Generative models:
- often generalise well and may therefore require less training data
- the models themselves may become more complex than is required for classification
- constructing such a model often requires specific domain expertise

Discriminative methods:
- methods usually perform better and are more efficient on specific (supervised) learning tasks
- the training data needs to be large enough to span the expected modes of variation in the data

Generative methods learn a generative likelihood model \( P(x|C_k) \) which can then be used for classification using Bayes' rule. Generative models have predictive power as they allow one to generate samples from the joint distribution \( P(x, C_k) \), and they are therefore popular for tasks such as the analysis and synthesis of facial expressions. Examples include probabilistic mixture models, most types of Bayesian networks, active appearance models, Hidden Markov models, and Markov random fields.

Discriminative methods learn a function \( q_k(x) \) which maps input features \( x \) to class labels \( C_k \) (see section 10.5), something that can also be done probabilistically according to the posterior probabilities \( q_k(x) = P(C_k|x) \). Examples include artificial neural networks, support vector machines, boosting methods, and linear discriminant analysis.
Support Vector Machines (SVMs)

- Discriminative classifier based on an optimal separating hyperplane (i.e., line for 2D case)
- Maximise the margin between the positive and negative training examples

Non-Linear SVMs: Feature Spaces

- General idea: using a kernel function, the original input space can be mapped to some higher-dimensional feature space where the training set is separable:

\[ \Phi: \mathbf{x} \rightarrow \phi(\mathbf{x}) \]

Dr Chris Town

Neural Networks

- Use a weighted sum of elements at the previous layer to compute results at next layer
- Apply a smooth threshold (activation) function from each layer to the next (introduces non-linearity)
- Initialise the network with small random weights

Optical character recognition (OCR)

Some applications:
- Postal and bank cheque routing
- Document and book digitisation
- Automated number plate recognition (ANPR)
- Text-to-speech synthesis for the blind
- Handwriting recognition for portable device interfaces

Modern approaches make heavy use of machine learning to allow recognition of multiple fonts and to cope with distortions, noise, and variations in size, slant, and line thickness.
The first stage of the network is a **convolutional layer** consisting of 6 feature maps. The neurons in each feature map have 25 adaptable weights corresponding to the elements of a 5x5 kernel which is convolved with the input image, plus an adaptable bias weight. Each feature map therefore has 28x28 (32-5+1=28) neurons, all of which share the same 26 weights.

Weights are shared by all the neurons in each convolutional feature map. The weights define a convolution kernel.

Feature extraction by convolution:

Outputs $o_{ij}$ of each first layer neuron $i$ are the result of applying an activation function $f_{act}$ (such as tanh) to the sum of its inputs (pixels in the input image $I$) multiplied by each of its weights $w_{mn}$ after adding an additional bias term $w_0$:

$$o_{ij} = f_{act}(w_0 + \sum \sum w_{mn}I_{i-m,j-n})$$

Note how this resembles a discrete convolution:

$$result(i,j) = \sum \sum kernel(m,n) \cdot image(i-m,j-n)$$

To allow combinations of visual feature responses to be processed in a more position independent way, the output of the first layer is first **sub-sampled** by a factor of 2 in each spatial dimension and the result is fed to another convolutional layer. This layer uses a set of 12 feature maps with 5x5 kernels.

After another stage of subsampling, we are left with 5x5 feature maps. A final stage of convolution (layer 5) with 5x5 kernels produces single outputs (5-5+1=1). Layer 5 contains 100 such neurons. Finally there are 10 outputs corresponding to the digits 0-9.
Shifting the input image results in a corresponding shift in the output of the feature maps. 

- can be used as an efficient scanning window detector

LeNet is used to classify handwritten digits. Notice that the test error rate is not the same as the training error rate, because the learning “overfits” to the training data.

The 82 errors made by LeNet5

Notice that most of the errors are cases that people find quite easy. The human error rate is probably 20 to 30 errors.

Limitations of textual image retrieval

The Web: Google, Yahoo, Microsoft Bing etc. only index text

But: Only ~27% of internet is text, can’t search media content

The Home:

- 300 million digital cameras and over 500 million camera phones are sold each year
- over 500 billion digital consumer pictures
  → often called “DSC00xxx”...
  → no way of searching, organising, or browsing by content

Source: Berkeley, 2003

Image search - Challenges

- What is in the picture?
  - Metadata
  - Visual content (CBIR, content based image retrieval)

- What is a good query?
  - Metadata: keyword search sometimes “hit and miss”
  - Visual content: different query mechanisms

Problems with CBIR: the “semantic gap”
Evolution of Content Based Image Retrieval (CBIR)

What is in the picture?
- Colour histograms
- Texture analysis
- Histograms of filter outputs

1st Generation

Evolution of Content Based Image Retrieval (CBIR)

What is in the picture?
- Texture
- Colour histogram
- Shading histograms

Evolution of Content Based Image Retrieval (CBIR)

What is in the picture?
- Region segmentation
- Features detection
- Object detection
- Object models

2nd Generation

Evolution of Content Based Image Retrieval (CBIR)

What is in the picture?
- Ontologies
- Machine learning
- Statistical methods
- Object and scene classifiers

3rd Generation

Limitations of textual image retrieval

- Query by annotation or document context: keyword search on text annotations, image metadata or image document context (e.g. Google image search)
  But: images rarely come with usable/consistent annotations or captions, automatic descriptions are unreliable

What is a good query?

