THE UNIVERSITY OF WESTERN AUSTRALIA

CITS 4402 Computer Vision

A/Prof Ajmal Mian
Adj/A/Prof Mehdi Ravanbakhsh

Lecture 06 – Object Recognition

Objectives

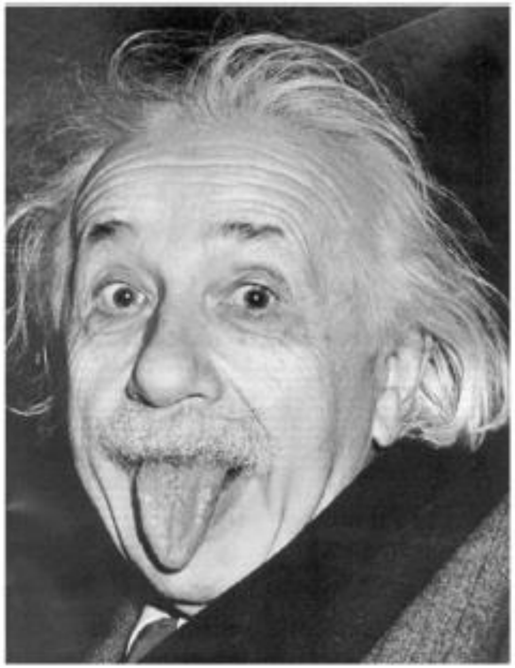
- To understand the concept of image based object recognition
- To learn how to match images beyond simple template matching
- To study the object recognition pipeline
- To learn about classification algorithms
- A brief introduction to machine learning

Specific Recognition Tasks

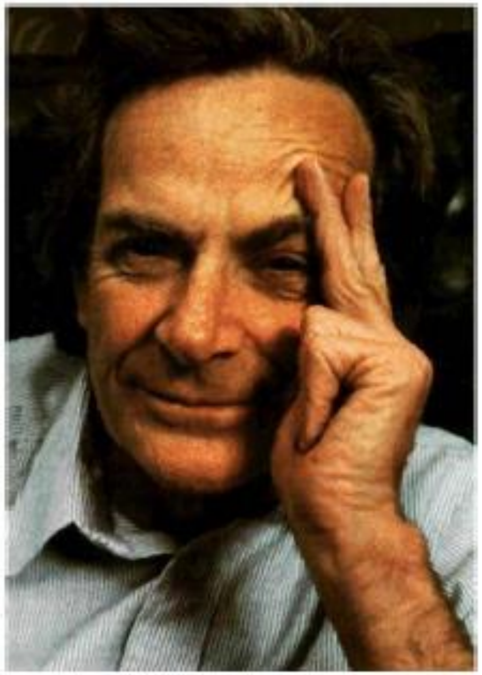
- Recognition is a core computer vision problem
- Scene classification (Outdoor / Indoor)
- Image tagging (street, people, tourism, mountain, cloudy, …)
- Find pedestrians

What is “Recognition”?

- Identification



vs.



What is “Recognition”?

- Categorization

vs.

Detection versus Recognition?

1. Face detection: Where are faces in an image?

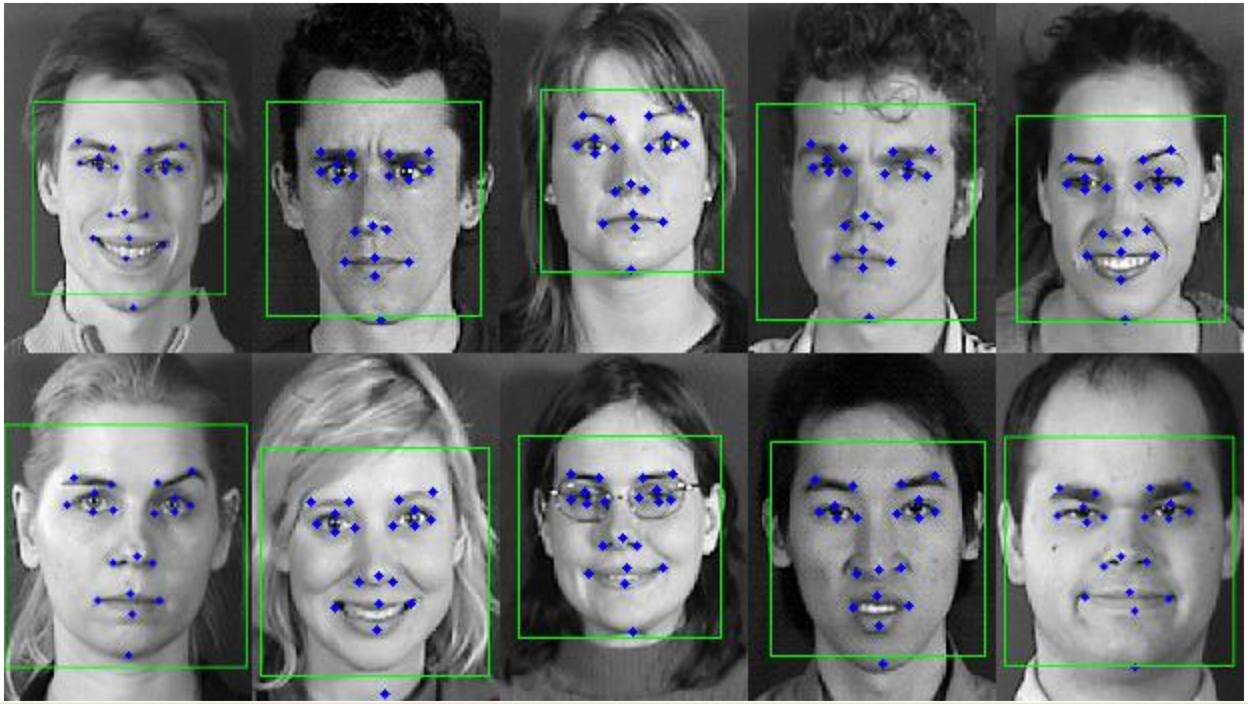
2. *Specific Person detection: Where is Person X in an image?*

3. Face recognition: Given a face image (from step 1), find the identity of the person.

4. Can you define pedestrian detection and stop sign detection now?

Scale of detection

- Faces may need to be detected at different scales
- We can have nested detections
  - Detect face
  - Detect features such as eye corners, nose tip etc



Visual Recognition

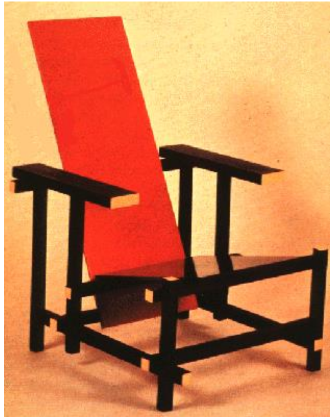
- Design algorithms that are capable of
  - Classifying images or videos
  - Detect and localize image
  - Estimate semantic and geometrical attributes
  - Classify human activity and events

- Why is this challenging?

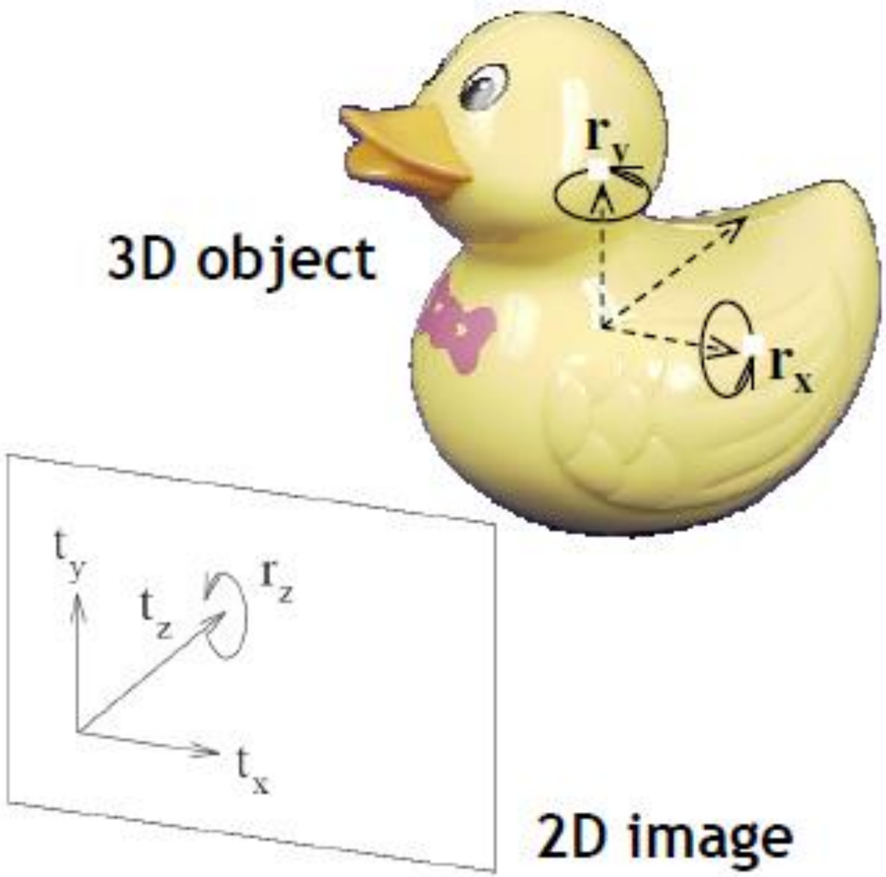
How many Object Categories are there?

