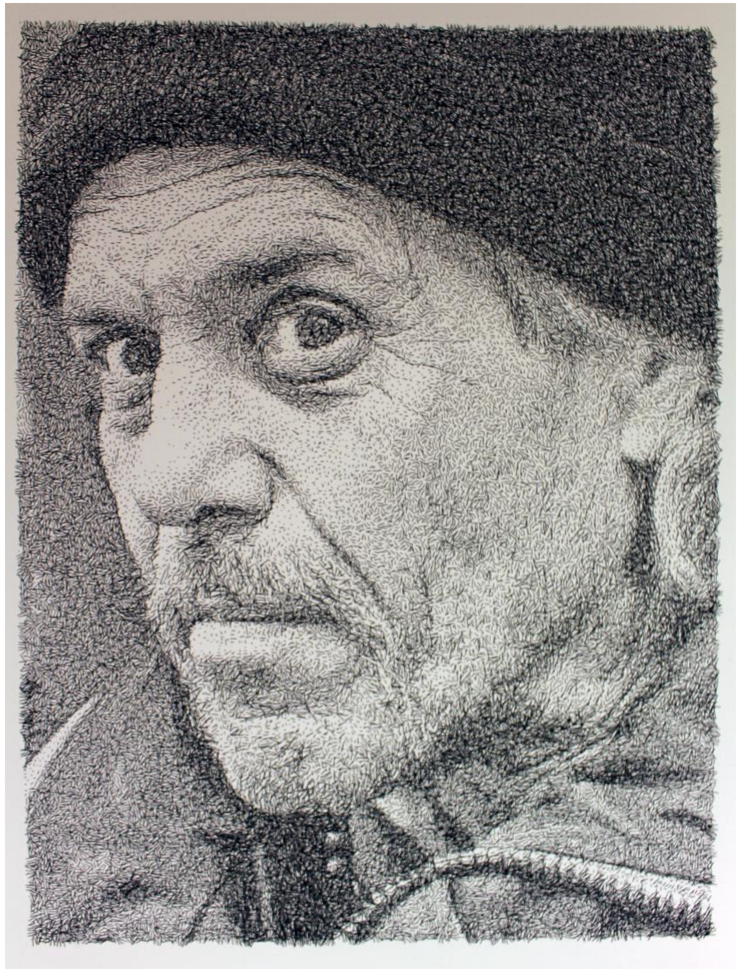


Machine

Short Turing Test



Human

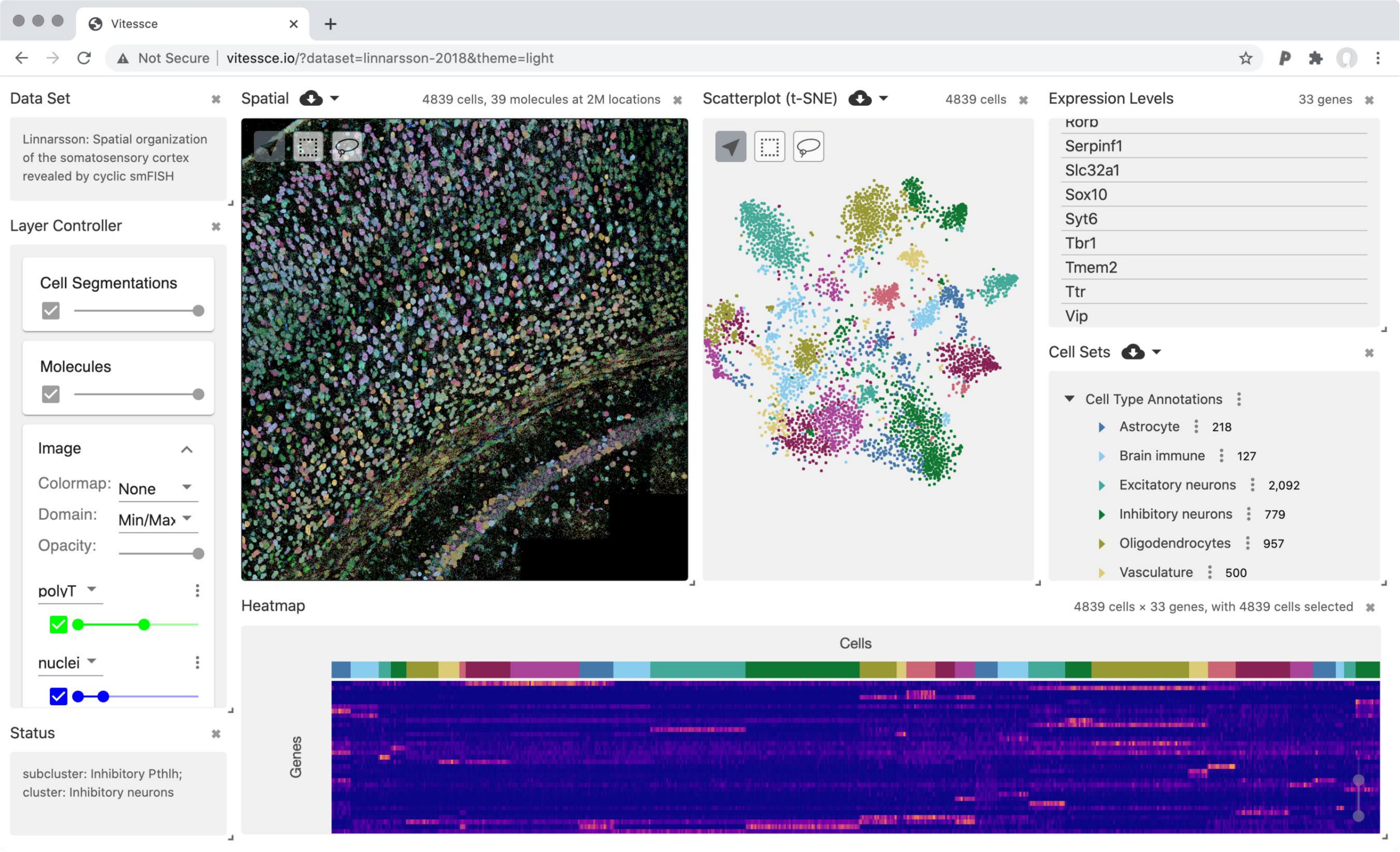


Machine

Short Turing Test

Machine

Human


Vitessce
Not Secure
vitessce.io/?dataset=linnarsson-2018&theme=light
Data Set
Linnarsson: Spatial organization of the somatosensory cortex revealed by cyclic smFISH
Layer Controller
Cell Segmentations
Molecules
Image
Colormap: None
Domain: Min/Max
Opacity:
polyT
nuclei
Status
subcluster: Inhibitory Pthlh; cluster: Inhibitory neurons
