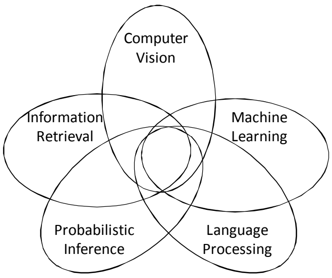
Computer Vision

Computer Science Tripos Part II

Dr Christopher Town

11. Learning and statistical methods in vision. Optical character recognition and Content based image retrieval.

UNIVERSITY OF CAMBRIDGE



- 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 $y_k(x)$ 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 $y_k(x) = P(C_k|x)$. Examples include artificial neural networks, support vector machines, boosting methods, and linear discriminant analysis.

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

Discriminative vs. generative

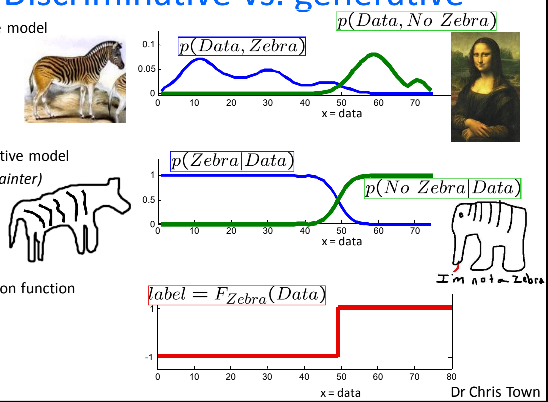
- Generative model

(*The artist*)

- Discriminative model

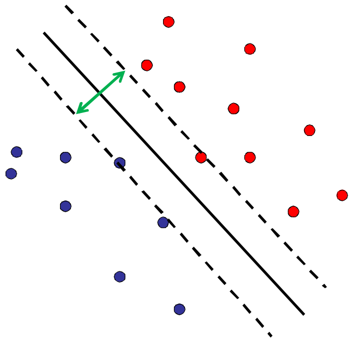
(*The lousy painter*)

- Classification function



A.Torralba

Support Vector Machines (SVMs)

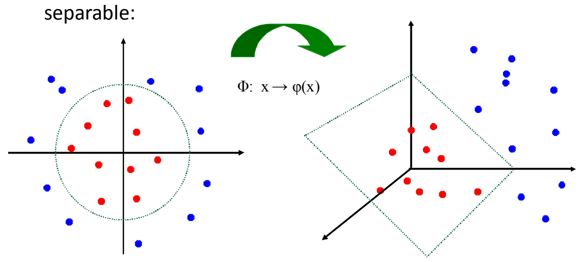


- Discriminative classifier based on an ***optimal separating hyperplane*** *(i.e. line for 2D case)*
- Maximise the **margin** between the positive and negative training examples

Slide credit: Kristen Grauman

Non-Linear SVMs: Feature Spaces

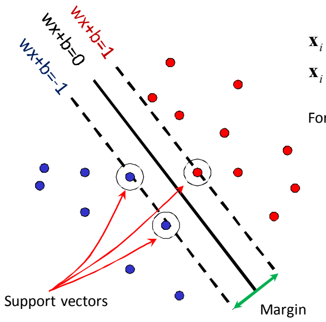
- General idea: using a **kernel function**, the original input space can be mapped to some higher-dimensional feature space where the training set is separable:



Slide from Andrew Moore's tutorial: http://www.autonlab.org/tutorials/svm.html

Support Vector Machines (SVMs)

- Support vectors represent the line that maximises the margin between feature vectors belonging to the two classes



$\mathbf{x}_i$ positive $(y_i = 1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

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

For support, vectors, $\quad \mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

Quadratic optimisation problem

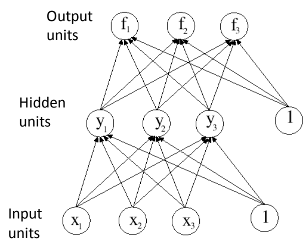
Minimize $\frac{1}{2}\mathbf{w}^T\mathbf{w}$