Challenges – Shape and Appearance Variations

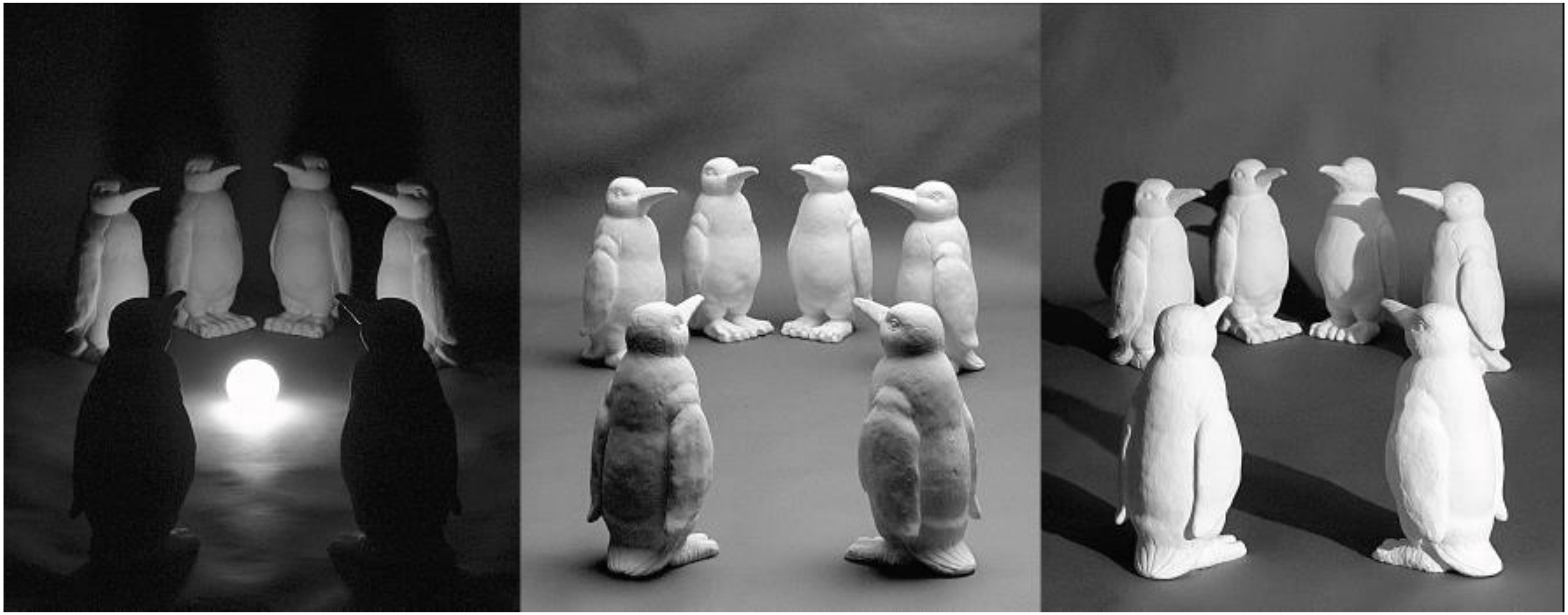


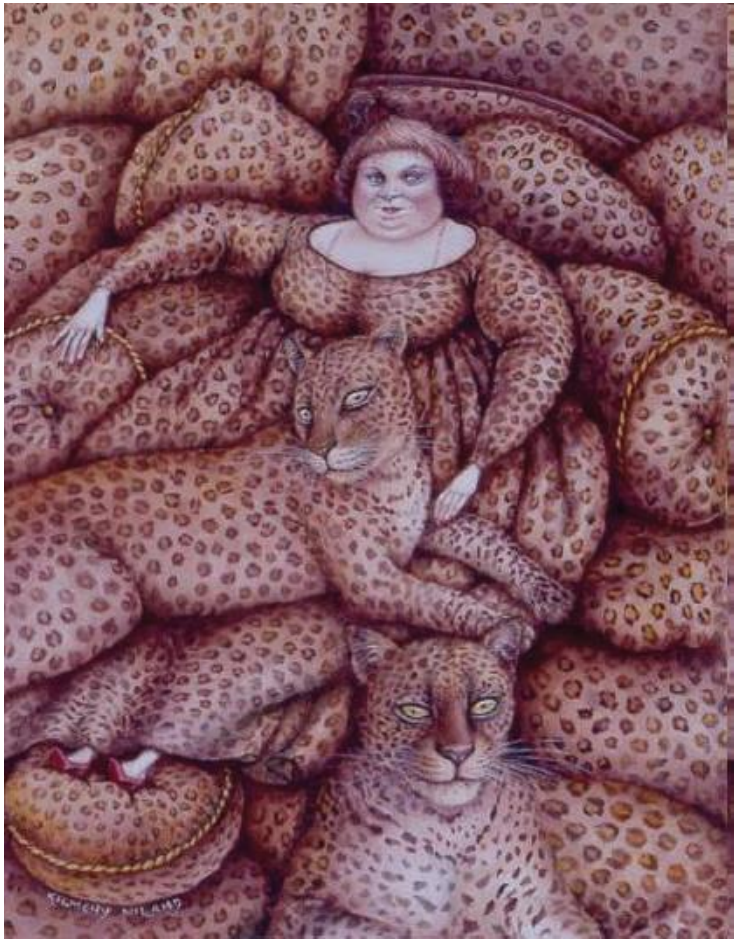
Challenges – Viewpoint Variations



Challenges – Illumination



Challenges – Background Clutter



Challenges – Scale

Challenges – Occlusion

Challenges do not appear in Isolation!

- Task: Detect phones in this image

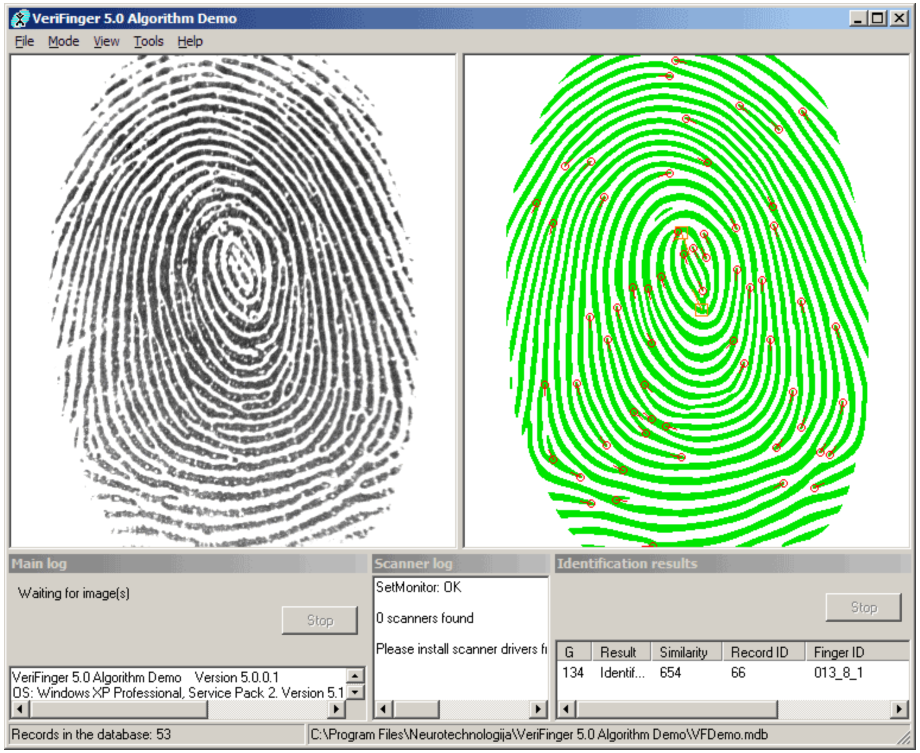
- Appearance variations
- Viewpoint variations
- Illumination variations
- Background clutter
- Scale changes
- Occlusion

What "Works" Today

- Reading license plates, zip codes, checks

What "Works" Today

- Reading license plates, zip codes, checks
- Fingerprint recognition



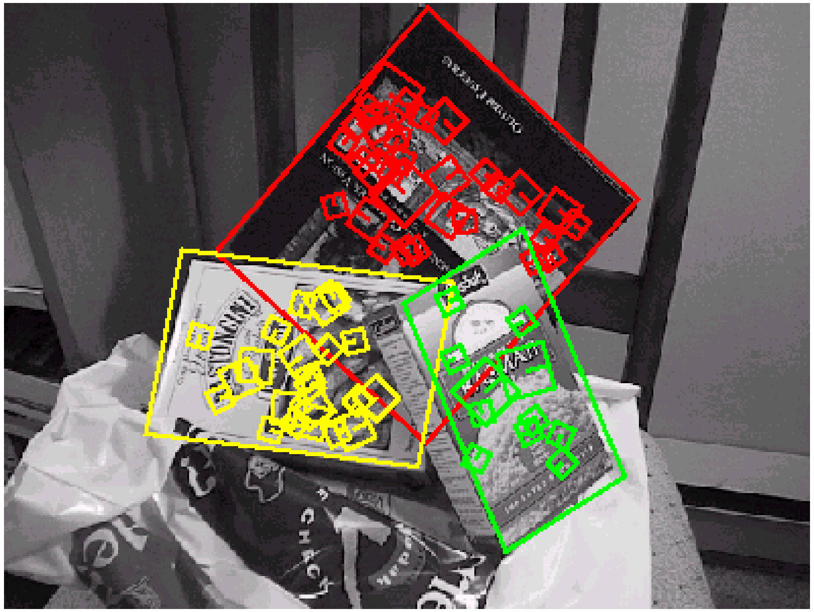
What "Works" Today

- Reading license plates, zip codes, checks
- Fingerprint recognition
- Face detection

[Face priority AE] When a bright part of the face is too bright

What "Works" Today

- Reading license plates, zip codes, checks
- Fingerprint recognition
- Face detection
- Recognition of flat textured objects (CD covers, book covers, etc)



What works today

- Who has the largest database of tagged/labelled faces?
- DeepFace is developed by Facebook
- "A 9-layer deep neural network with over 120 million parameters using several locally connected layers…..Thus we (facebook) trained it on the largest facial dataset to-date, an identity labelled dataset of 4 million facial images belonging to more than 4,000 identities, where each identity has an average of over a thousand samples."
- 97% accuracy -> closely approaching human-level performance.

A simple Object Recognition Pipeline

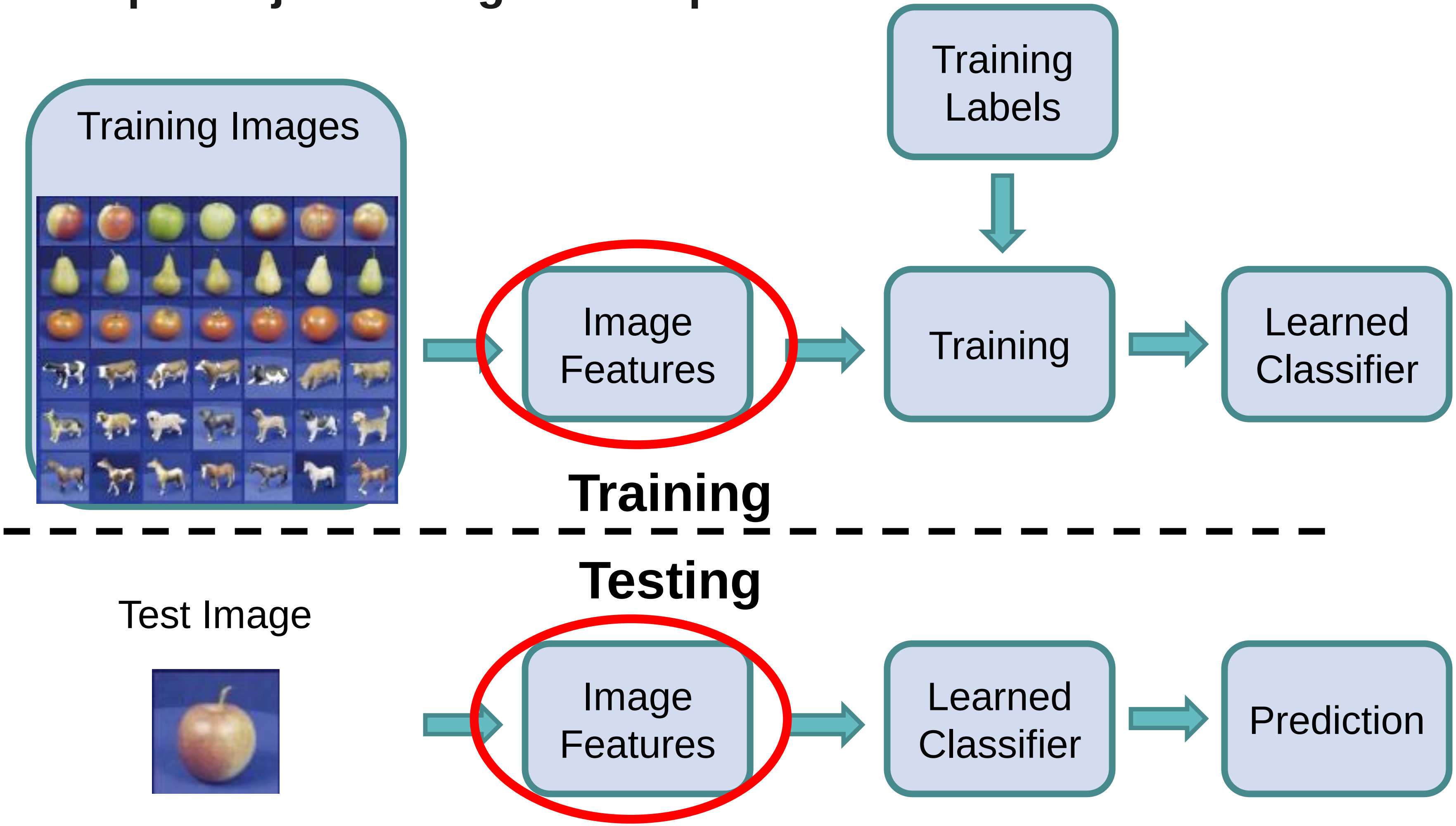


Image Features

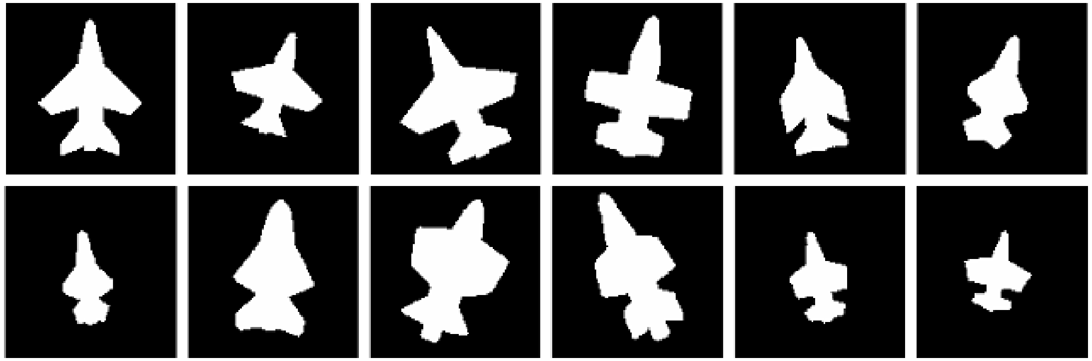
- Two primary characteristics for object recognition:

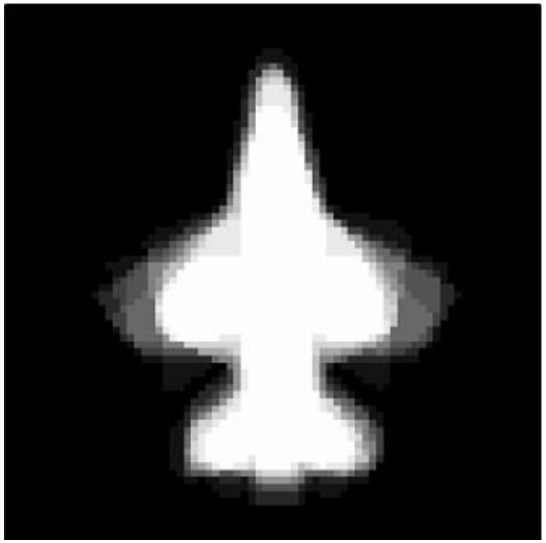
  shape and **appearance**

- Shape can be modeled with **Principal Component Analysis (PCA)**
- **PCA can also model appearance**
- Appearance can be modeled with **Colour Histograms**

