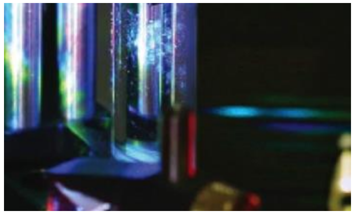
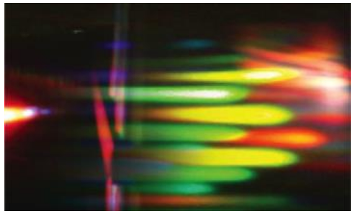
Imperial College London





COMP70058 Computer Vision

# Lecture 12 – Object Recognition


Stamatia (Matina) Giannarou, PhD
The Hamlyn Centre for Robotic Surgery

stamatia.giannarou@imperial.ac.uk


The Hamlyn Centre for Robotic Surgery

# Object Recognition

- Object Recognition
- Bag of Features
  - Origins
  - Representing the Visual Vocabulary
  - Classification
- Other Object Recognition Techniques

Adapted from slides by S. Lazebnik, A. Torralba, L. Fei-Fei, D. Lowe, C. Szurka

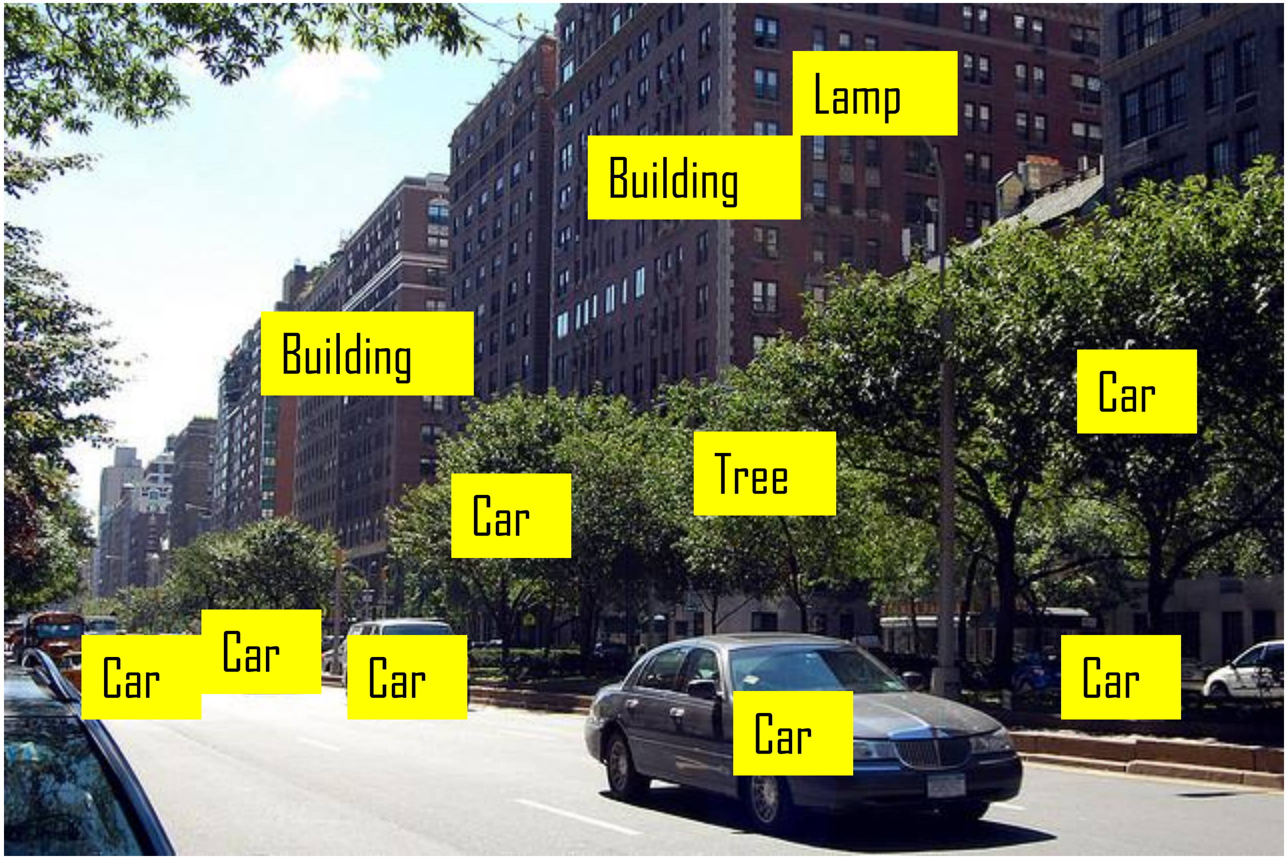
# The Problem of Object Recognition

# Verification: Is it a Car?

# Detection: Are There Cars?

# Identification: Is this a New York taxi?

# Object Categorisation



# Scene and Context Categorisation

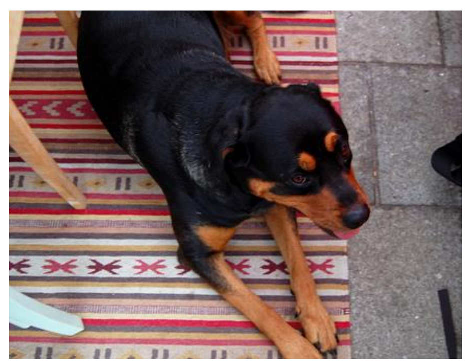
- Street? Beach? Jungle? Room? Night-time?