Subject to $y_i(\mathbf{w} \cdot x_i + b) \geq 1$

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Neural Networks

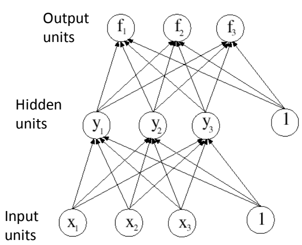
- Use a weighted sum of elements at the previous layer to compute results at next layer
- Apply a smooth threshold (activation) function from each layer to the next (introduces non-linearity)
- Initialise the network with small random weights



Slide credit: David Lowe

Neural Networks

- Perform gradient descent, making small changes in the direction of the derivative of error with respect to each parameter
- Network structure (and feature input) is often designed by hand to suit the problem, so only the weights are learned



Slide credit: David Lowe

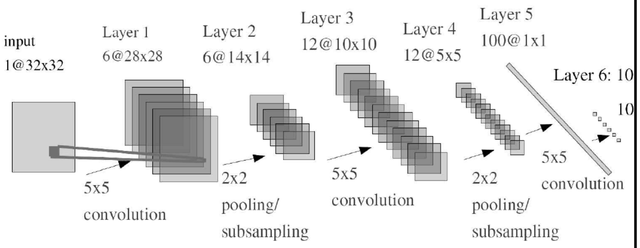
Optical character recognition (OCR)

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.

Convolutional Neural Network



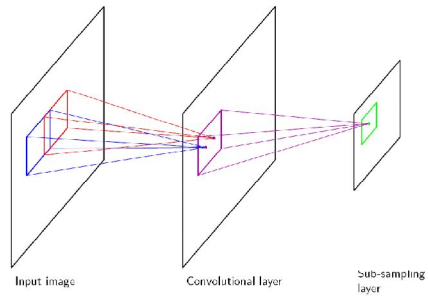
The first stage of the network is a **convolutional layer** consisting of 6 **feature maps**. The neurons in each feature map have 25 adaptable weights corresponding to the elements of a **5x5 kernel** which is **convolved** with the input image, plus an **adaptable bias** weight. Each feature map therefore has 28x28 (32-5+1=28) neurons, all of which **share the same 26 weights**.

Dr Chris Town

Convolutional Neural Network

Weights are **shared** by all the neurons in each convolutional feature map

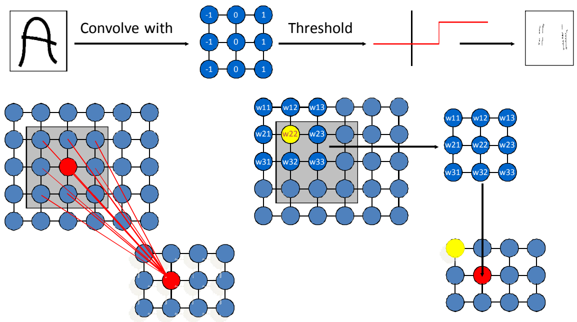
The weights define a **convolution kernel**



B. Frey

Dr Chris Town

Feature extraction by convolution



Dr Chris Town

Convolutional Neural Network

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:

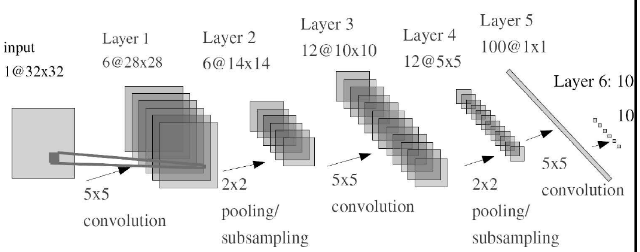
$$o_{ij} = f_{act}(w_0 + \sum_m \sum_n w_{mn} I_{i-m,j-n})$$

Note how this resembles a discrete convolution:

$$\text{result}(i,j) = \sum_m \sum_n \text{kernel}(m,n) \cdot \text{image}(i-m, j-n)$$

Dr Chris Town

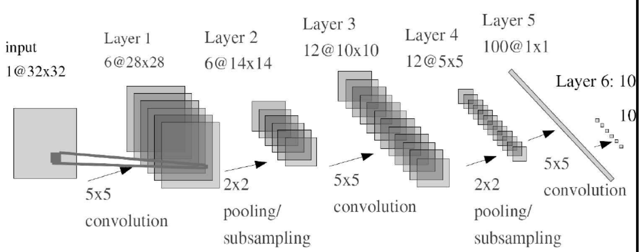
Convolutional Neural Network



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.