Example of Shape Modeling using PCA

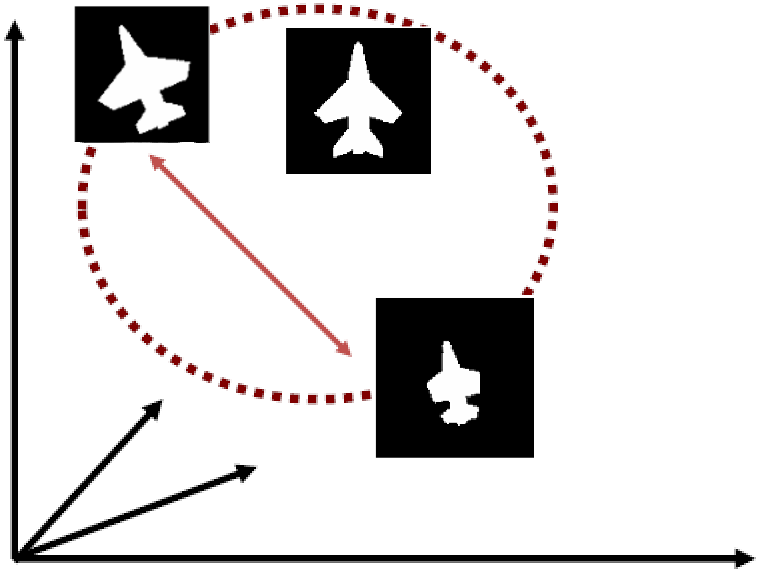
- What is the shape of an aircraft?





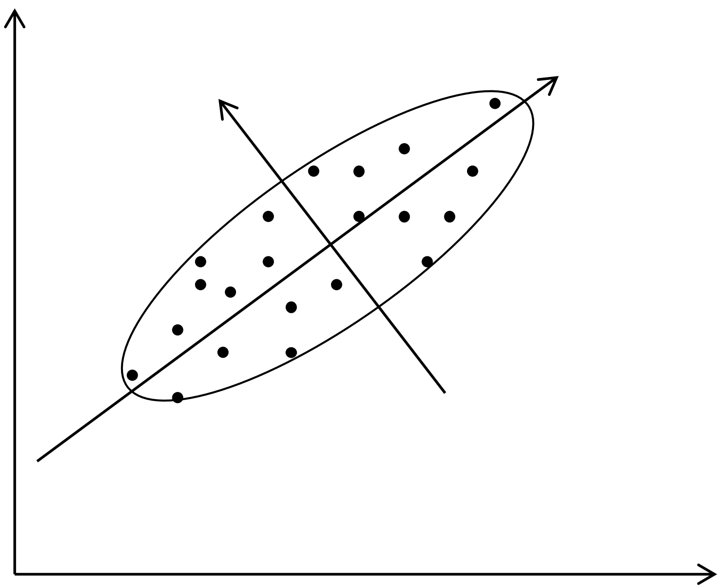
Images as Data Points

- A *N* x *N* pixel image represented as a vector occupies a single point in N^2-dimensional image space.
- Images of particular objects being similar in overall configuration, will not be randomly distributed in this huge image space, but will form *clusters*.
- Therefore, they can be compactly represented and modelled in a low dimensional subspace.



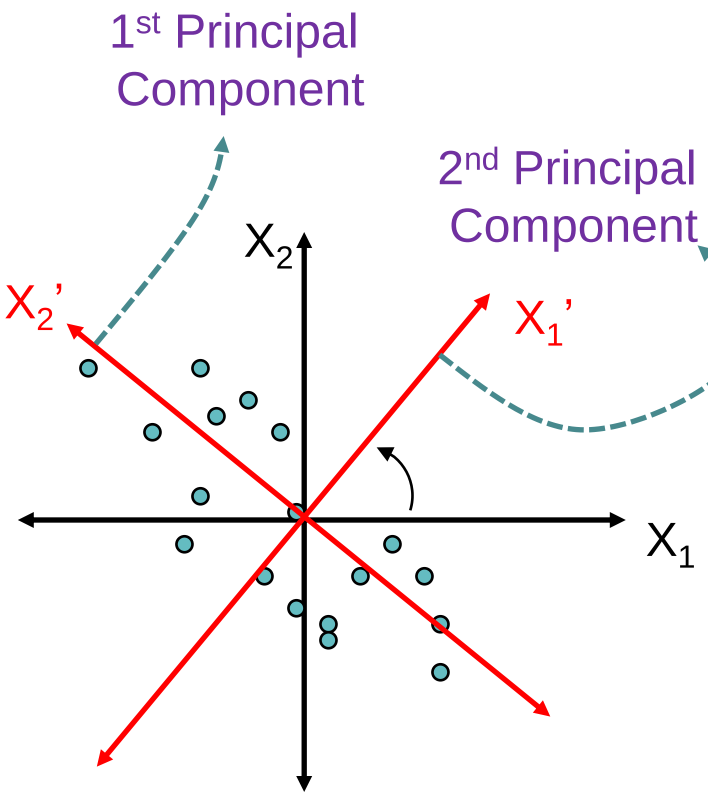
Principal Component Analysis (PCA)

- Calculate vectors that account for the maximum variance of data
  - These vectors are called *Eigen Vectors*.
- Eigen Vectors show the direction of axes a fitted ellipsoid
- Eigen Values show the significance of the corresponding axis.
  - Large value -> more variance
- For high dimensional data, only a few of the Eigen values are significant



Principal Component Analysis (PCA)

- Find Eigen Values and Eigen Vectors
- Chose the highest P Eigen Values
- Form a new coordinate system defined by the significant Eigen vectors
- Project data to the new space (rotate the basis)
- Compressed Data



Principal Component Analysis (Maths)

- Let X be a matrix of the training images (each column is a vectorized image)
- Finds its column mean $\mu = \frac{1}{n}\sum X_i$ (average face)
- Subtract mean from all data $\hat{X} = X - \mu$
- $USV^T = \hat{X}\hat{X}^T$ (singular value decomposition)
- Columns of U are the Eigen Vectors and diagonal of S are the Eigenvalues (sorted in decreasing value)
- Chose P eigenvectors i.e. first P columns of U
- If m is the number of pixels and $m > n$ then use the following trick
- $U_1SV^T = \hat{X}^T\hat{X}$ and $U = \hat{X}U_1$

- Any data sample x can be projected to the PCA space as $U_p^T(x - \mu)$ where U_p contains the top (first) P eigenVectors of U

PCA for Recognition

- U (the Eigenvector matrix) is calculated from training data
- Training data X is projected to the PCA space using U
- Test data is also projected to the same PCA space (same U)
- Nearest neighbor is used for classification
- If the original images were of $m \times m = 50 \times 50 = 2500$ dimension and we chose $P = 20$. The projected images will be only 20 dimensional
- If our training samples $n = 100$, the total possible Eigenvectors with non-zero eigenvalues will always be < 100 (99 at most)

Case Study – Face Recognition

- Milestone methods in face detection / recognition

1. **PCA and Eigenfaces** (Turk & Pentland, 1991)
2. LDA and Fisherfaces (Bellumeur et al. 1997)
3. AdaBoost (Viola & Jones, 2001)
4. Local Binary Patterns (Ahonen et al. 2004)
5. DeepFace (Facebook, 2014)

PCA and Eigenfaces – Training

1. Align training images $x_1, x_2, \dots, x_N$

2. Compute average face $\mu = \frac{1}{N}\Sigma x_i$

3. Compute PCA of the covariance matrices of the difference images
4. Compute training projections $a_1, a_2, \dots, a_N$

PCA and Eigenfaces – Testing

Visalization of Eigenfaces

These are the first 4 eigenfaces (eigenvectors) from a training set of 400 images

1. Take query image y
2. Project y into the Eigenface space $\omega = U_p^T(y - \mu)$
3. Compare projection ω with all training projection a_i
4. Identity of the query image X is chosen as that of the nearest image (i.e. the one with the lowest $||w - a_i||$

Reconstruction using PCA

- Only selecting the top P eigenfaces reduces the dimensionality.
- Fewer eigenfaces result in more information loss, and hence less discrimination between faces.

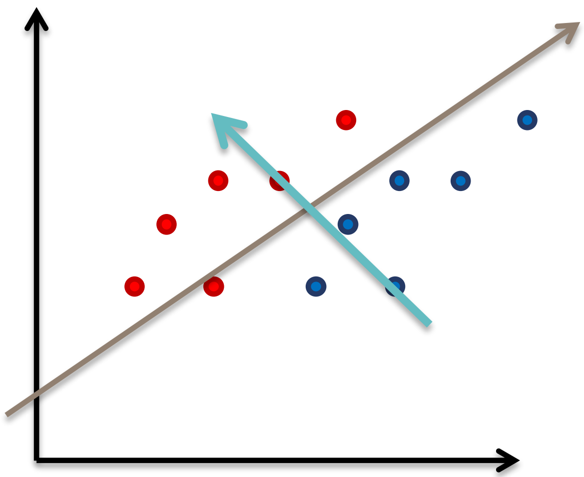
P = 4

P = 200

P = 400

PCA Final Note

- PCA finds directions of maximum variance of the data.
- This may not separate classes at all.
- Basic PCA is also sensitive to noise and outliers (read other variants e.g. Robust PCA).
- Linear Discriminant Analysis LDA finds the direction along which between class distance is maximum.



- Sometimes PCA is followed by LDA to combine the advantages of both.
- Eigen eyes, eigen nose, eigen X your imagination is the only limination.

Importance of Colors in Object Detection / Recognition

Colour Histogram

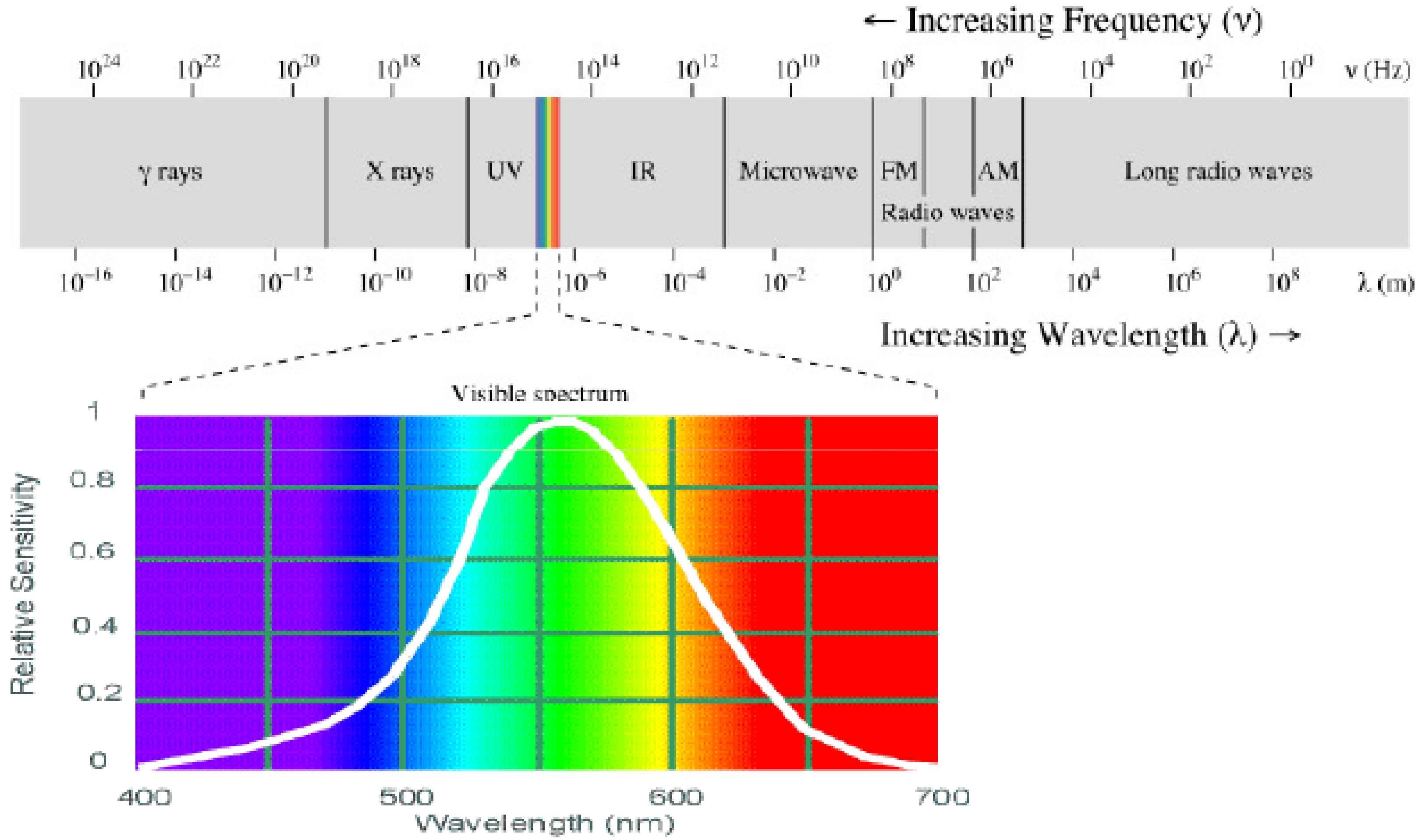
- Colour stays constant under geometric transformations