Spatial
4839 cells, 39 molecules at 2M locations
Scatterplot (t-SNE)
4839 cells
Expression Levels
33 genes
Rorb
Serpinf1
Slc32a1
Sox10
Syt6
Tbr1
Tmem2
Ttr
Vip
Cell Sets
Cell Type Annotations
Astrocyte 218
Brain immune 127
Excitatory neurons 2,092
Inhibitory neurons 779
Oligodendrocytes 957
Vasculature 500
Heatmap
4839 cells × 33 genes, with 4839 cells selected
Cells
Genes

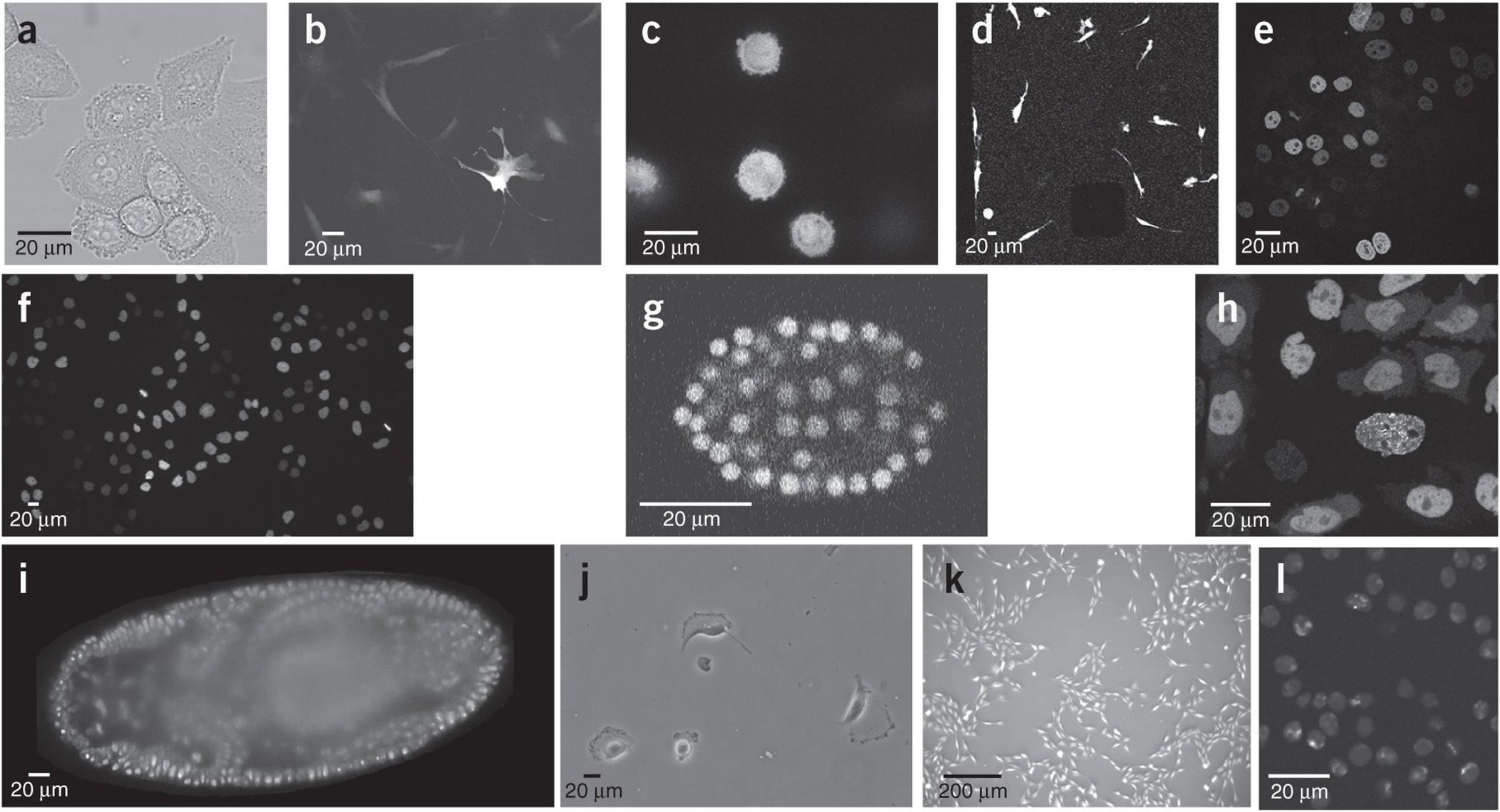
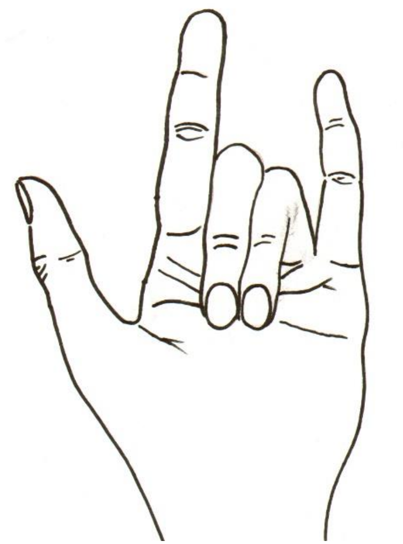