Dr Chris Town

Convolutional Neural Network



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.

Dr Chris Town

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**

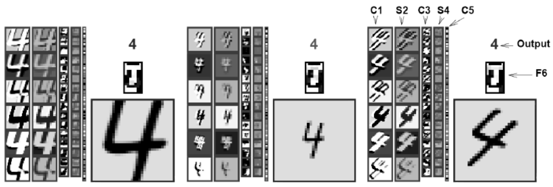


Fig. 4. Size-normalized examples from the MNIST database.

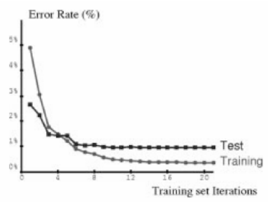
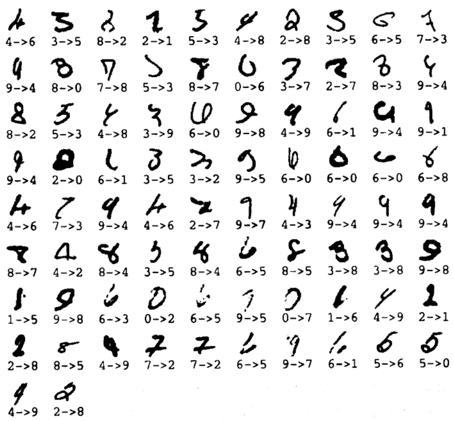


Fig. 5. Training and test error of LeNet-5 as a function of the number of passes through the 60 000 pattern training set (without distortions). The average training error is measured on-the-fly as training proceeds. This explains why the training error appears to be larger than the test error initially. Convergence is attained after 10–12 passes through the training set.

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.

Figure from "Gradient-Based Learning Applied to Document Recognition", Y. Lecun et al Proc. IEEE, 1998 copyright 1998, IEEE

The 82 errors made by LeNet5

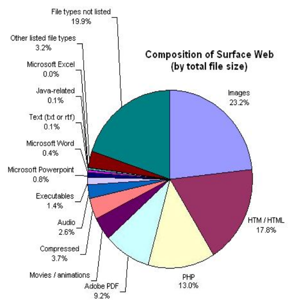


Notice that most of the errors are cases that people find quite easy.

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***



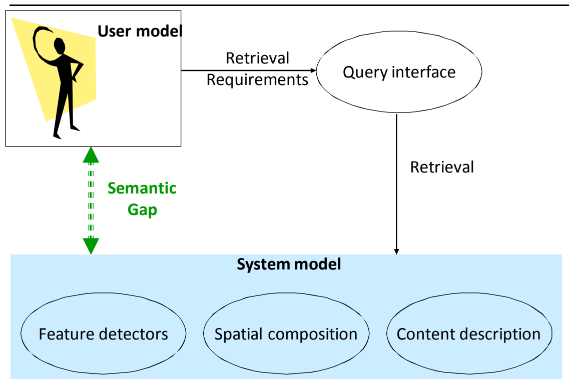
Source: Berkeley, 2003

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***

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

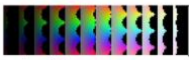
Problems with CBIR: the "semantic gap"



Evolution of Content Based Image Retrieval (CBIR)

What is in the picture?

texture



colour histogram

shading histogram

Colour histograms

Texture analysis

Histograms of filter outputs

1st Generation

Evolution of Content Based Image Retrieval (CBIR)

What is in the picture?

Region segmentation

Features detection

Object detection

Object models

2nd Generation

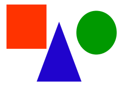
Colo

Textu

Histo

1st Generation

edge detection and segmentation



shape

Evolution of Content Based Image Retrieval (CBIR)

What is in the picture?