# Challenges in Object Recognition

- Variability within objects
  - View point changes (camera position)
  - Illumination
  - Occlusions
  - Internal camera parameters
  - Scale
  - Deformation

- Variability within class
  - The example of dogs on the previous slide
  - Too many classes

# Challenges in Object Recognition



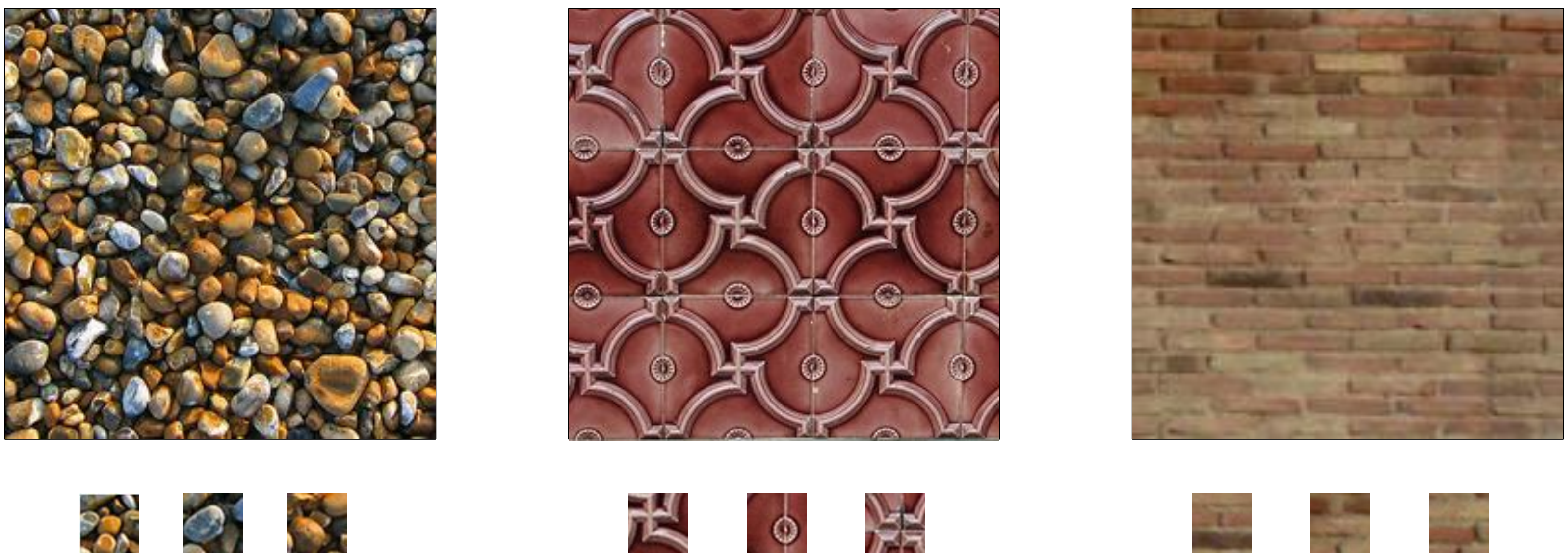
# Object Recognition

- Any computer vision method for object recognition must address the following properties:
  - Representation
    - How will an object category be presented?
    - What classification scheme will be used?
  - Learning
    - How will the classifier be learned?
    - (assuming there’s training data)
  - Recognition
    - How will the classifier be used on new data?

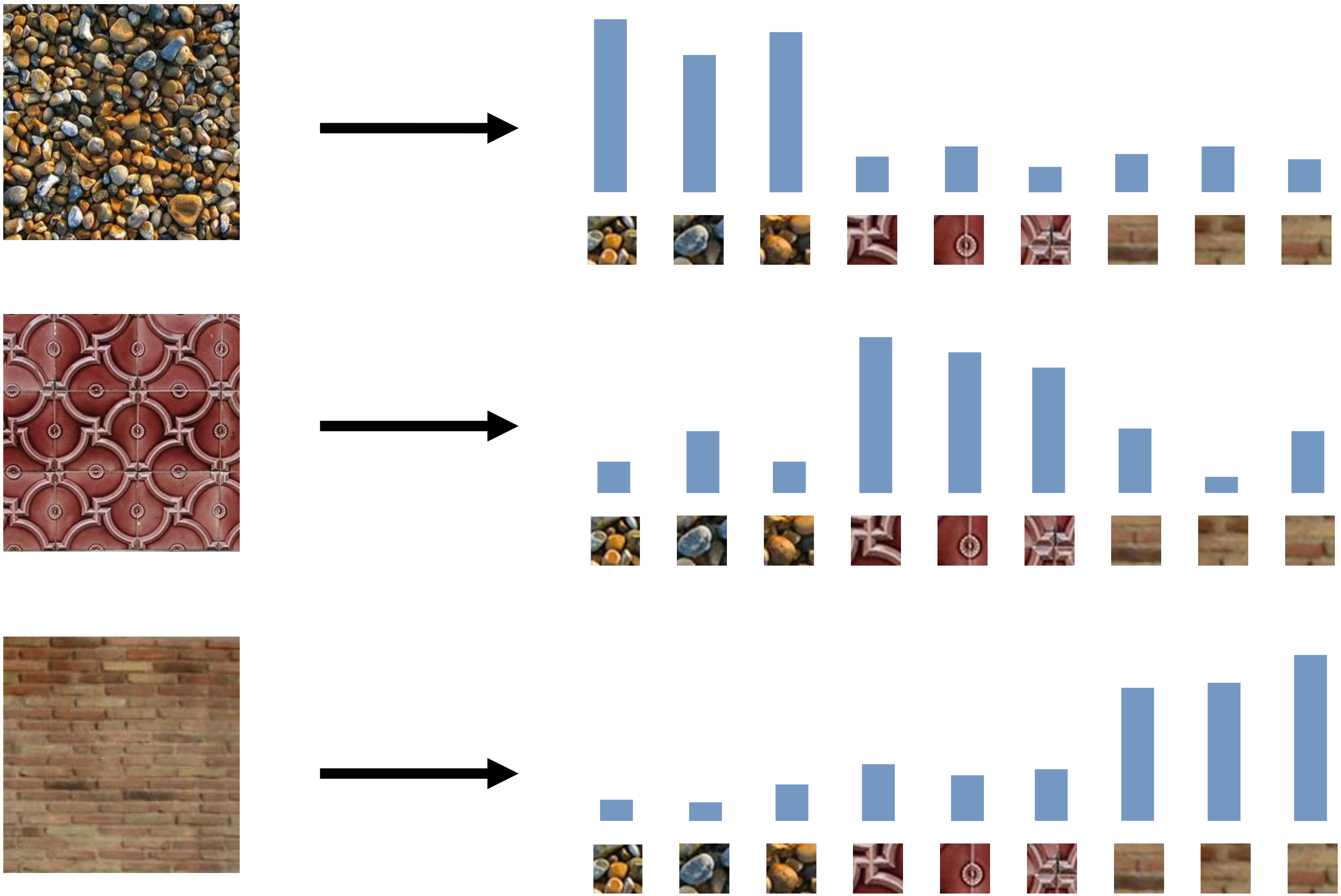
- We’ll present one such method – Bag of Words/Bag of Features