a
20 µm
b
20 µm
c
20 µm
d
20 µm
e
20 µm
f
20 µm
g
20 µm
h
20 µm
i
20 µm
j
20 µm
k
200 µm
l
20 µm

Image Recognition

- This lecture focuses on a single example of image recognition
- Humans mostly focus on local outstanding features and contours
  - Need a technique to detect those local characteristics

Source: Artwork by Matt Small



Source: Arts with Miss Griffin; Types of Contours

Image Recognition

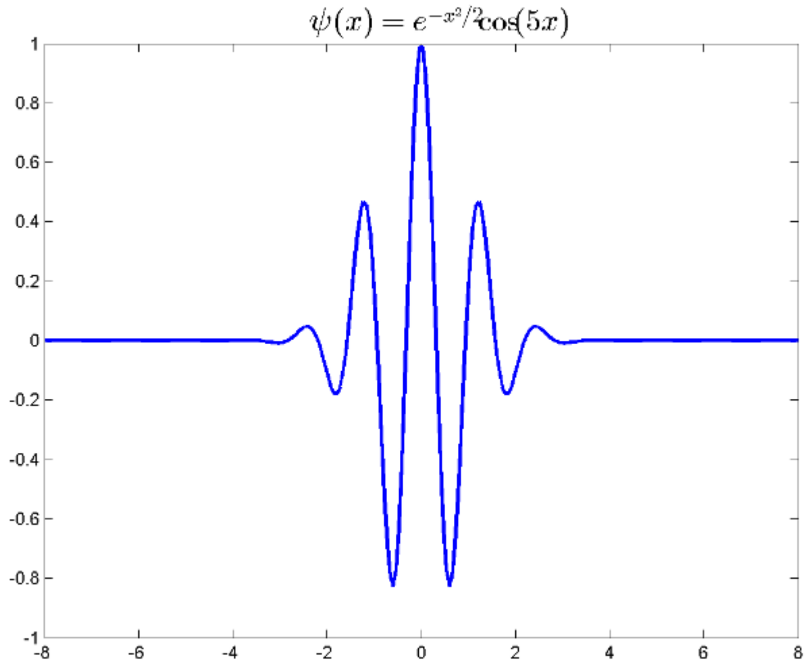
- Goal: a program that recognizes classes (circles and rectangles) in an image, learned through a labeled training set
- TODO's
  - Transfer the images to a basis suitable for edge detection and local features
    - Wavelet decomposition
  - Find the features associated with different classes
    - Principal components
  - Design a statistical decision mechanism for determination of new objects
    - Linear discrimination analysis

Decomposition Revisited

- Some well-known decompositions
  - SVD, PCA
  - …
- There are many more decompositions out there
- Principle
  - Find a suitable basis
  - Find coefficients to represent the data
- Wavelet decomposition is yet another decomposition where the basis consists of wavelets

What is a Wavelet?