- Colour is a local feature
  - It is defined for each pixel
  - It is robust to partial occlusion

- Idea:
  - can use object colours directly for recognition, or
  - better – use statistics of object colours

- Colour histogram is a type of appearance features

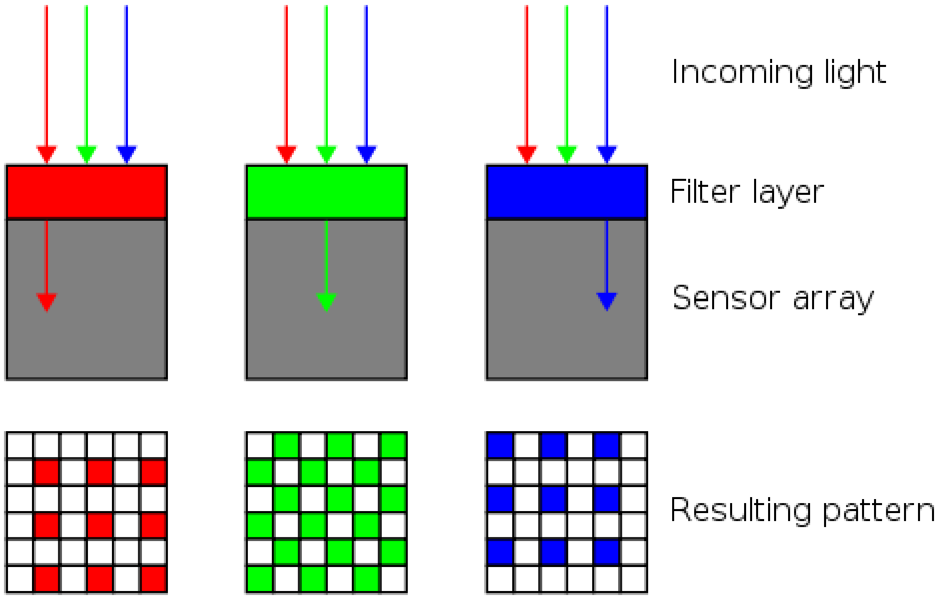
Colour Sensing



Human Luminance Sensitivity Function

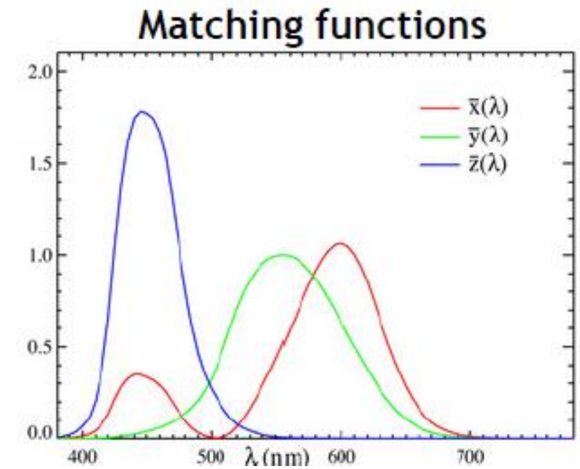
Colour Spaces – RGB

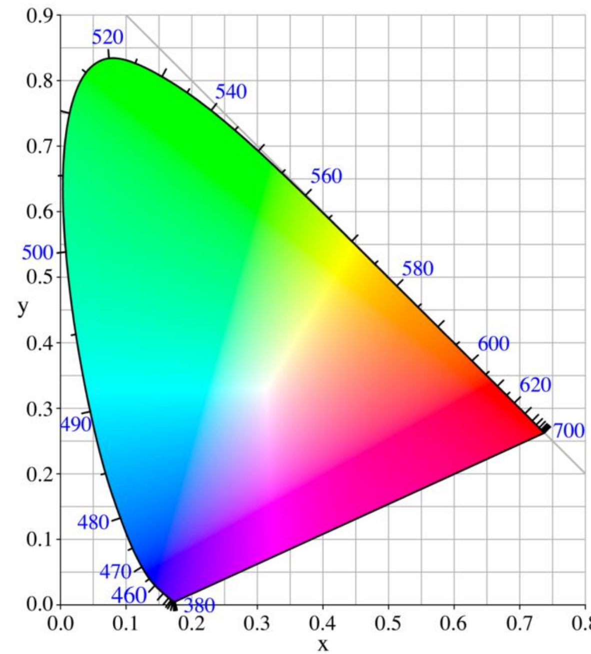
- Primaries are monochromatic lights
  - for camera: Bayer filter pattern
  - for monitors; they correspond to the 3 types of phosphors



Colour Spaces – CIE XYZ

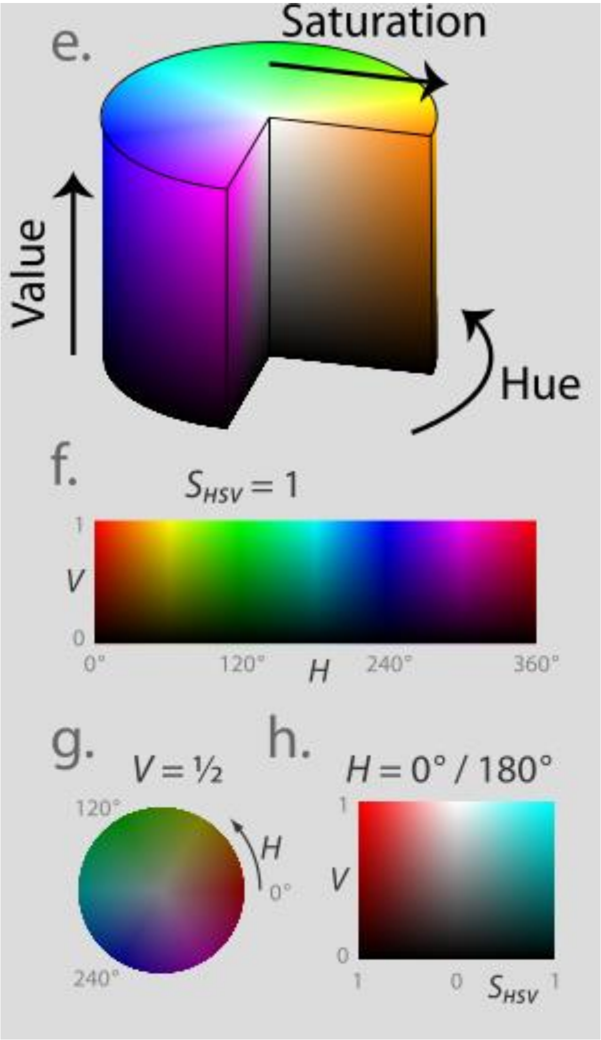
- Links physical pure colours (i.e wavelengths) in the electromagnetic visible spectrum and physiological perceived colours in human colour vision.
- Primaries $X, Y,$ and Z are imaginary, but the matching functions are everywhere positive
- 2D Visualization: illustrates the x and y values where $x = X/(X+Y+Z)$ and $y = Y/(X+Y+Z)$. The value of $z = 1 - x - y$.





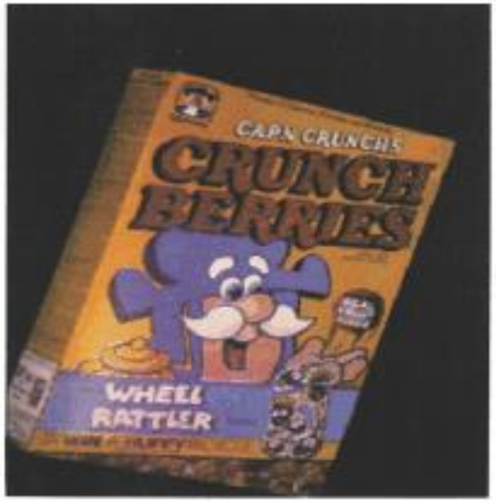
Colour Spaces – HSV

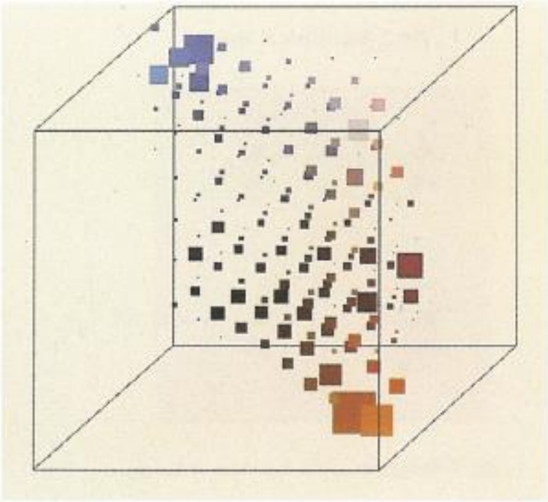
- **HSV** - Hue, Saturation, Value (Brightness)
  - Nonlinear – reflects topology of colours by coding hue as an angle
  - Matlab functions: `hsv2rgb`, `rgb2hsv`



Colour Histograms

- Colour histograms are colour statistics
  - Here, RGB as an example
  - Given: tristimulus R, G, B for each pixel
  - Compute a 3D histogram
  - $h(R, G, B) = \#(\text{pixels with colour } (R, G, B))$





Colour Normalization

- One component of the 3D colour space is intensity
  - If a colour vector is multiplied by a scalar, the intensity changes but not the colour itself.
  - This means colours can be normalized by the intensity.
  - Note: intensity is given by $I = (R + G + B)/3$
  - Chromatic representation:

$$r = \frac{R}{R+G+B} \qquad g = \frac{G}{R+G+B} \qquad b = \frac{B}{R+G+B}$$

Since $r + g + b = 1$, only 2 parameters are needed to represent colour (knowing r and g, we can deduce $b = 1 - r - g$).

⇒ Can compute colour histogram using r, g, and b instead.

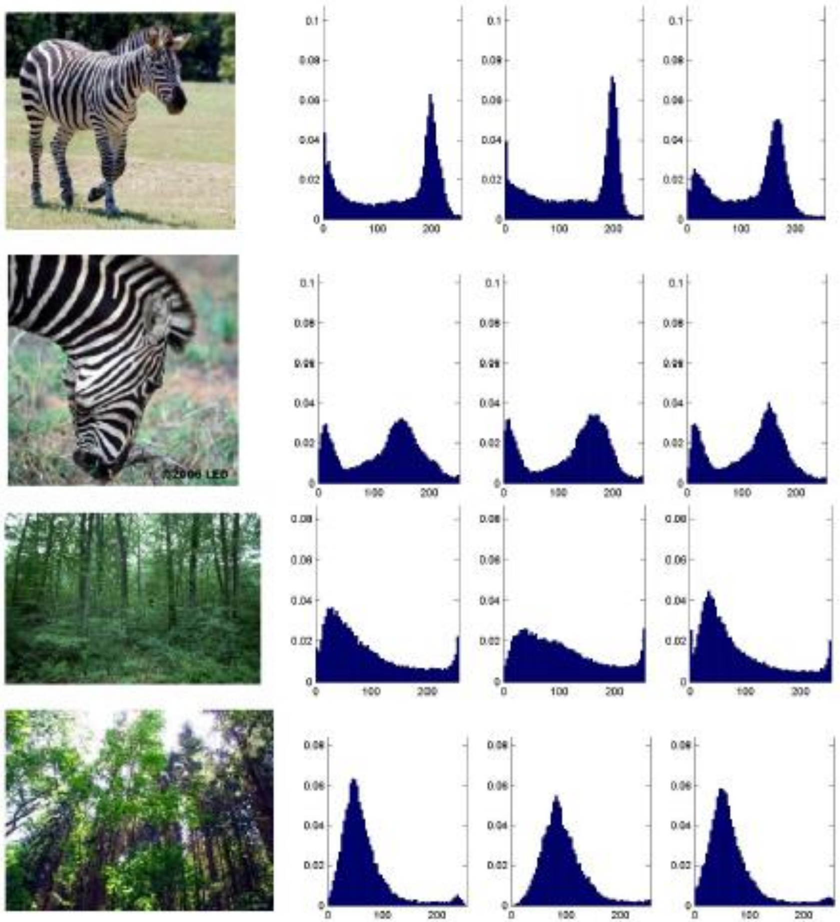
Object Recognition based on Colour Histograms