# Origins: Texture Recognition

- As we've seen before, textures are made up of repeating basic elements (or textons)
- For stochastic textures, the identity of the textons matters and not their spatial arrangement

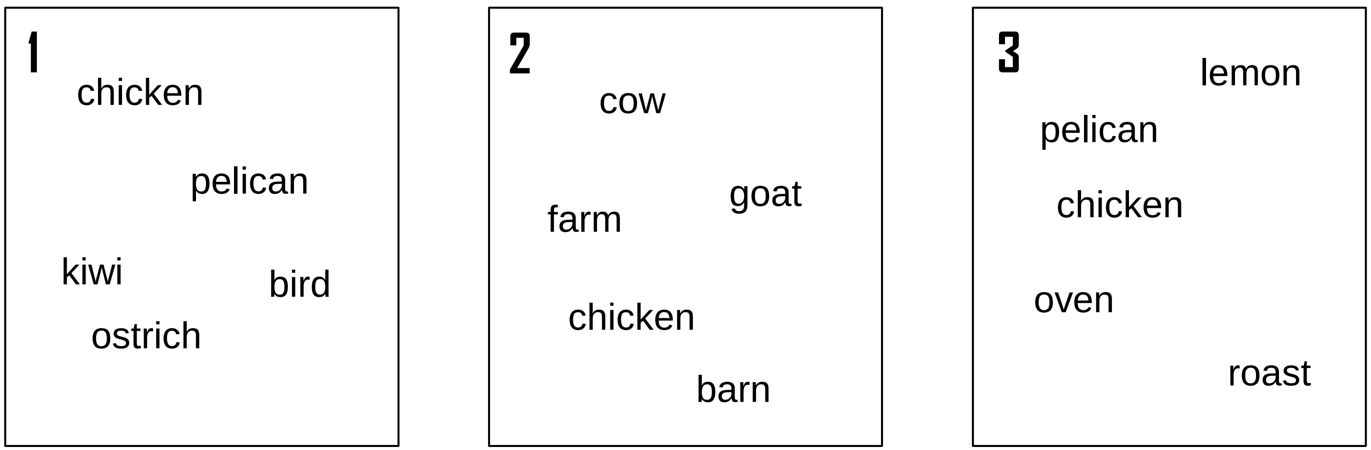


# Texture Recognition



# Origins: Bag of Words

- Text is represented as an unordered collection of words
- The frequency of occurrence of each word is treated as a feature for training a classifier
- If we had the following documents:




- We can examine the histograms of these words:

| | | | |
|---|---|---|---|
| chicken | 4 | 6 | 5 |
| roast | 0 | 1 | 4 |
| farm | 1 | 7 | 1 |
| bird | 6 | 1 | 2 |

# Bag of Words for Document Classification

- What results is a histogram of words

- Classification can be performed on the histogram

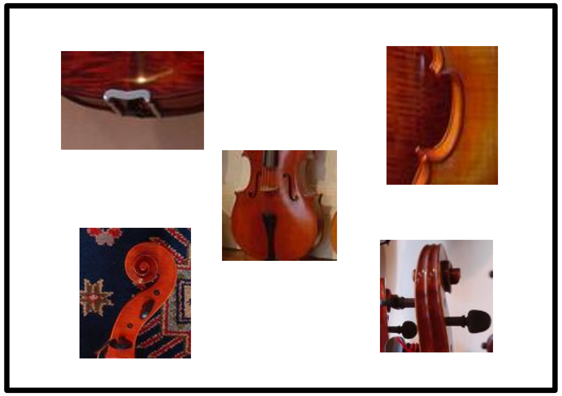
- For the example on the previous slide, it may be possible to classify the documents as about:
  1. Birds
  2. Farms
  3. Recipes

- The method has been applied successfully for email filtering
  - Is it spam or is it ham?

# Bag of Features for Image Classification

1. Extract features
2. Learn the ‘visual vocabulary’ (i.e. The ‘dictionary’)
3. Quantise the features using the visual vocabulary
4. Represent images by frequencies of ‘visual words’

Bicycle

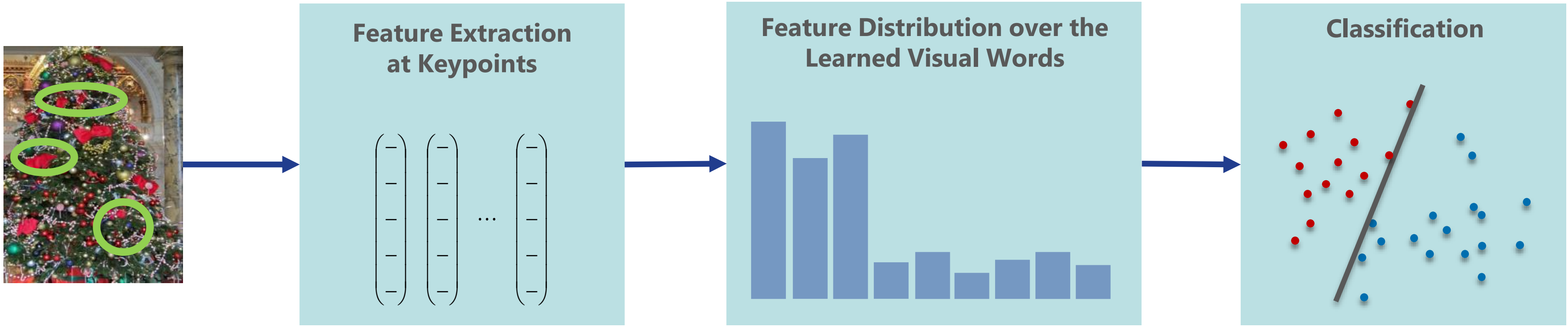


Violin

Christmas Tree

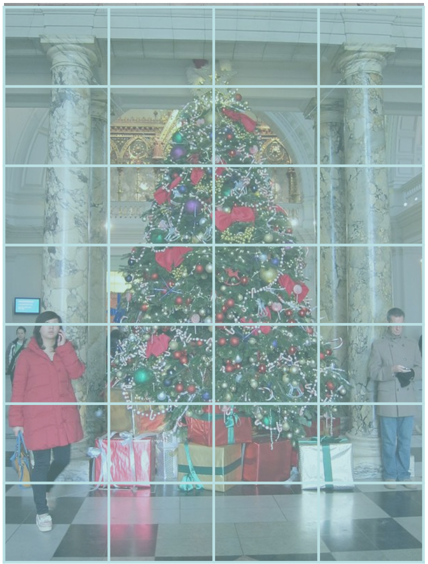
# Bag of Features for Image Classification