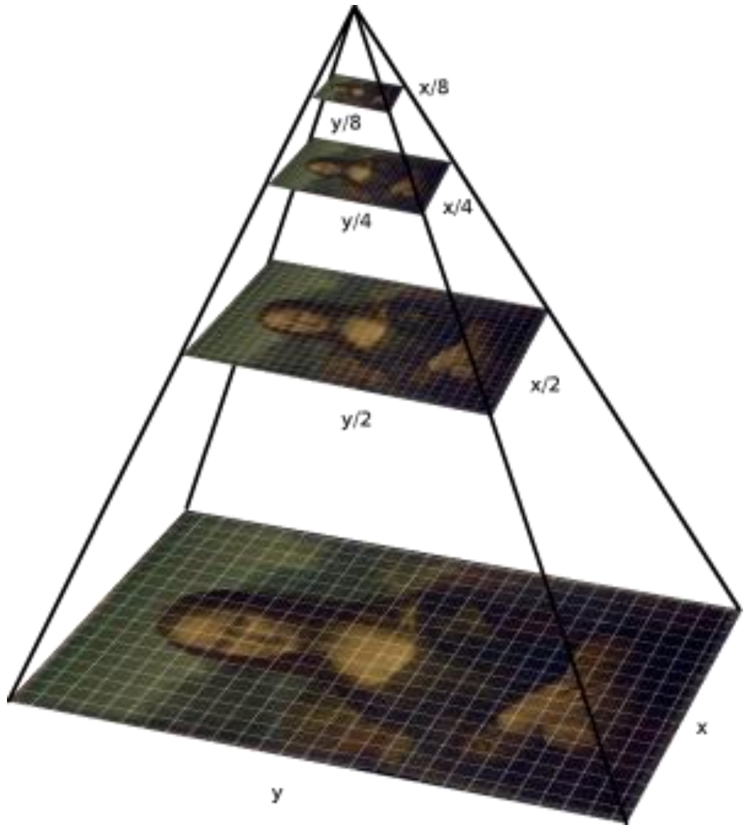
- A wavelet is an oscillation function, with an amplitude that begins at zero, increases and ends at zero.
- Wavelets can be combined to create other more complex functions.



Source: Mathworks

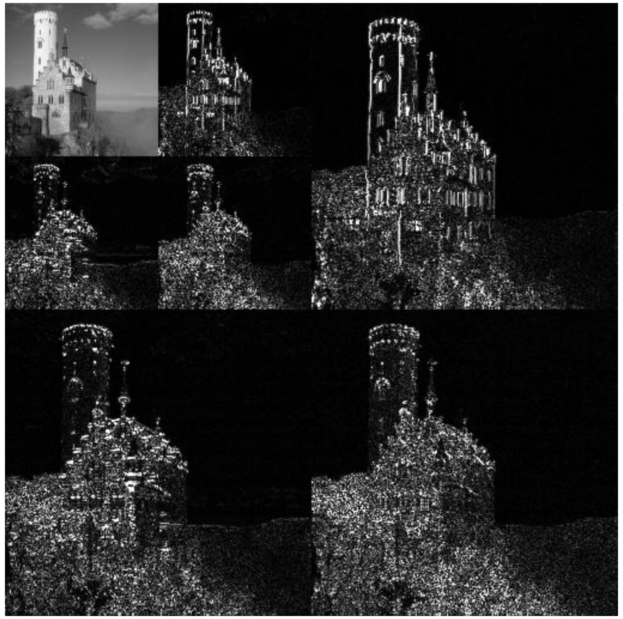
Why Wavelets?

- Wavelets are spatially localized
- Perfect for non periodic functions/signals
- Pyramid representation



Wavelet Decomposition

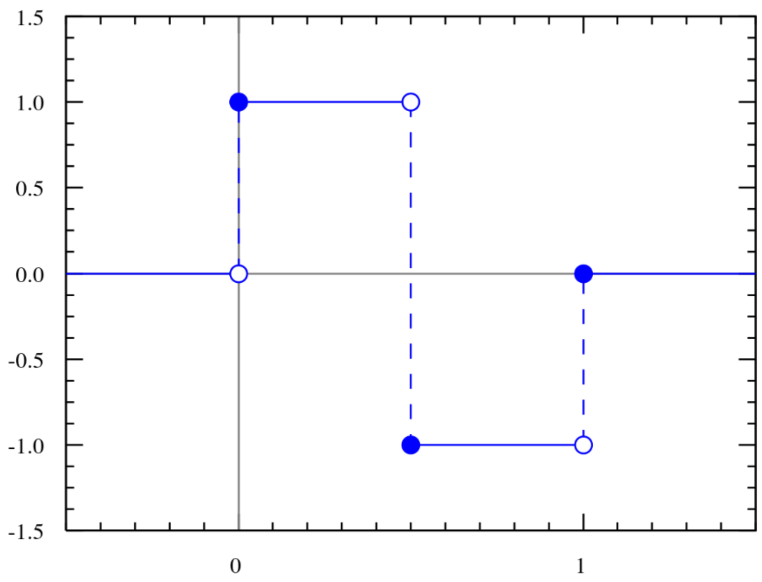
- Wavelets are ideal way to represent multi-scale information
  - Very efficient in detecting and highlighting of edges
  - Image data is often represented in wavelets for machine learning and data analysis
  - Wavelets are able to detect local changes in the data - they can «march along the data“