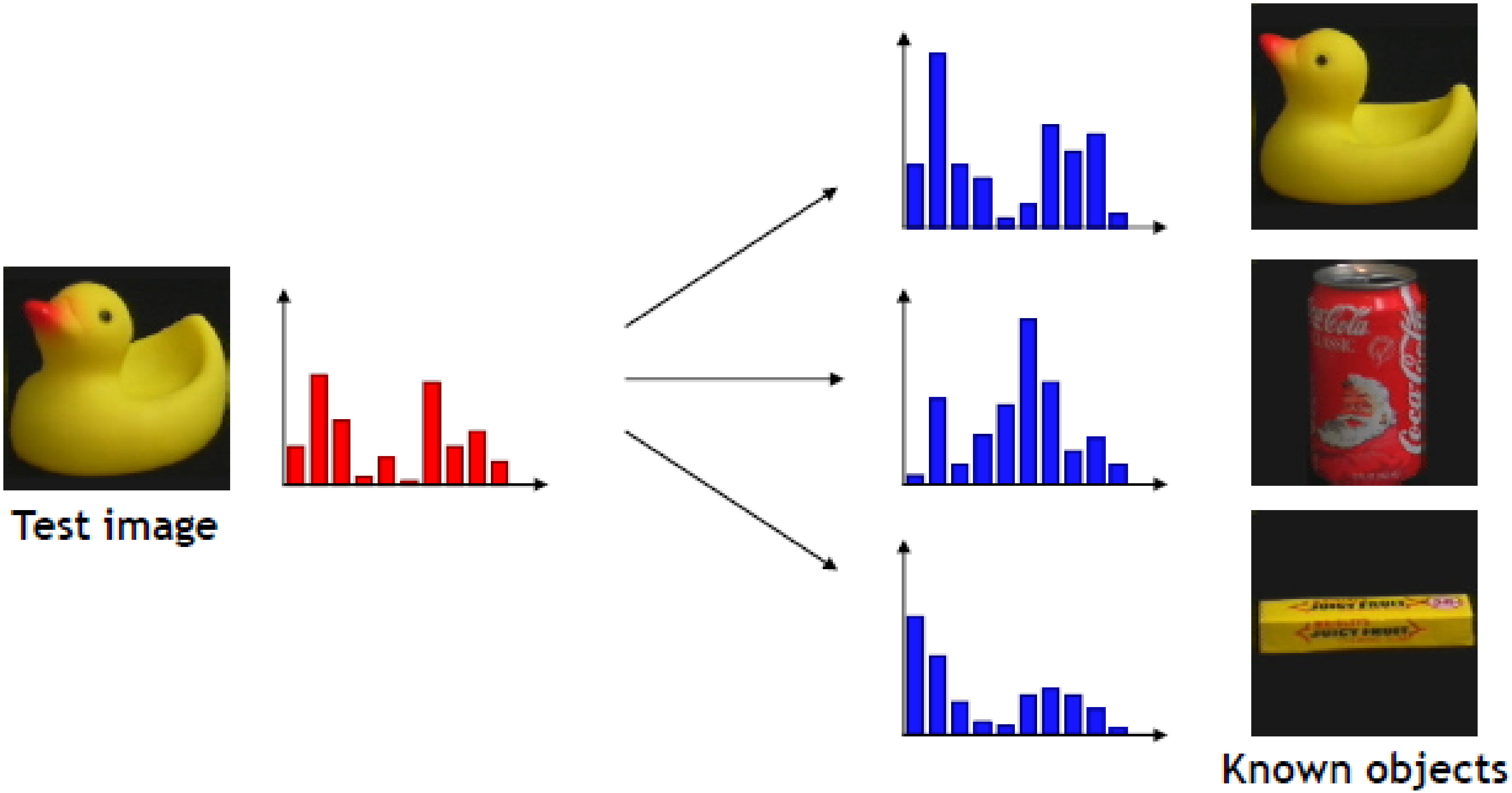
- Proposed by Swain and Ballard (1991).
- Objects are identified by matching a colour histogram from an image region with a colour histogram from a sample of the object.
- Technique has been shown to work remarkably robust to
  - changes in object's orientation
  - changes of scale of the object
  - partial occlusion, and
  - changes of viewing position and direction.

Object Recognition based on Colour Histograms

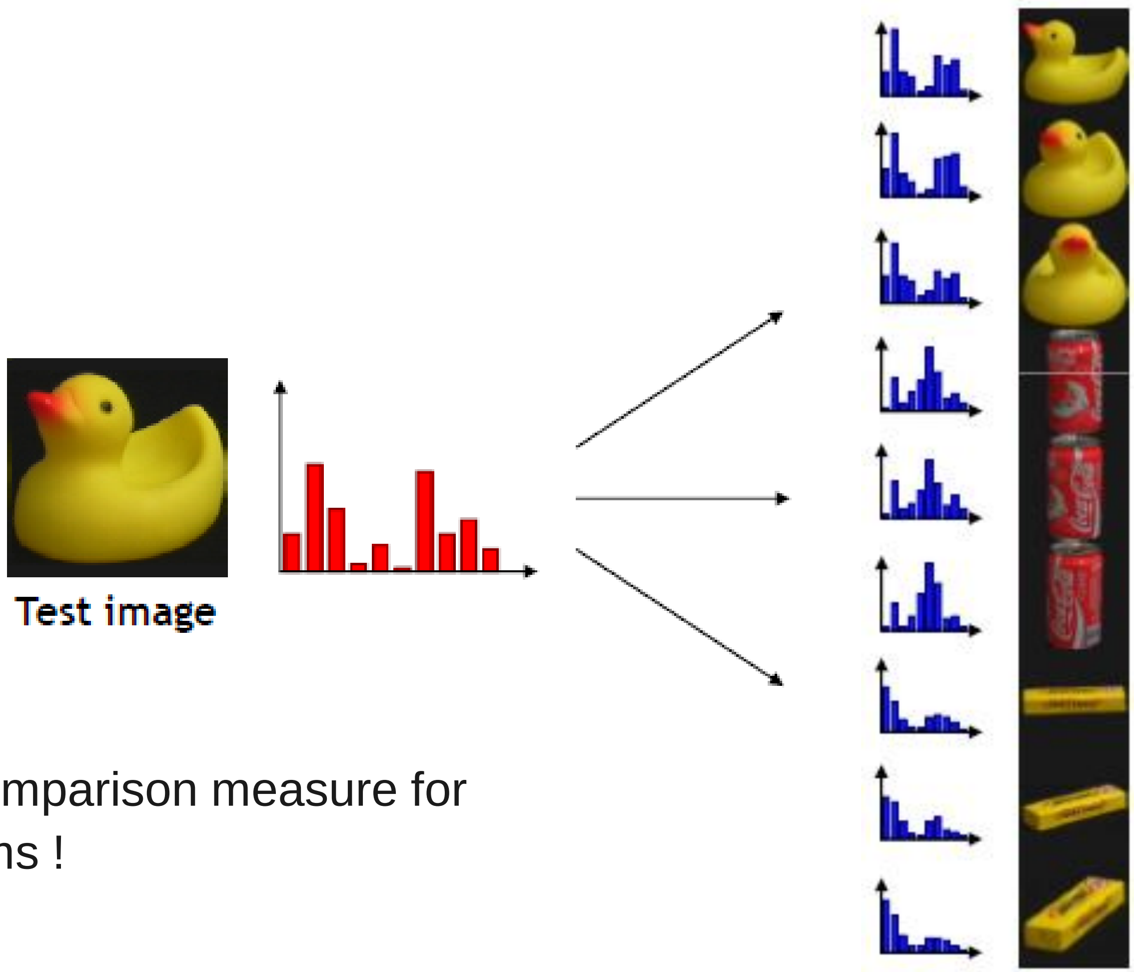
Colour histograms

- are discrete approximation of the colour distribution of an image.
- contain no spatial information ⇒ invariant to translation, scale, and rotation




Test image
Known objects

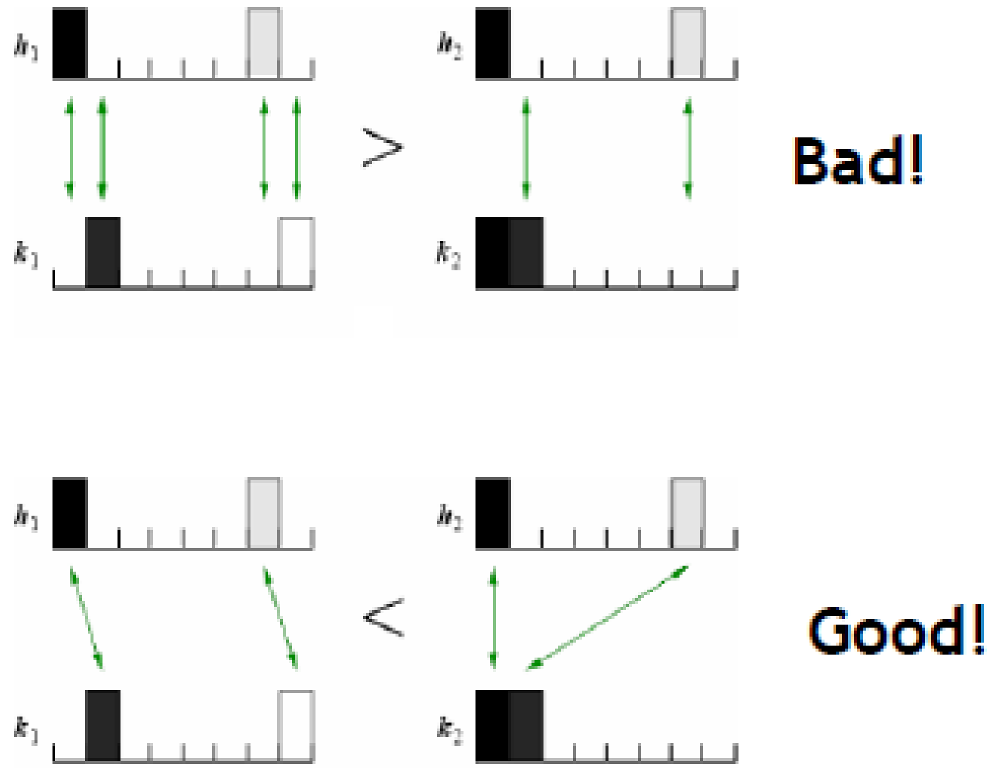
Histogram Comparison with Multiple Training Views



⇒ Need a good comparison measure for colour histograms !

What is a Good Comparison Measure?

- How to define matching cost?

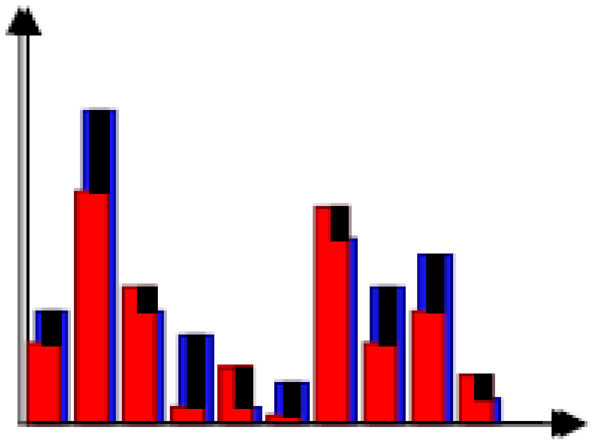


Comparison Measures

Euclidean distance (L_2 norm)

$$d(\boldsymbol{q}, \boldsymbol{v}) = \sum_i (q_i - v_i)^2$$

- Motivation of the Euclidean distance:
  - Focuses on the differences between the histograms.
  - Interpretation: distance in the feature space.
  - Range: $[0, \infty)$.
  - All cells are weighted equally.
  - Not very robust to outliers !



Comparison Measures (Cont.)

Chi-Square distance:

$$d(\boldsymbol{q},\boldsymbol{v}) = \sum_i \frac{(q_i - v_i)^2}{q_i + v_i}$$

- Motivation of the χ^2 distance:
  - Statistical background
  - Test if two distributions are different.
  - Possible to compute a significance score.
  - Range: $[0,\infty)$.
  - Cells are not weighted equally !
  - More robust to outliers than the Euclidean distance, if the histograms contain enough observations…

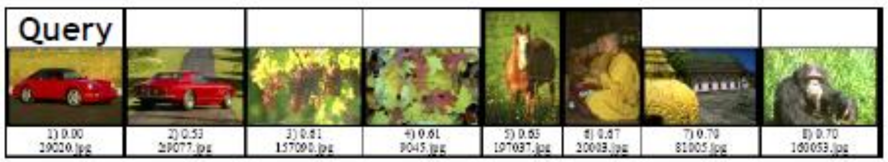
Chi-Square distance:

$$d(\boldsymbol{q},\boldsymbol{v}) = \sum_i \frac{(q_i - v_i)^2}{q_i + v_i}$$

Motivation of the χ^2 distance:
- Statistical background
- Test if two distributions are different.
- Possible to compute a significance score.
- Range: $[0,\infty)$.
- Cells are not weighted equally !
- More robust to outliers than the Euclidean distance, if the histograms contain enough observations…

Comparison for Image Retrieval

- The image retrieval problem concerns the retrieval of those images in a database that best match a query image.