1. Extract features
2. Learn the 'visual vocabulary' (i.e. The 'dictionary')
3. Quantise the features using the visual vocabulary
4. Represent images by frequencies of 'visual words'



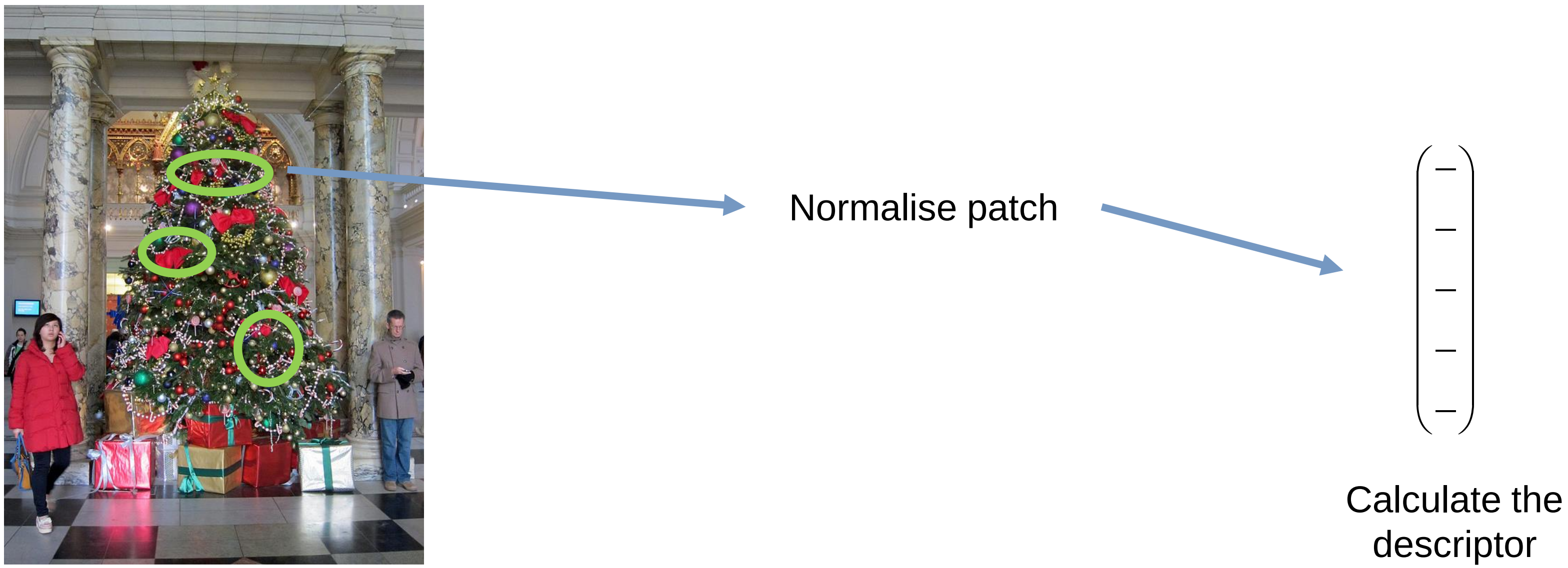
# Feature Detection and Representation

- You saw in a previous lecture (Image Sequence Processing, Part 1) how to extract corner or SIFT features
- Other methods include:
  - Regular grid – dividing the image using a regular grid
  - Interest point detectors
  - Random sampling
  - Segmentation-based patches



# Feature Detection and Representation

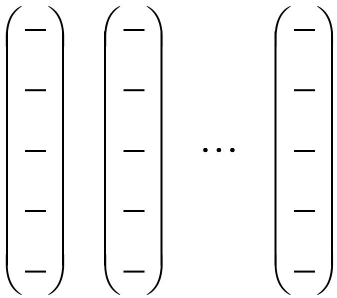
- You saw in a previous lecture (Image Sequence Processing, Part 1) how to extract corner or SIFT features
- Other methods include:
  - Regular grid – dividing the image using a regular grid
  - Interest point detectors
  - Random sampling
  - Segmentation-based patches

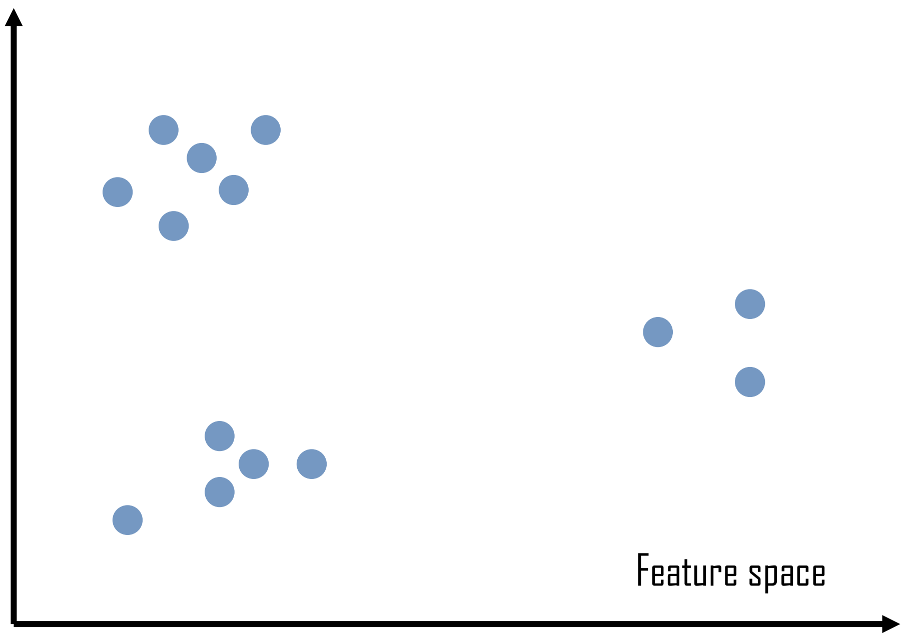



Detect patches

# Learning the Visual Vocabulary

- Like Bag of Words, we need a histogram of ‘words’
- Each descriptor needs to be converted into a ‘word’