- Query by feature range or predicate: users set thresholds on global (e.g. colour histogram) or local (e.g. localised texture pattern) appearance features
  But: quite low-level, requires user sophistication (and patience...)

What is a good query?

- Query by template, region selection, or sketch: users sketch or select parts of the images they are looking for
  But: time consuming, hard to represent abstractions and invariants
What is a good query?

- Query-by-example: users provide one or more (weighted) sample images.
  But: “chicken and egg” problem, saliency is ill-defined.
Problems with CBIR: the "semantic gap"
The case for ontology based CBIR

Problems with current image search technology:

• search-by-context (e.g. web search): ignores the image
• search-by-content: cumbersome interfaces, not enough semantics

Ontology-based approach:

• search "inside the picture", i.e. the actual content of an image
  → fast fully automatic image analysis
  → no need for image annotations or metadata
• flexible query language based on an ontology
  → no need for example images or sketches
  → easy to integrate (con)text or make multilingual

Ontologies

• Ontology is the theory of objects in terms of the criteria which allow one to distinguish between different types of objects and the relations, dependencies, and properties through which they may be described.
  → What you’re looking for and how to find it

• Explicit representation of ontological commitments (concepts):
  Categories - Objects - Attributes - Relations

• Bridges between high-level concepts and low-level primitives

• Allows concise representation of context and world knowledge at a meta level

Ontologies - examples

Above: Semantic Web architecture
Left: John Wilkins (1668): "An essay towards a real character and philosophical language"

OQUEL – Image retrieval syntax

OQUEL (ICON) Grammar:

```
G : { 
  Sentence S -> R 
  Requirement R -> modifier? (category | SB | BS) 
  not? R CB BS? 
  Relation BR -> SB interpolation SB 
  Specification block SB -> (CS | PS) + LS* 
  Content specification CS -> visualcategory | semanticcategory | not? CS (CB CS)* 
  Location specification LS -> location | not? LS (CB LS)* 
  Property specification PS -> shapdescriptor | colordescriptor | not? PS (CB PS)* 
  Connective CB -> and | or | xor 
} 
```

Tokens and Vocabulary

• Vocabulary of about 400 words augmented with WordNet synsets

• Categories of terminal symbols:
  • Modifier: Quantifiers such as "a lot of", "none", "as much as possible"
  • Scene descriptor: e.g. "countryside", "city", "indoors"
  • Binary relation: e.g. "larger than", "close to", "similar size as", "above", "similar content"
  • Visual category: e.g. "water", "skin", "cloud"
  • Semantically category: Derived categories, e.g. "people", "vehicles"
  • Location: e.g. "background", "lower half", "top right corner"
  • Shapedescriptor: e.g. "straight line", "blob shaped"
  • Colouredescriptor: e.g. "bright red", "vivid colours", "RGB(0,0,128)"
  • Sizedescriptor: e.g. "at least 10%" [of image area], "largest region"

Content Extraction and Representation

• Image segmentation and region properties
colour, shape, texture, size, absolute position
• Region classification by trained neural networks
visual categories of "stuff" (grass, sky, skin, ...)
• Face Detection using skin and geometric features
• Region mask: pixel region membership
• Region graph of relative spatial relationships
adjacency, boundaries, containment
• Grid pyramid of stuff classifications
  • Overall classification
  • Image fifths
  • Chess board
**Image segmentation**

Images are segmented into non-overlapping regions and classified using neural networks.

Image segmentation according to Sinclair:

1. **Full three colour edge detection**
   \[ dI^2 = dR^2 + dG^2 + 3dB^2 \]
   \[ dI = dR + dG + dB, \]
   \[ dC^2 = (dR - dG)^2 + (dR - dB)^2 + (dG - dB)^2 \]
   \[ + (dR - dB)^2 + (dG - dB)^2 + (dG - dI)^2 \]

**Face Detection and Colour Labelling**

- **Face detection**: ellipse fitting of skin regions followed by eye detection. Candidate features are extracted from a binarised version of the image. Eyes are detected by a nearest-neighbour shape classifier derived by pairwise geometric histogram binning of feature orientations and distances.
- **Colour descriptors**: Nearest-neighbour classifiers using Euclidean distances in HSV or RGB space ("black", "blue", "cyan", "grey", "green", "magenta", "orange", "pink", "red", "white", "yellow", "brown").

**OQUEL – Sample Query A**

- some sky which is close to trees in upper corner, size at least 20%
- indoors & people in foreground

**Region classification**

- Region shape, colour, shading, and texture properties serve as feature vectors for trained neural network (MLP and RBF) classifiers for visual categories: Brick, Clouds, Cloth, Grass, Internal Walls, Skin, Sky, Snow, Tarmac, Trees, Water, Wood

**OQUEL – Examples**

- indoors & people in foreground
Google: “bright red and stripy” (stripey)

Dr Chris Town

OQUEL - Results

Dr Chris Town

“Imense” - Image Analysis

Dr Chris Town

“Imense” - Image Retrieval

Dr Chris Town