Source: Wikipedia Wavelet_transform

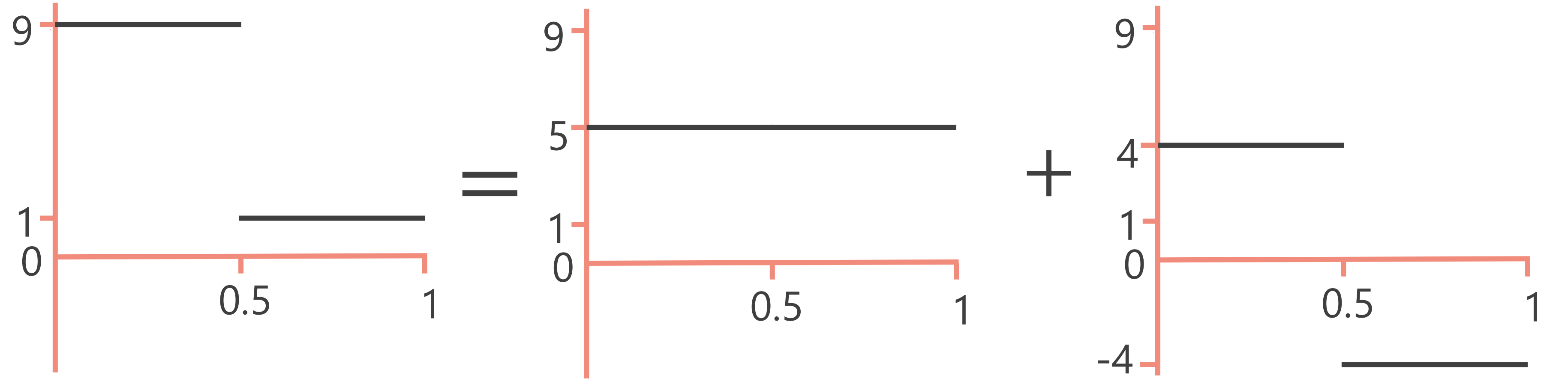
Wavelet Decomposition

- Every wavelet can be described through a mother wavelet function ψ and a mother scaling function ϕ.
- The simplest wavelet is the Haar wavelet
    - Was developed by Alfred Haar in 1909
- The simplest and most widely adopted wavelet basis and looks like this:



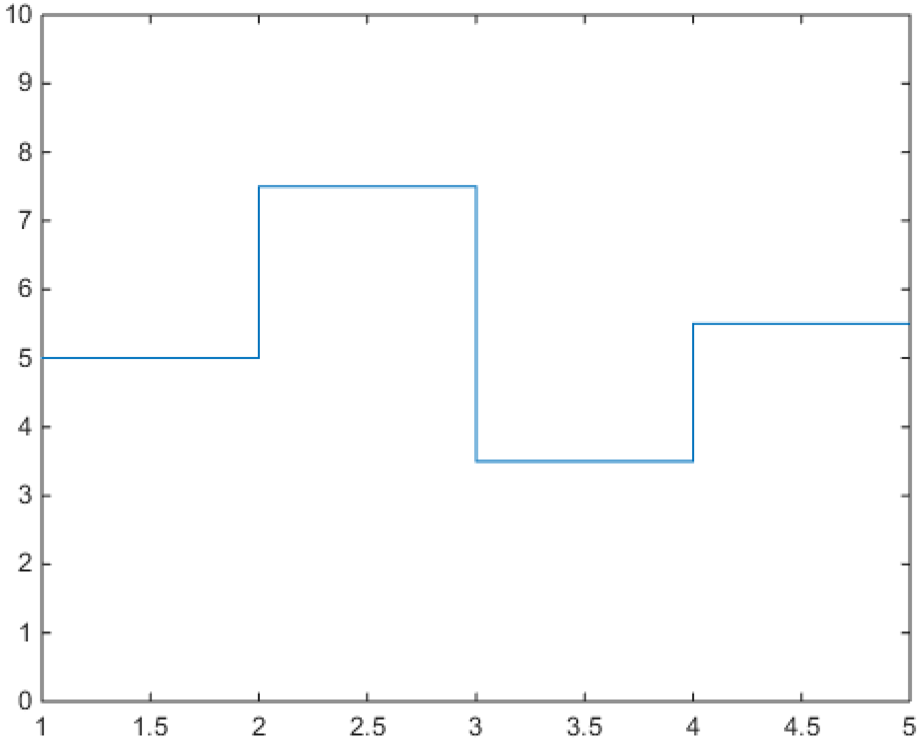
Haar Wavelet

- Let’s start with an example
- The signal below can be expressed as a combination of an average and difference function:
  - $5 \cdot \phi(t)$ $\qquad \phi(t) = \begin{cases} 1 & 0 \leq t < 1, \\ 0 & \text{otherwise.} \end{cases}$ mother scaling function
  - $4 \cdot \psi(t)$ $\qquad \psi(t) = \begin{cases} 1 & 0 \leq t < \frac{1}{2}, \\ -1 & \frac{1}{2} \leq t < 1, \\ 0 & \text{otherwise.} \end{cases}$ mother wavelet function



Haar Wavelet

- Now consider a more complex signal
- For a 1-D discrete signal with length 2^N we can remove the average & difference of two neighboring values to obtain 2^{N-1}scale coefficient and 2^{N-1} detail coefficients.