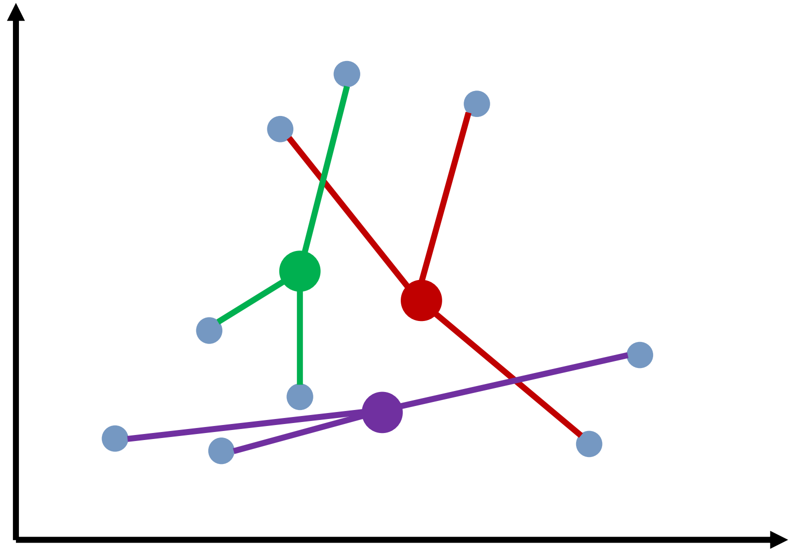

- One method to define ‘words’ is by using a clustering algorithm such as k-means

# K-Means Clustering

- One method for performing data clustering
  - K is the number of clusters required and is a user input
  - Minimise the sum of squared Euclidean distances between the points $x_i$ and their nearest cluster centres $m_k$

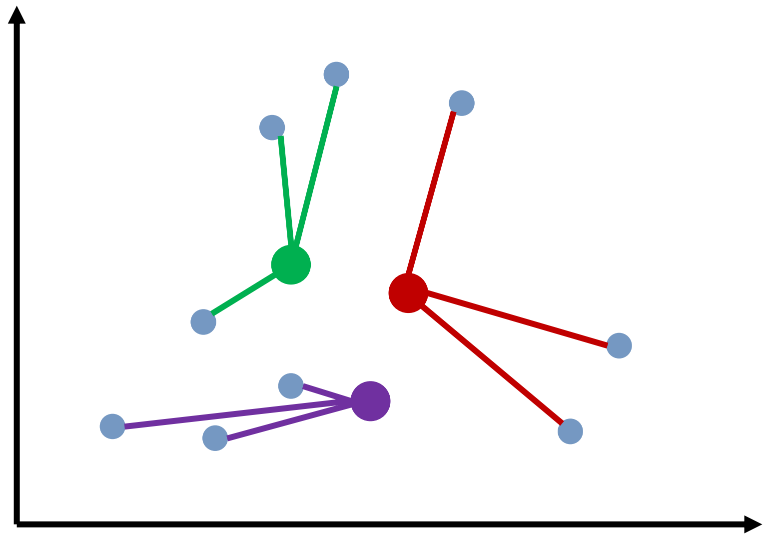
$$D(X, M) = \sum_{k} \sum_{i} (x_i - m_k)^2$$

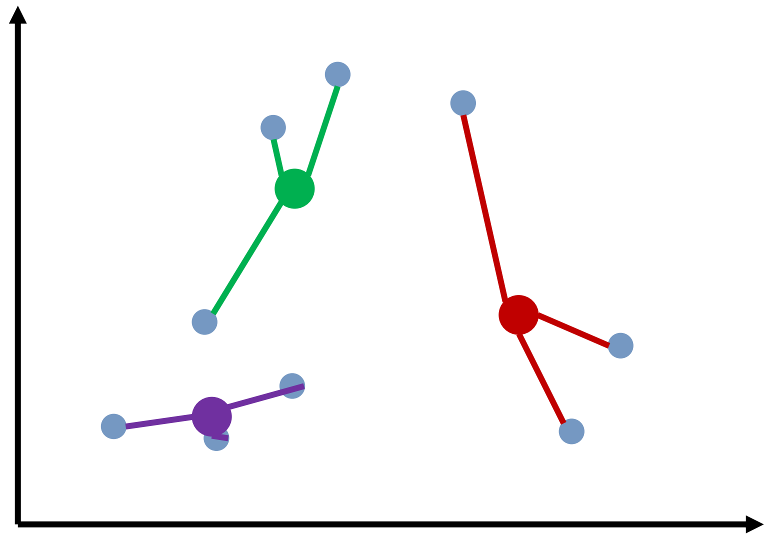
1. The data points are assigned randomly into k groups and the cluster centroids are calculated
   - The initial cluster centroids may also be user defined