L2 distance

Jeffrey divergence

χ^2 distance

Earth Movers Distance

Histogram Comparison

- Which measure is the best?
  - It depends on the application
  - Euclidean distance is often not robust enough.
  - Generally, χ^2 distance gives good performance for histograms
  - KL/Jeffreys divergence works well sometimes, but is expensive
  - EMD is the most powerful, but also very expensive.

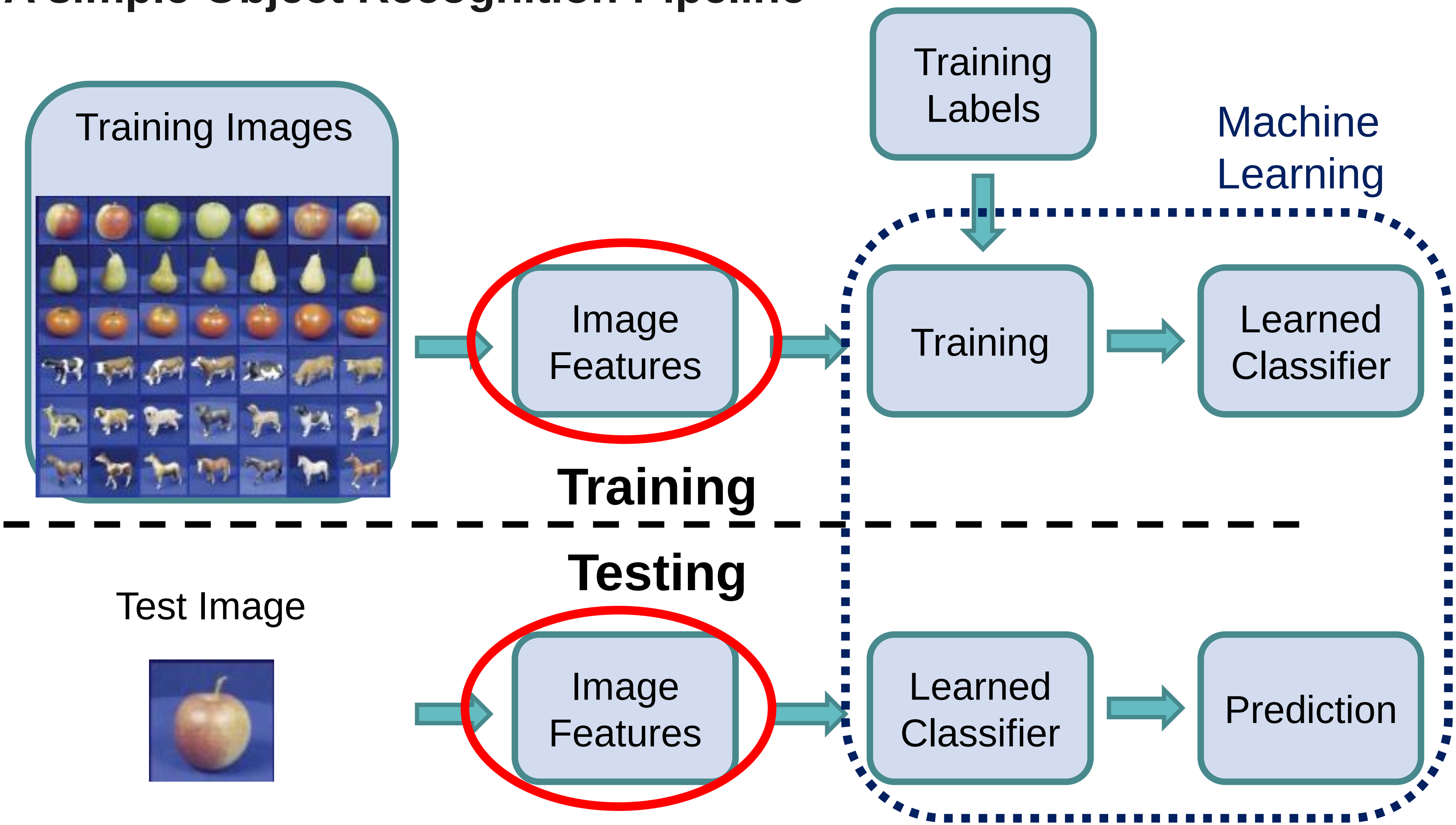
Object Recognition Using Histograms – Summary

- Simple algorithm

1. Build a set of histograms $H = \{\boldsymbol{h}_i\}$ for each known object.
   - More exactly, for each view of each object.
2. Build a histogram $\boldsymbol{h}_t$ for the test image.
3. Compare $\boldsymbol{h}_t$ with each $\boldsymbol{h}_i \in H$ using a suitable histogram comparison measure.
4. Select the object with the best matching score;
   or reject the test image if no object is similar enough.

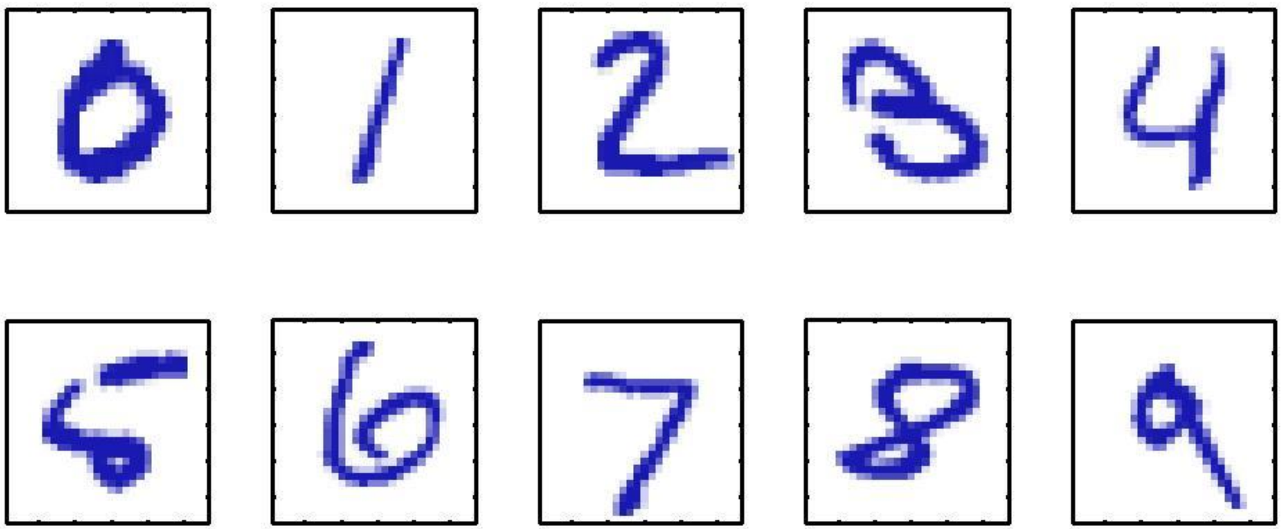
This is known as the "**nearest-neighbour**" strategy.

A simple Object Recognition Pipeline



Goal of Machine Learning

- Consider a 28 x 28 pixel image
- Represented by a 784 dimensional vector ***x***
- Goal: build a machine that takes the vector ***x*** as input and produces the identity of digit 0,…,9 as the output



The Machine Learning Framework

- **Training data** consists of *data samples* and the *target vectors*
- **Learning / Training:** Machine takes training data and automatically learns mapping from data samples to target vectors

$$y = f(x)$$

Output Label

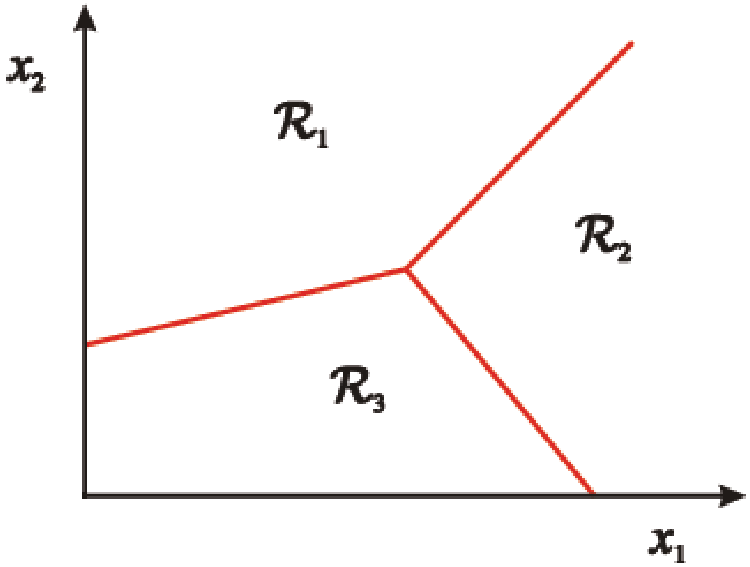
Image Feature

Prediction Function

- **Test data**
  - Target vectors are concealed from the machine
  - Machine predicts the target vectors based on previously learned model
  - Accuracy can be evaluated by comparing the predicted vectors to the actual vectors

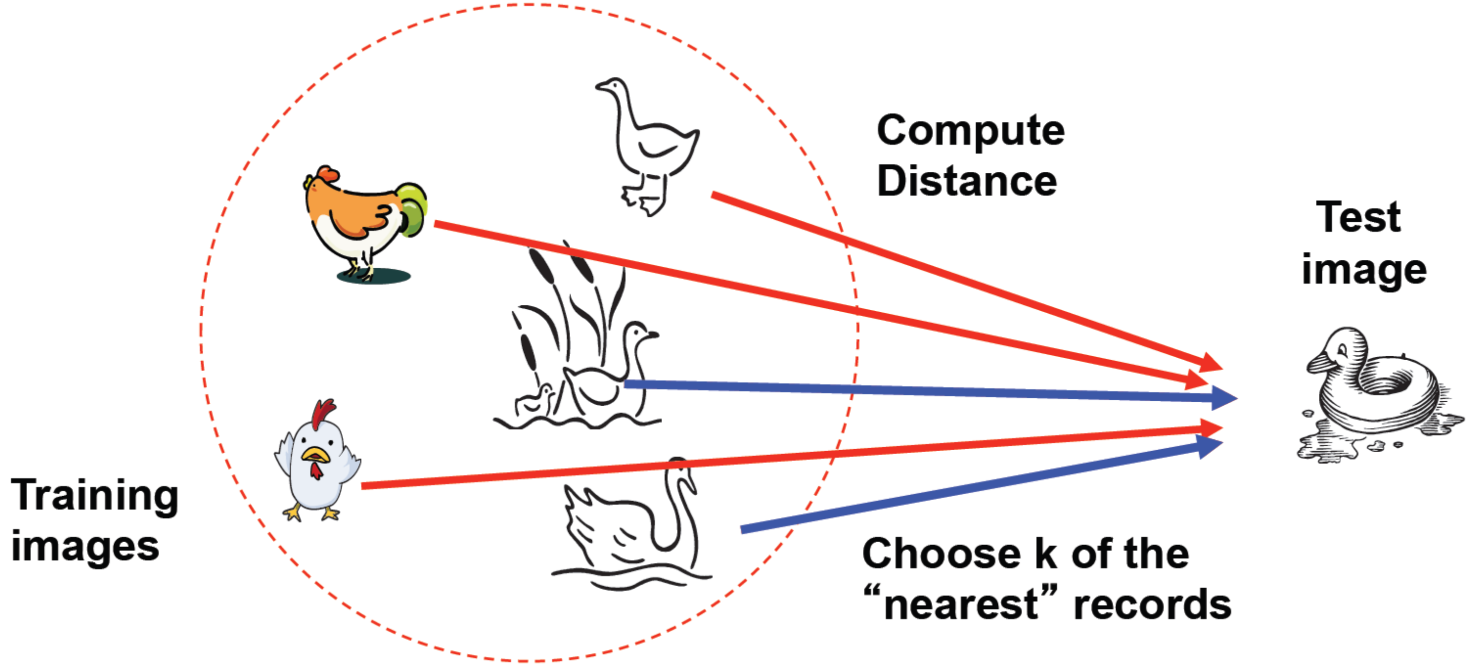
Classification

- Assign input vector to one of two or more classes
- Any decision rule divides input space into *decision regions* separated by *decision boundaries*