Haar Wavelet

- Let's look at the values:
  - Y = [1, 9, 8, 7, 3, 4, 5, 6]
- The averages of the neighboring values and the differences:
  - cA = [5.0, 7.5 , 3.5, 5.5] and cD = [-4.0, 0.5 , -0.5, -0.5]
- Vector [cA,cD] is a single level wavelet decomposition
- Single level means that the decomposition step was performed once

Haar Wavelet

- Let's look at the values:
  - Y = [1, 9, 8, 7, 3, 4, 5, 6]
- The averages of the neighboring values and the differences:
  - cA1 = [5.0, 7.5 , 3.5, 5.5] and cD1 = [-4.0, 0.5 , -0.5, -0.5]
- Repeat with new averages:
  - cA2 = [6.25, 4.5] and cD2 = [-1.25, -1]
- Oone last time:
  - cA3 = 5.375 and cD3 = 0.875
- The 3-level wavelet transform of Y is now [cA3,cD3,cD2,cD1]

Orthonormal Wavelet Basis

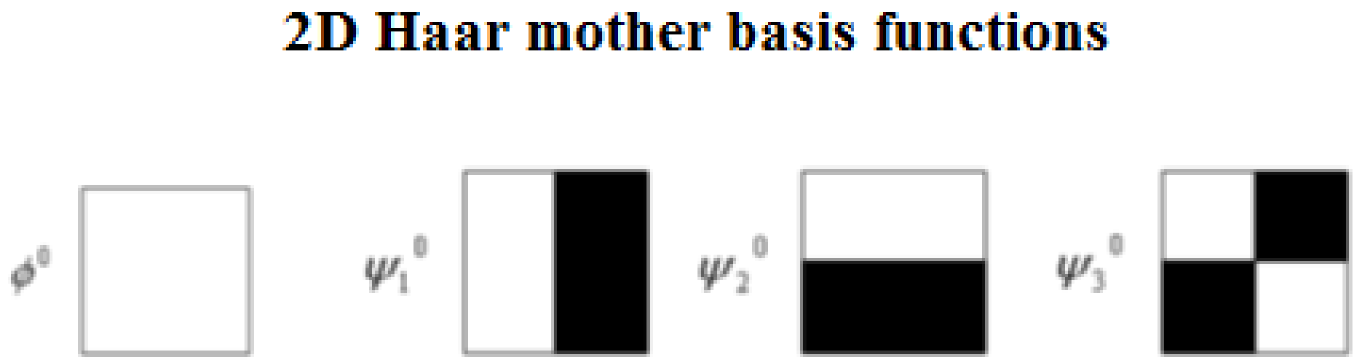
- Need orthonormal basis for representation
- Orthonormal if:
  - Means via: (a+b)/sqrt(2)
  - Differences via: (a-b)/sqrt(2)

Odd Length Signals

- Two strategies
  - Preferable: copy the last value
  - Alternative: remove last data point

Haar Wavelet

- What about 2D?
- In 2D the principle stays the same, for single level decomposition:
  - Average over cells with four elements
  - Compute the horizontal differences
  - Compute the vertical differences
  - Compute the diagonal differences



Source: Wavelets for computer graphics: A Primer

Haar Wavelet

- Multi-level decomposition one has to choose the order of the decomposition
- Iterate over averages
- Remeber all computed diffeerences

Figure 5 Standard decomposition of an image.

An example from **Wavelets for Computer Graphics: A Primer** [1]

Figure 6 Nonstandard decomposition of an image.

An example from **Wavelets for Computer Graphis: A Primer** [1]

Haar Wavelet

Source: https://chengtsolin.wordpress.com

100%

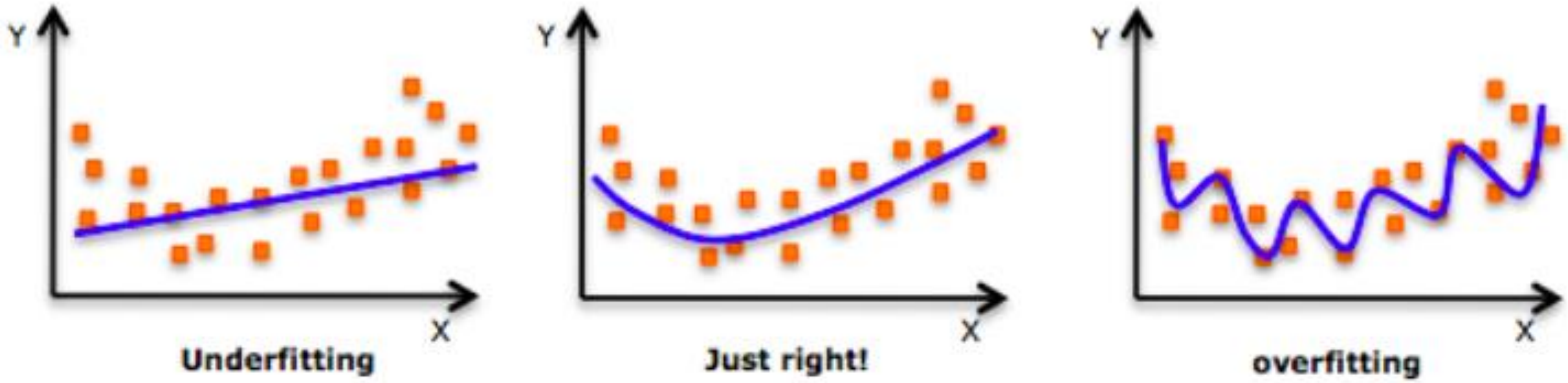
1%

0.1%

Back to Image Recognition

- Achieved: new representation of the labeled images
  - Local changes encoded
- TODO: create a decision mechanism based on edges
  - Circle vs. rectangle
- Create new data which encodes the edges
- Only small fraction of PC needed to describe the data sufficiently
  - Remember SVD
- Next step find a number of principal components which are associated with the objects