Ontologies

Machine learning

Statistical methods

Object and scene classifiers

3rd Generation

Regi

Featu

Obje

Objec

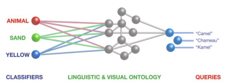
2nd Generation

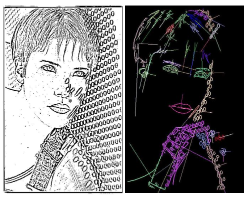
Colo

Textu

Histo

1st Generation

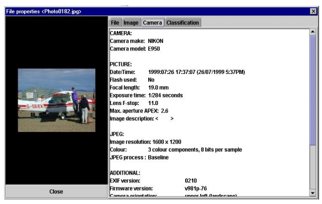


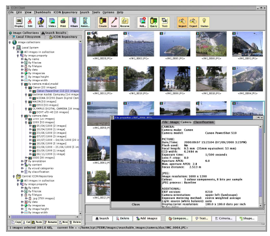


Limitations of textual image retrieval

- **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

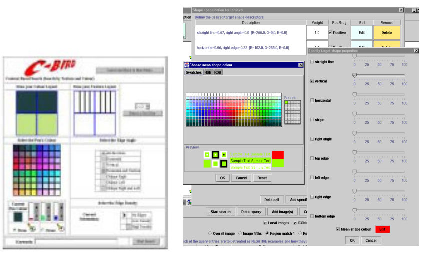




What is a good query?

- **Query by feature range or predicate**: users set thresholds on global (e.g. colour histogram) or local (e.g. localised texture pattern) appearance features

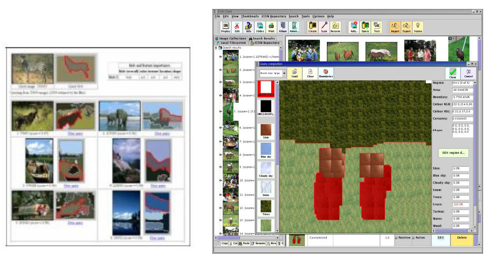
But: quite low-level, requires user sophistication (and patience…)



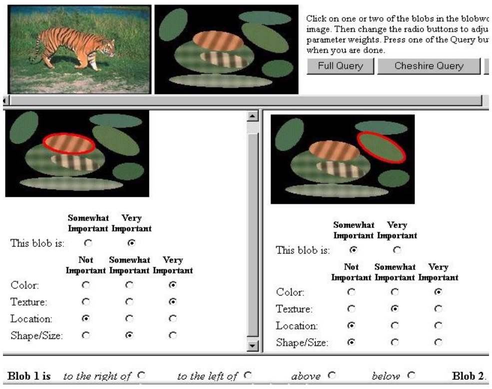
What is a good query?

- **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

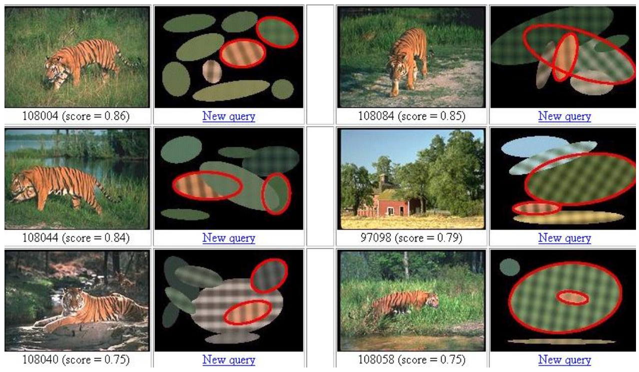


Berkeley “Blobworld”



Dr Chris Town

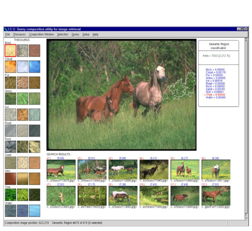
Berkeley “Blobworld”

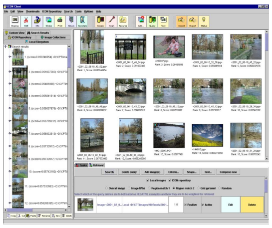


Dr Chris Town

What is a good query?