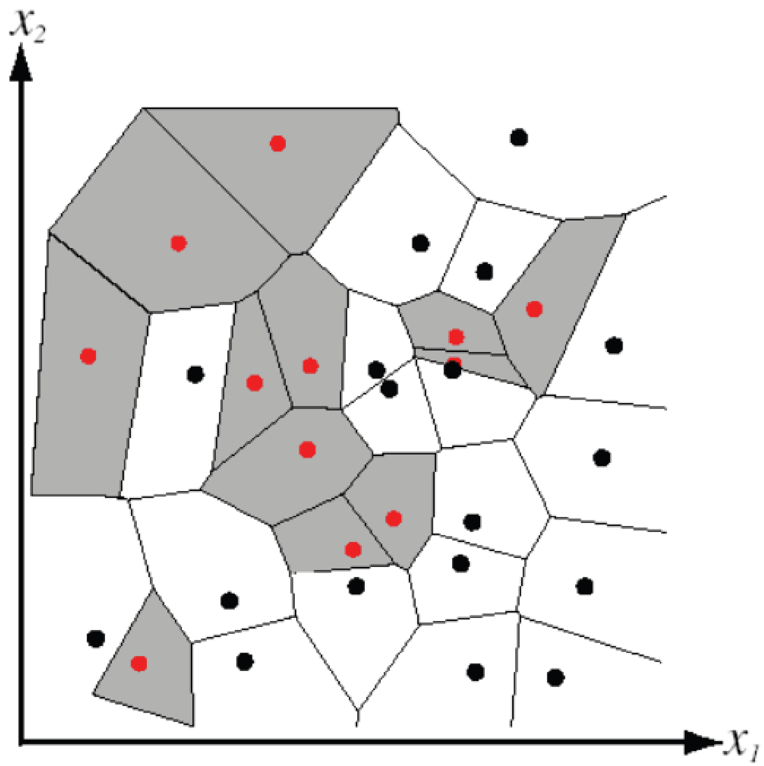
Nearest Neighbour Classifier

- Assign label of nearest training data point to each test data point



Nearest Neighbour Classifier

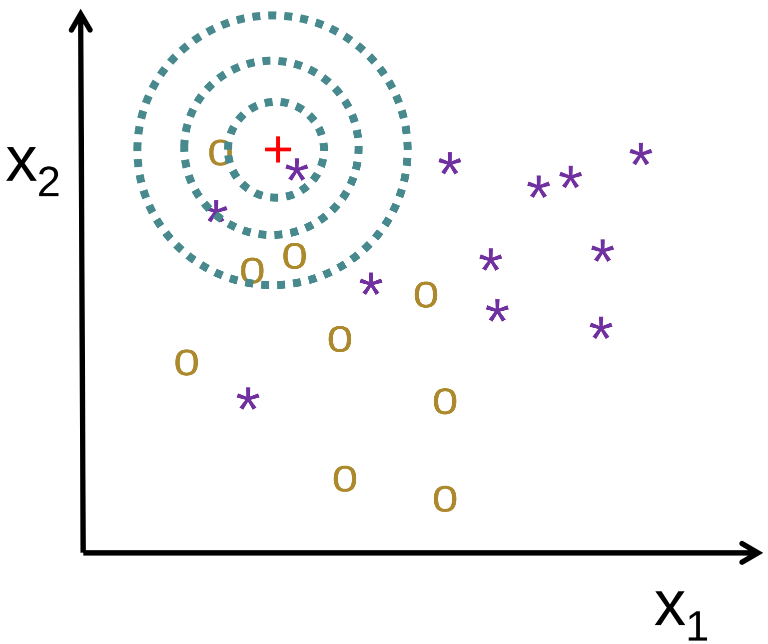
Partitioning of feature space for two-category 2D data using 1-nearest-neighbour



K-nearest-neighbour

- Distance measure – Euclidean

$$D(X,Y) = \sqrt{\sum_{i=1}^{D} (x_i, y_i)^2}$$



- 1-nearest-neighbour

$$f(+) = *$$

- 3-nearest-neighbour

$$f(+) = *$$

- 5-nearest-neighbour

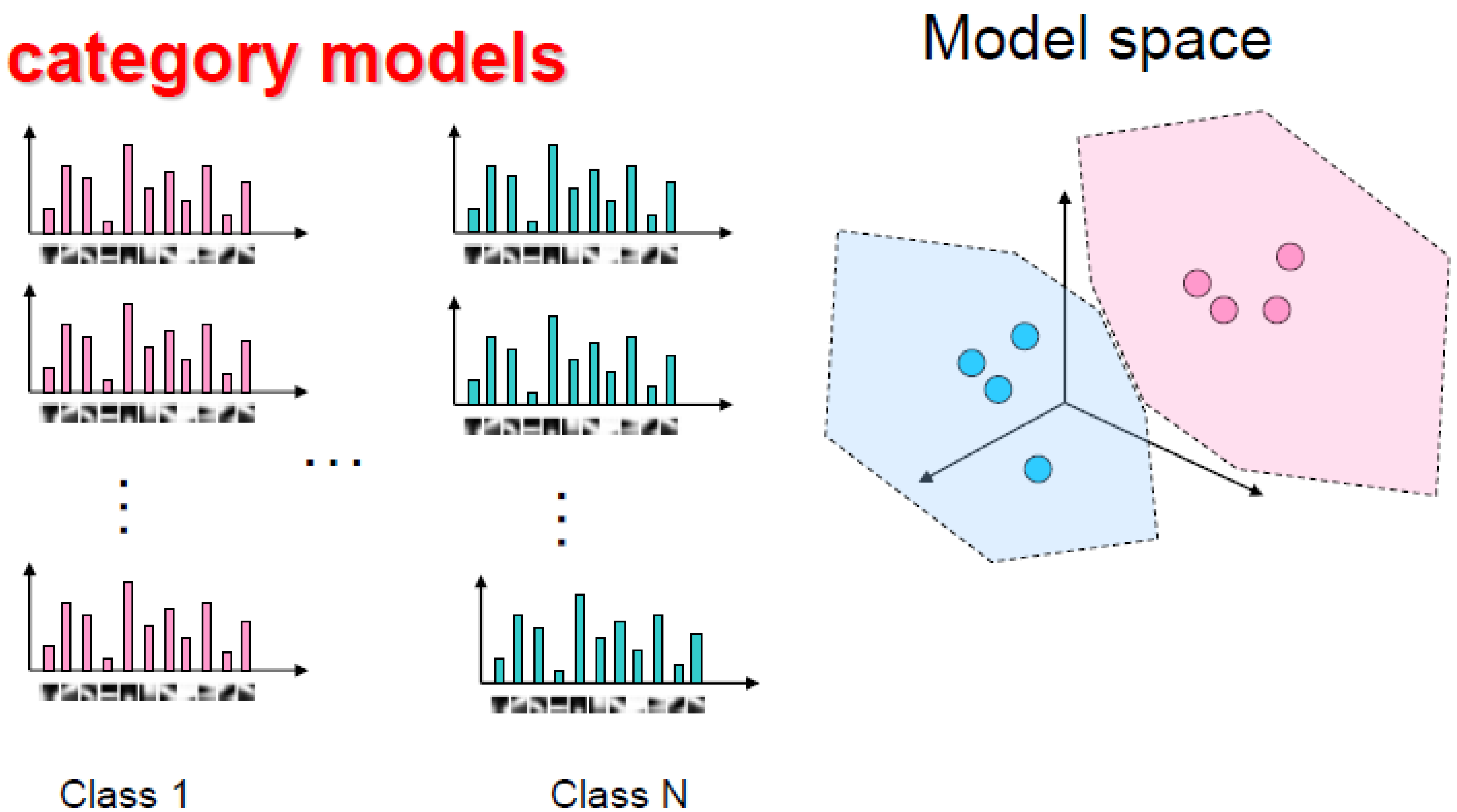
$$f(+) = \mathrm{o}$$

K-NN Practical Matters

- Choosing the value of k
  - If too small, sensitive to noise points
  - If too large, neighbourhood may include points from other classes
  - Solution: cross-validation
- Can produce counter-intuitive results
  - Each feature may have a different scale
  - Solution: normalize each feature to zero mean, unit variance
- Curse of dimensionality
  - Solution: no good solution exists so far

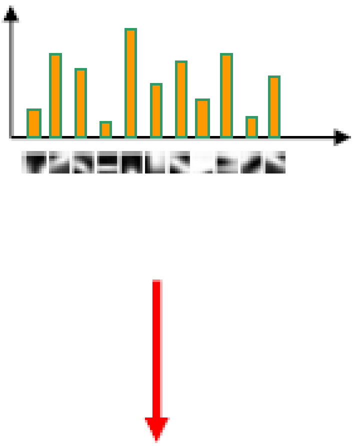
- This classifier works well provided there are **lots of training data** and the **distance function is good**.