Underfitting / Overfitting

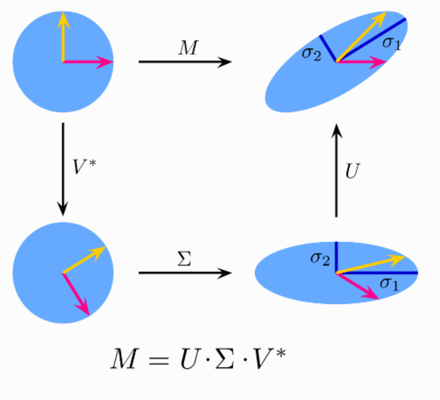


https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/

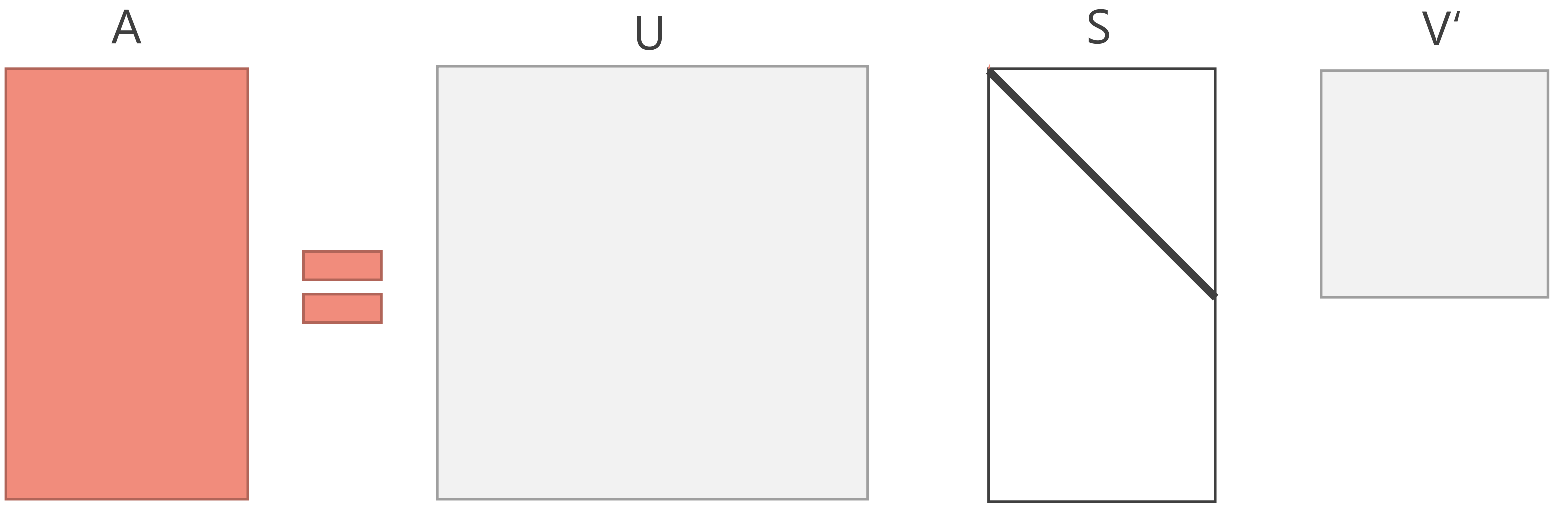
SVD for Image Recognition

- Assume we have the edge data of n cube images and m sphere images (ed_cu,ed_sp). We want to characterize the images based on k features.
- Perform SVD on the stacked data:
  - [U,S,V] = svd([ed_cu,ed_sp])
- Lets take a closer look at the decomposed matrices.

SVD for Image Recognition



- [U,S,V] = svd([ed_cu,ed_sp])= svd(A)
- S: impact of single values
- objects=S*V’ is a new basis
- Size of objects is dependent on the number of samples only