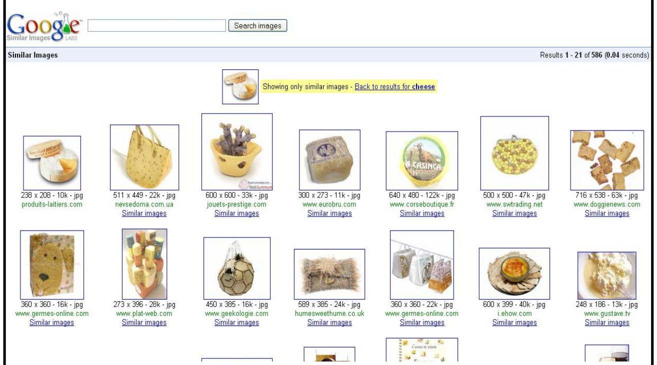
– **Query-by-example**: users provide one or more (weighted) sample images
But: “chicken and egg” problem, saliency is ill-defined





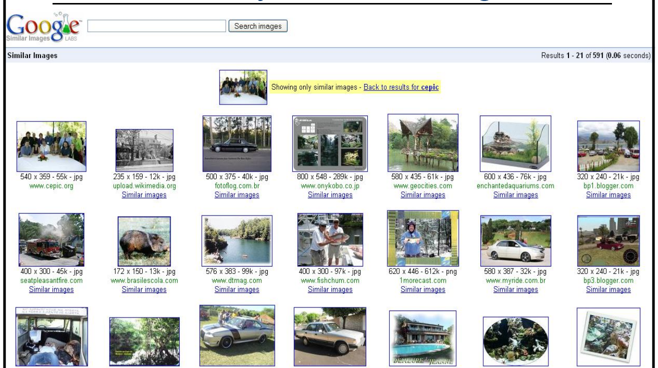
Dr Chris Town

Similarity Search - Google



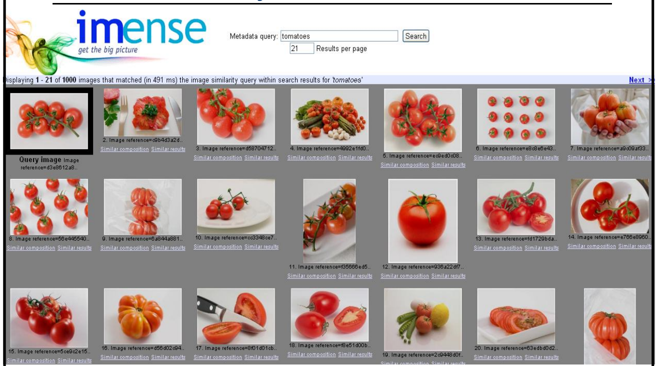
Dr Chris Town

Similarity Search - Google



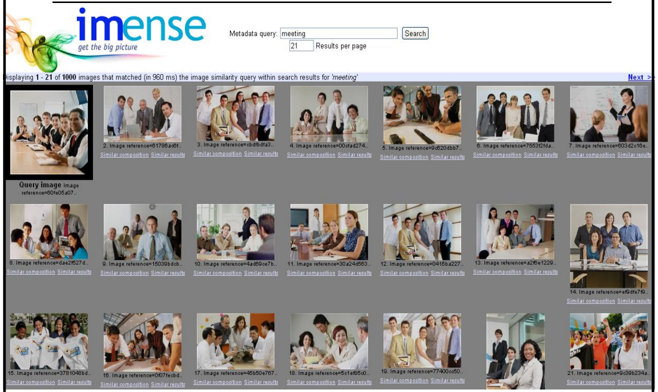
Dr Chris Town

Similarity Search - imense

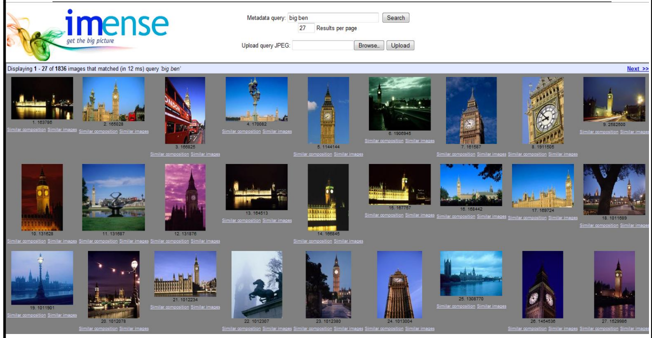


Dr Chris Town

Similarity Search - imense

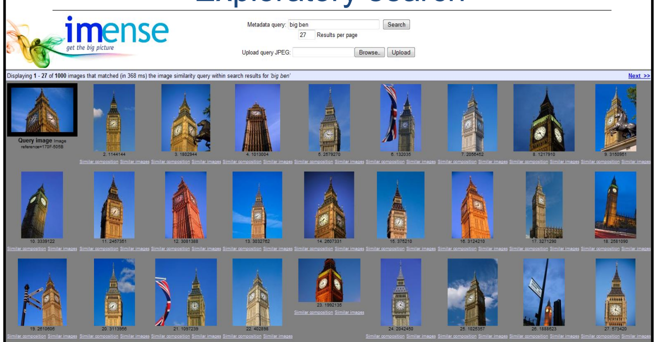


Exploratory search

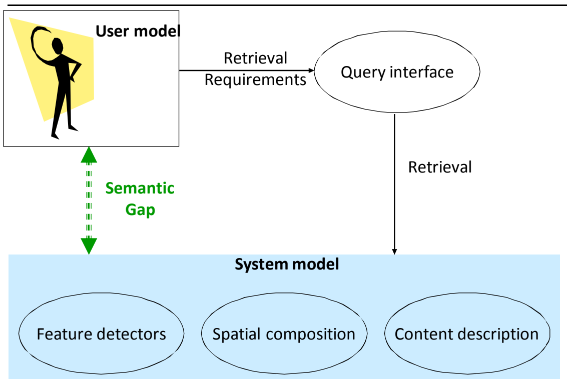


Exploratory search