Discriminative Classifiers



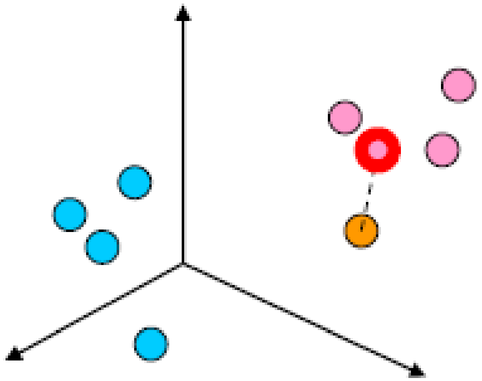
Nearest Neighbours Classifier

Query image



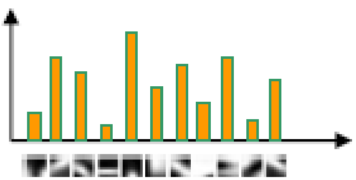
Winning class: pink

Model space

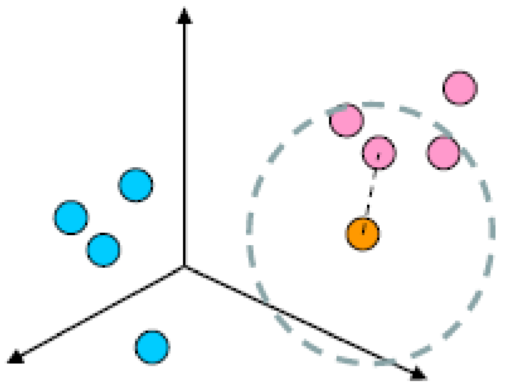


K-Nearest Neighbours Classifier

Query image



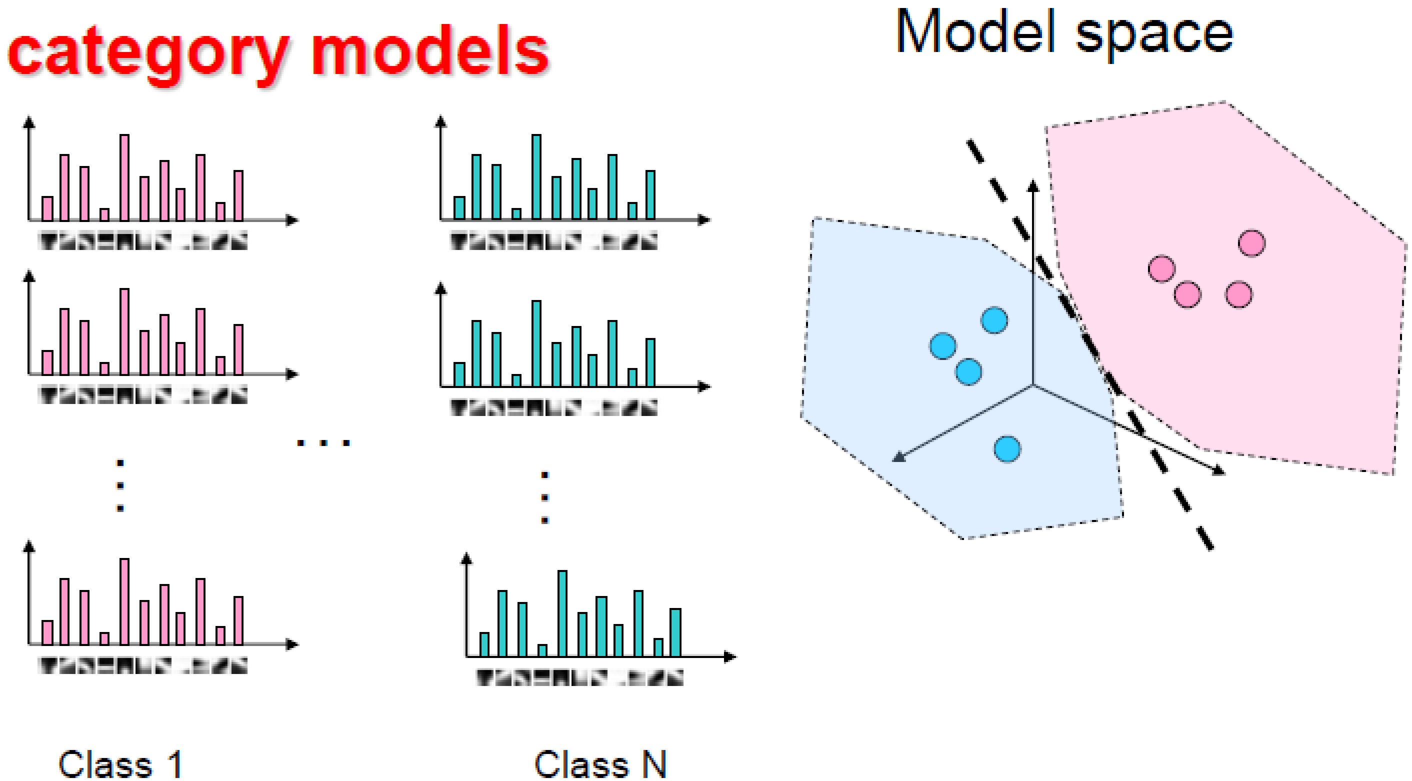
Model space



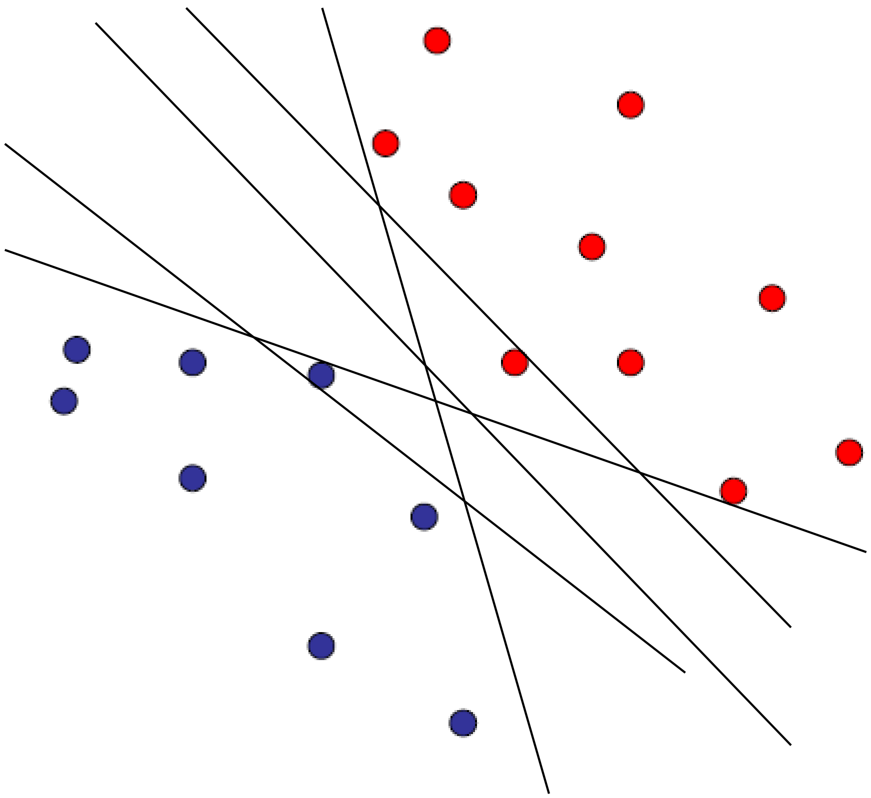
Winning class: pink

Linear Classifiers

- Support Vector Machines: find the hyper-planes (if the features are linearly separable) that separate these classes in the model space



Linear Classifiers

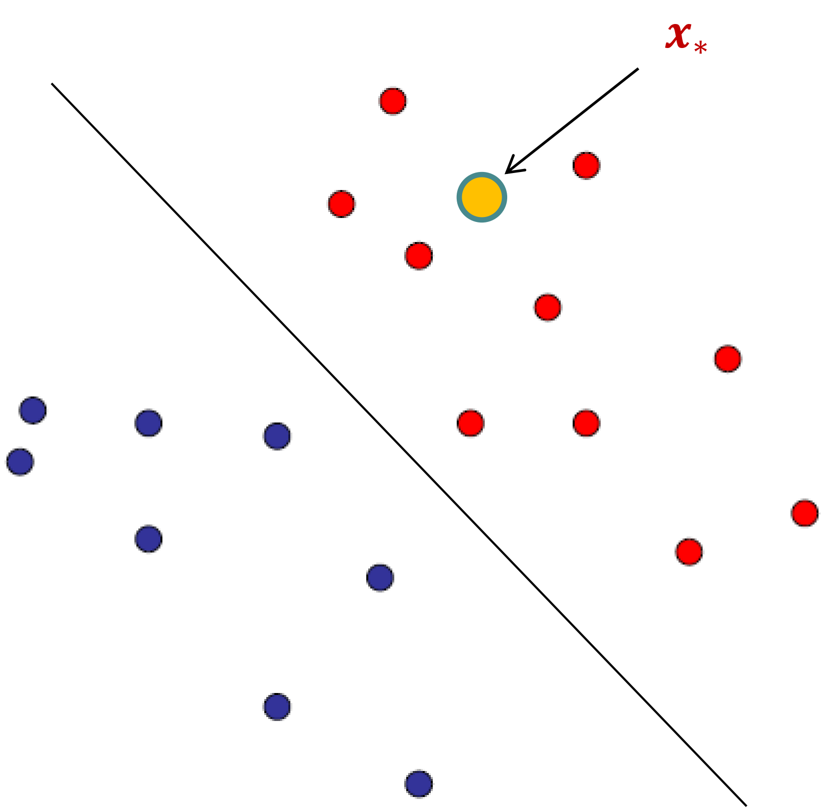


Suppose that the points are in 2D.
Points in class 1 have label $y_i = +1$;
points in class 2 have label $y_i = -1$.

Given $\{(\boldsymbol{x}_i, y_i), \text{where } y_i \in \{-1, +1\}\}$,
for $i = 1, \ldots, N$. Here $\boldsymbol{x}_i \in \mathbb{R}^2$.
Find $\boldsymbol{w}$ and b such that

$$\boldsymbol{w}^T \boldsymbol{x}_i + b \geq 1 \quad \text{if } y_i = +1$$
$$\boldsymbol{w}^T \boldsymbol{x}_i + b \leq -1 \quad \text{if } y_i = -1$$

Linear Classifiers

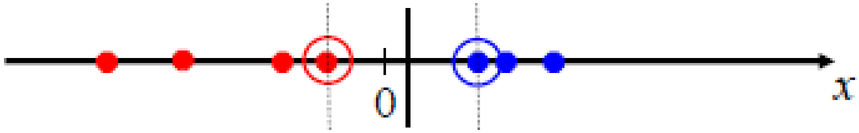


Once we have learned $\boldsymbol{w}$ and b, we can do classification on any given test point $\boldsymbol{x}_*$. This is known as the **testing stage**.

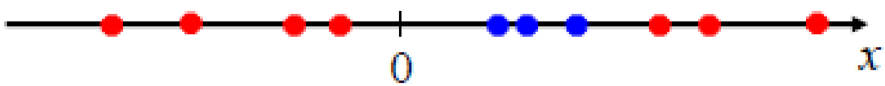
If $\boldsymbol{w}^T\boldsymbol{x}_* + b \geq +1$ then
    classify $\boldsymbol{x}_*$ into class 1
else
    classify $\boldsymbol{x}_*$ into class 2

Nonlinear SVMs

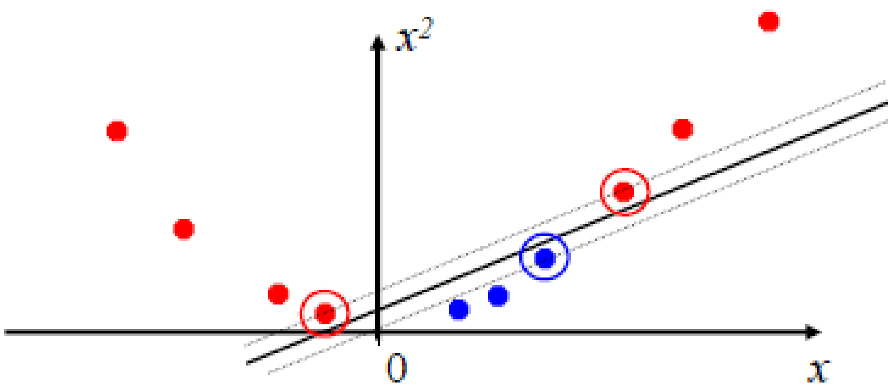
- The linear SVM works out great when the data are linearly separable. E.g. the 1D case below:



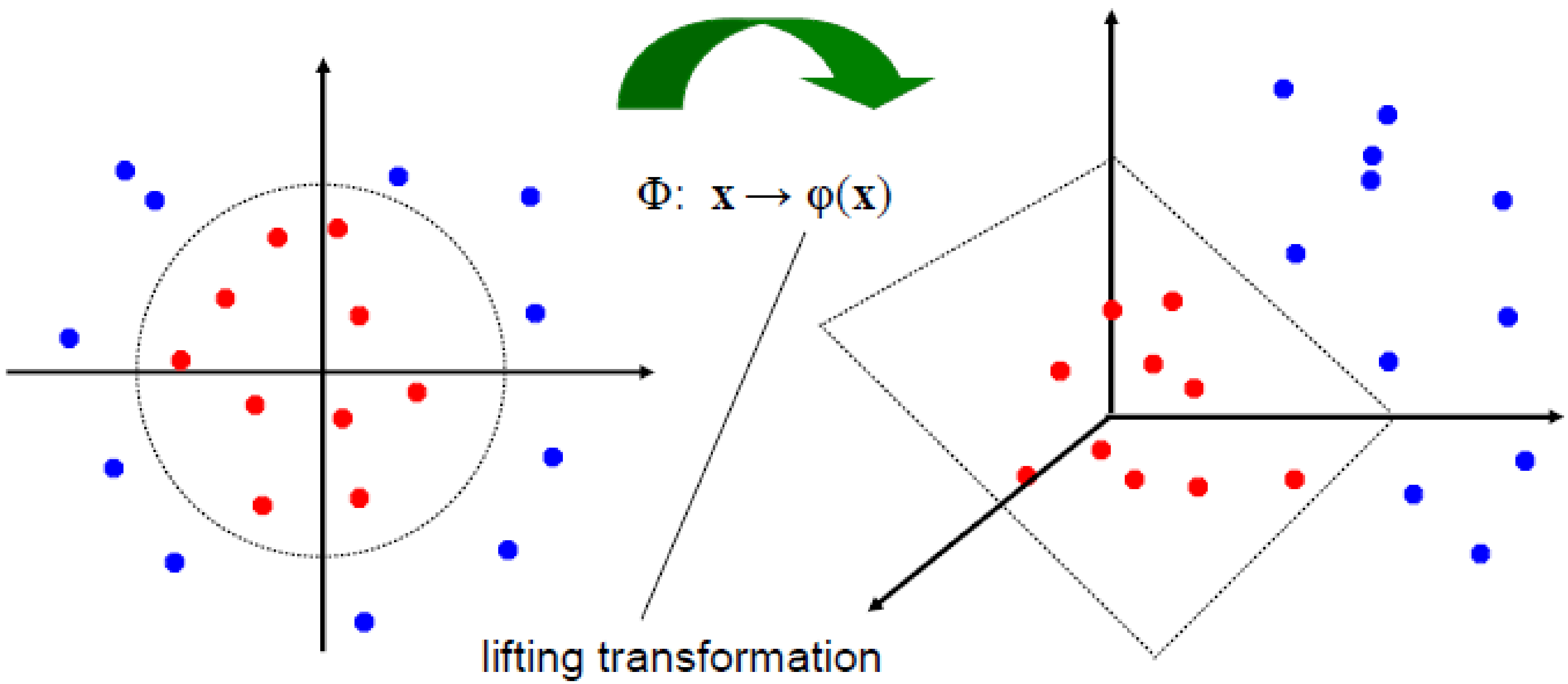
- But what is the data are more complicated? Like



- We can map the data to a higher-dimensional space:



Nonlinear SVMs



- We use a **lifting transformation** Φ to transform the feature vectors to a higher dimensional space.

Summary

- Challenges in Object Recognition
- A Simple Object Recognition Pipeline
- Principal Component Analysis
- Colour Histograms
- Discriminative Classifiers (k-NN and SVM)

Acknowledgements: The slides are based on previous lectures by A/Prof Du Huynh and Prof Peter Koveski. Other material has been taken from Wikipedia, computer vision textbook by Forsyth & Ponce, and Stanford's Computer Vision course by Fei-Fei Li.