# K-Means Clustering

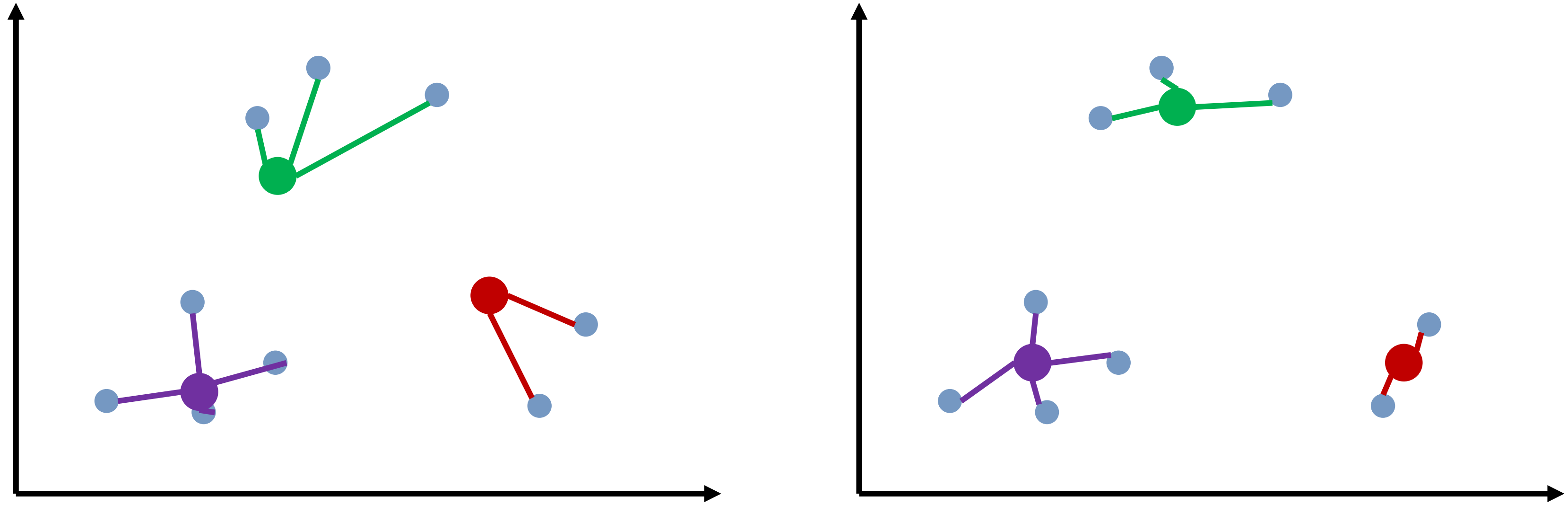
2. The points are reclassified by minimising the distance between each point and the previous cluster centroids
    1. This is the minimum distance algorithm
3. Recalculate the new class means





# K-Means Clustering

4. Repeat steps 2 and 3 (reclassifying and recalculating cluster centroids) until there is no further change in cluster centroids



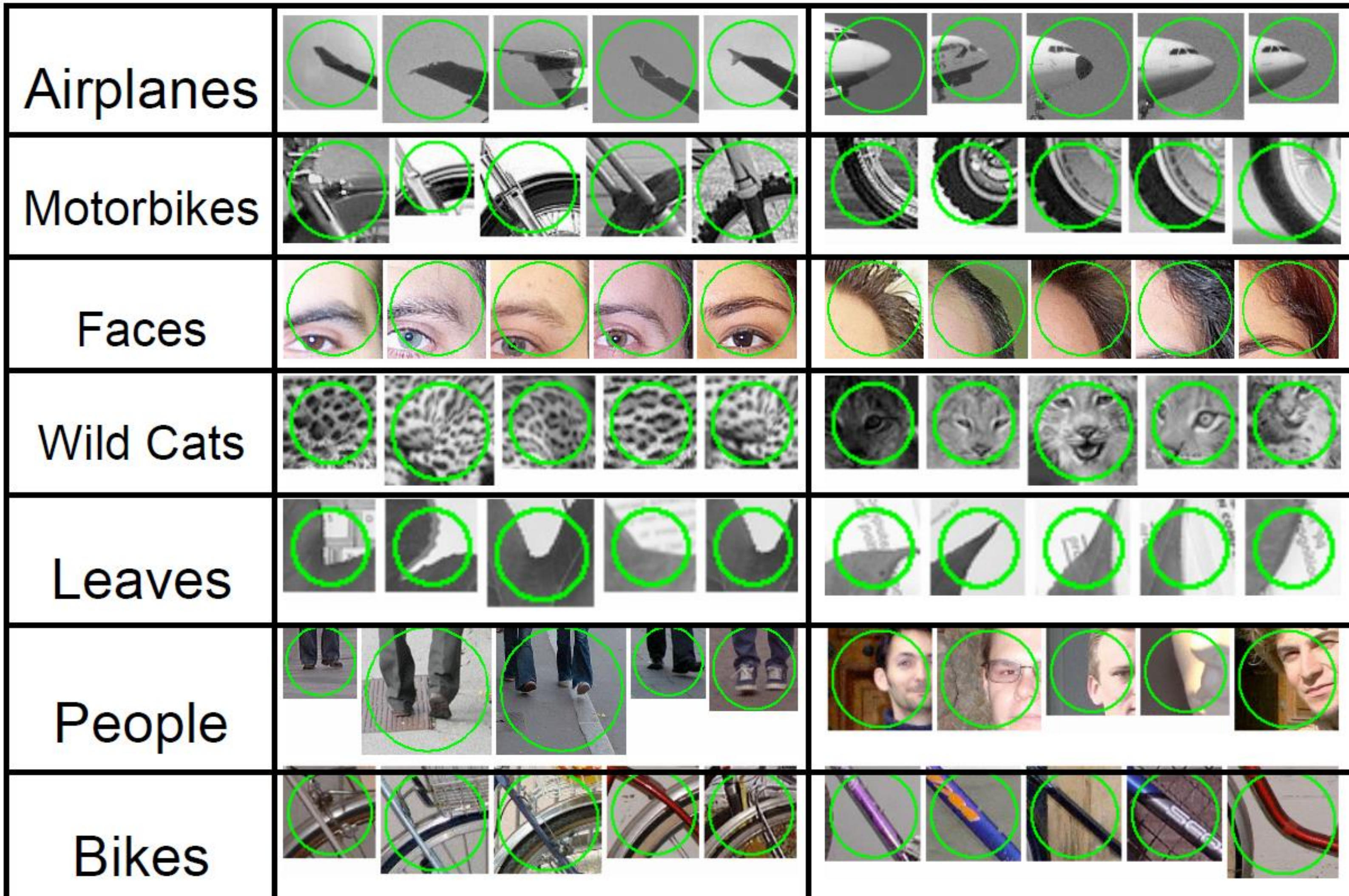
This is the visual vocabulary. Each final cluster centroid is a ‘word’ in feature space.