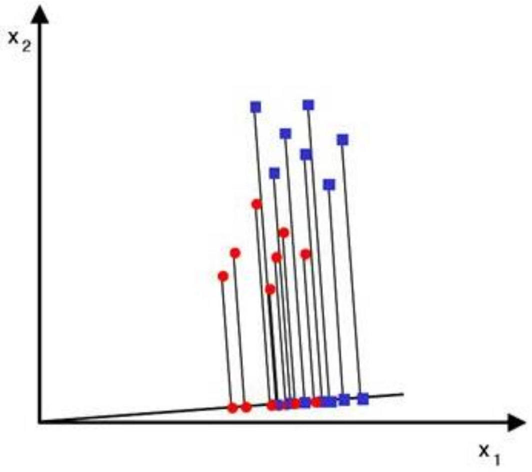


Linear Discrimination Analysis

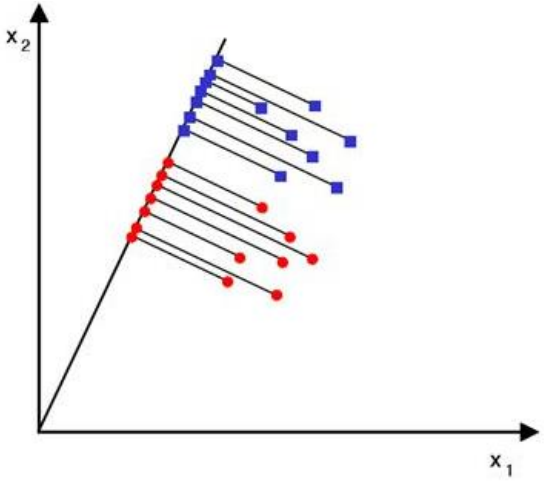
- Having detected a number of principal components of a class we can now set up a statistical decision mechanism to identify objects in new images.
- One possible way to do so is to use linear discrimination analysis (LDA)
- LDA aims to reduce the dimensionality while preserving as much of the class discriminatory information as possible

LDA Illustration

- Assume we have a set of high dimensional samples x with two classes ω_1 and ω_2: spheres and cubes.
- We seek to obtain a scalar y by projecting the samples x onto a line, $y = w^T x$.
- Of all the possible lines we want to select the one that maximizes the separability of the two groups.



Bad projection



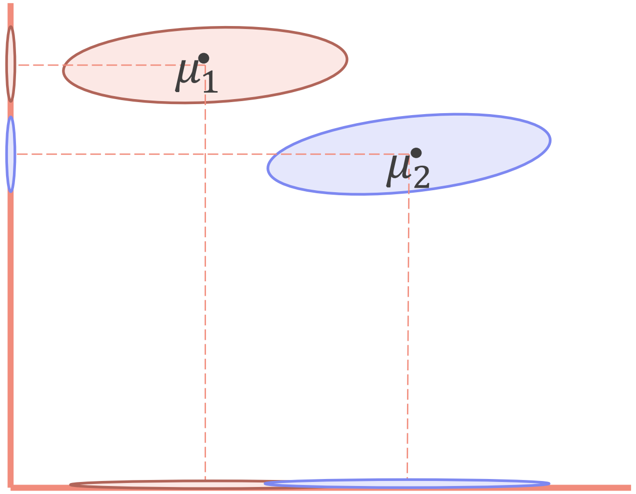
Good projection

LDA

- In order to find a good projection we need to define a measure of separation.
- One possibility is to compute the mean vector μ_i of each class ω_i and use the distance between the projected means as our objective function:

$$\tilde{S}_B = |\tilde{\mu}_1 - \tilde{\mu}_2| = |w^T (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T w| = w^T S_B w$$

- But considering just he mean is not enough:



Overlap
High distance

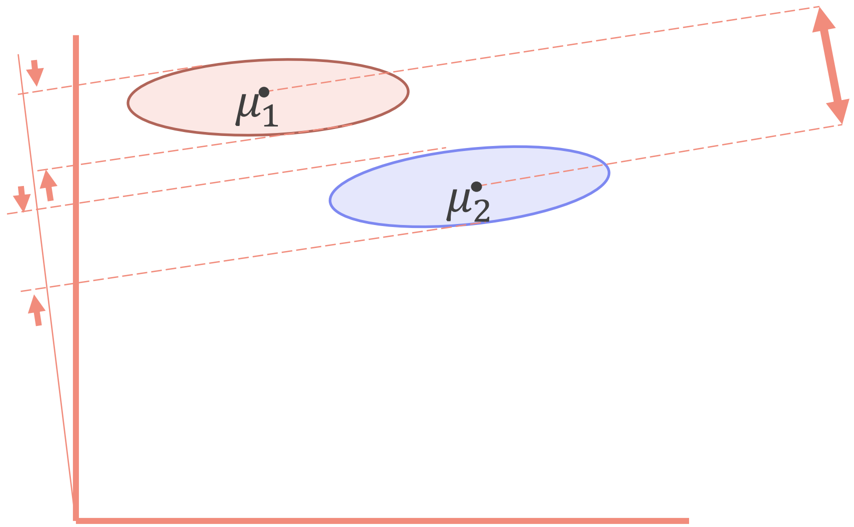
LDA

- Fisher suggested maximizing the difference between the means, normalized by a measure of the within-class differences.
- For each class define the scatter, an equivalent of the variance as

$$S_W = \sum_{j=1}^{2} \sum_{x} (x - \mu_J)(x - \mu_j)\,; \quad \tilde{S}_W = w^T S_W w$$

- The Fisher linear discriminant is defined as the linear projection w that maximizes the criterion function

$$J(w) = \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$



- Such a projection mimizes the distance within the class and maximizes the distance between classes

LDA

- Solving the generalized eigenvalue problem $S_W^{-1} S_B w = J(w) = \lambda w$ gives us the solution:

$$w^* = \arg max \left[\frac{w^T S_B w}{w^T S_W w} \right] = S_W^{-1} (\mu_1 - \mu_2)$$

- This solution is known as Fisher's linear discriminant, even though this is not a discriminant but a specific choice of the projection direction of the data down to one dimension.
- This projection can now be used to distinguish between the two groups.
- One simple method: $w^* x > threshold \Rightarrow$ cube, everything else is a sphere.
- More sophisticated methods can be used for the classification