# The Visual Vocabulary

- Each ‘word’ defined by the cluster centre is also known as a codevector
- The entire visual vocabulary (i.e. the set of ‘words’) is also known as a codebook
- The codebook can be learned on a separate training set
- The codebook is used for quantising features
  - A vector quantiser takes a feature vector and maps it to the index of the nearest codevector in a codebook

- How does one choose vocabulary size?
  - Too small – Visual words are not representative of all patches
  - Too big – Results in overfitting and quantisation artifacts

# The Visual Vocabulary




http://www.ens-lyon.fr/LIP/Arenaire/ERVision/bof_classification_winter.pdf

# Classification

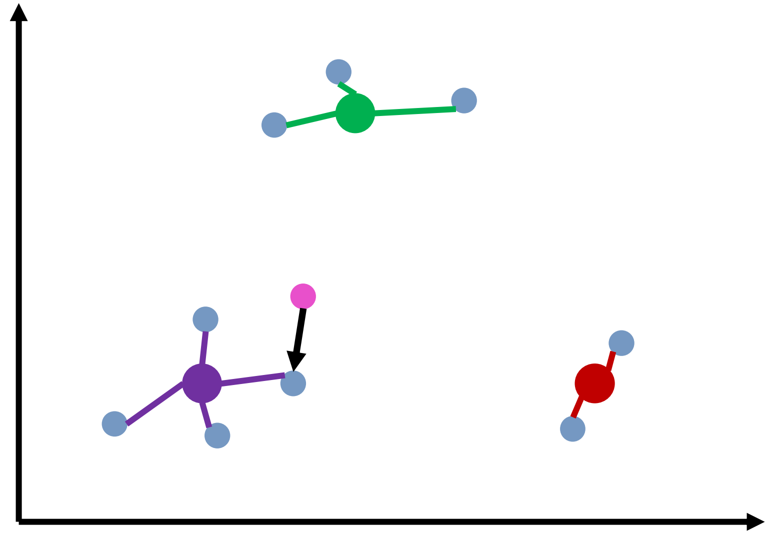
- Now that we have the bag-of-features representations of images from different classes, how do you learn a model for distinguishing between them?

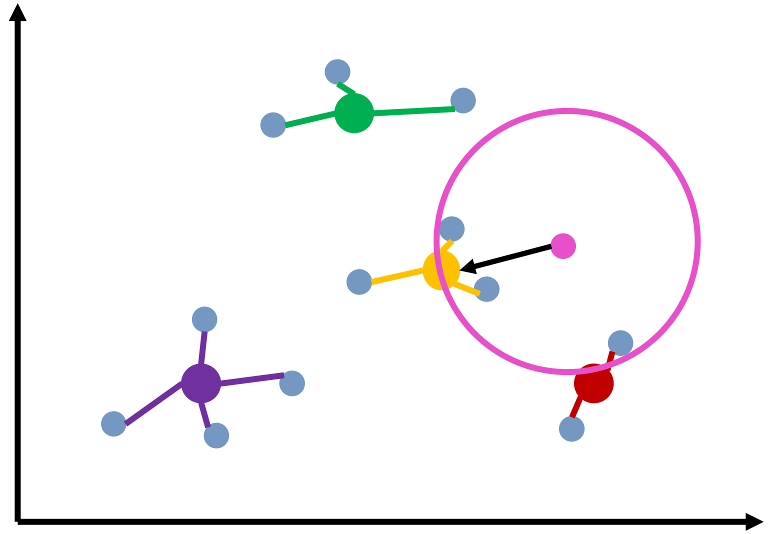
- There are two machine learning approaches:
  - Discriminative methods
    - Learns a decision rule (classifier) assigning bag-of-features representations of images to different classes
    - Examples include Nearest Neighbour, K-Nearest Neighbours, Support Vector Machines, AdaBoost
  - Generative learning methods
    - Models the probability of a bag of features given a class
    - Examples include the Naïve Bayes classifier or a hierarchical Bayesian models

# Nearest Neighbour and K-Nearest Neighbours

- With the Nearest Neighbour classifier, assign the label of the nearest training data point to each test data point

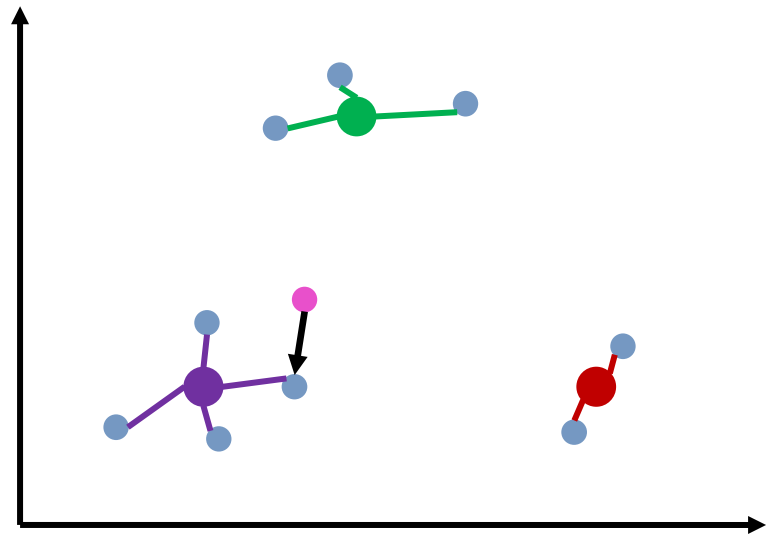
- With K-Nearest Neighbours, find the k closest points from the training data
- The labels of the k points vote to classify





# Nearest Neighbour and K-Nearest Neighbours

Two histograms can be compared using any of the following distances:



**Cosine distance**

$$D(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{x}\cdot\boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|} = \frac{x_1y_1 + \cdots + x_ny_n}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

**$x^2$ distance**

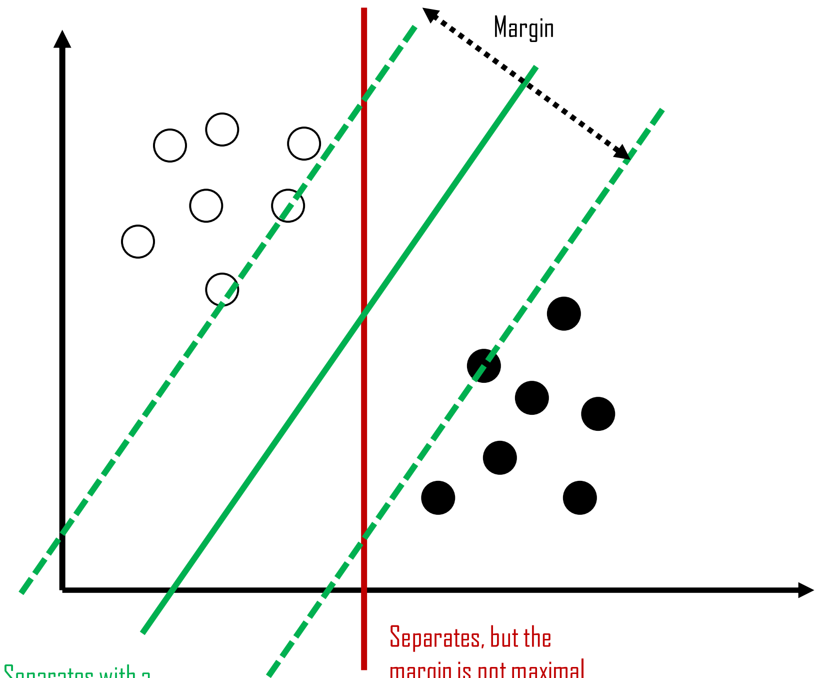
$$D(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{x_i + y_i}$$

**Quadratic distance**

$$D(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^2\right)$$

# Support Vector Machines

- Finds the hyperplane that maximises the margin between 2 classes (positive $y_i$=1 and negative $y_i$=-1)




The hyperplane is defined as the set of points **x** satisfying

$$\mathbf{w} \cdot \mathbf{x} - b = 0$$

And the two hyperplanes defining the margin are

$$\mathbf{w} \cdot \mathbf{x} - b = 1 \text{ and } \mathbf{w} \cdot \mathbf{x} - b = -1$$

Where w is the normal vector to the hyperplane, $b/\|\mathbf{w}\|$ is the offset of the hyperplane from the origin along **w**, and the distance between the two margin hyperplanes is $2/\|\mathbf{w}\|$. This margin should be maximised.

As well, all training data should be classified correctly

$$\mathbf{x}_i \; positive \; (y_i = 1){:} \, \mathbf{w} \cdot \mathbf{x} + b \geq 1$$
$$\mathbf{x}_i \; negative \; (y_i = -1){:} \, \mathbf{w} \cdot \mathbf{x} + b \leq -1$$

This is an optimisation problem

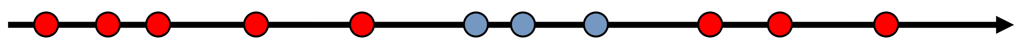
$$\min_{\mathbf{w},b} \frac{1}{2} \|\mathbf{w}\|^2$$

subject to:

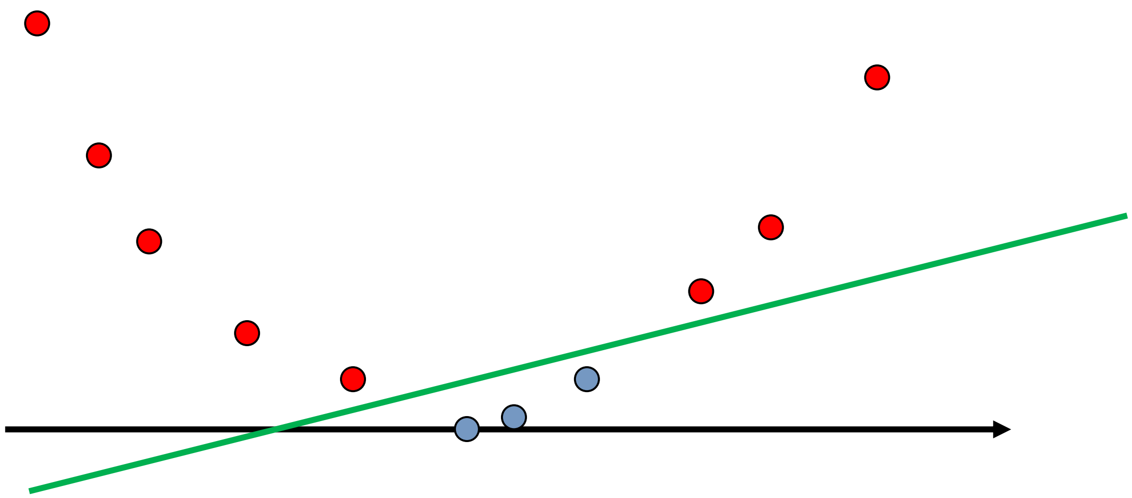
$$y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1$$

# Non Linear SVMs

- Sometimes, it's impossible to find a hyperplane that will separate the two groups



- Mapping it to a higher-dimensional space might help with separation



This is the kernel trick. The kernel is defined as $K(x, y) = \varphi(x) \cdot \varphi(y)$, where $\varphi(x)$ is a transform. In the original feature space, the decision boundary will be nonlinear. Common kernels include a histogram intersection kernel, generalised Gaussian kernel, etc.

# Multi-class SVMs

- As you'll already have spotted, SVMs only separate two possible classes
- What about >two classes?
- Most approaches reduce the single multiclass problem into multiple binary classification problems
  - One vs. Others
    - Training: Learn a SVM for each class vs. the others
    - Testing: Apply each learned SVM to test example and assign to the class of the SVM that returns the highest decision value
  - One vs. One
    - Training: Learn a SVM for each pair of classes
    - Testing: Use each learned SVM to 'vote' for a class to assign to the test example

# Image Representation and Classification

- So for new images:
  - Detect features
  - Classify each feature
  - Examine the frequency of each codeword (compare histograms)

# Other Object Recognition Techniques

- Recognition by Parts

- Appearance-based Methods
  - Edge matching
  - Greyscale matching
  - Gradient matching

- Feature-based Methods
  - Interpretation trees
    - Uses a tree search to find a mapping of model features to image features which is geometrically consistent
  - Invariants
    - Compute ‘global indices’ that do not change over viewing conditions

- Many more…

# Conclusion

- Object Recognition
- Bag of Features
  - Origins
  - Representing the Visual Vocabulary
  - Classification
- Other Object Recognition Techniques