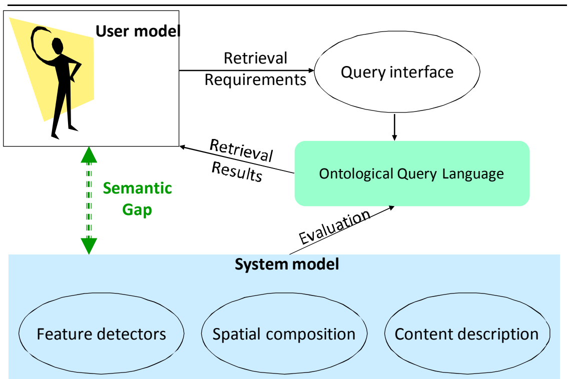
Exploratory search



Problems with CBIR: the “semantic gap”



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.

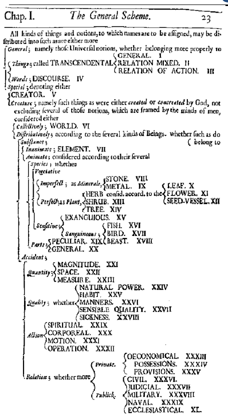
  → ***What you're looking for and how to find it***

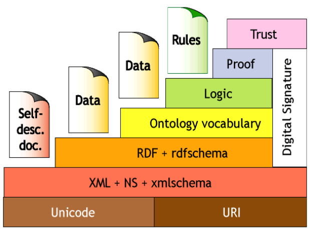
- Explicit representation of ontological commitments (concepts):

  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	$\rightarrow$	R
Requirement	R	$\rightarrow$	$modifier?\ (metacategory \mid SB \mid BR)$ $\mid not?\ R\ (CB\ R)?$
Relation	BR	$\rightarrow$	$SB\ binaryrelation\ SB$
Specification block	SB	$\rightarrow$	$(CS \mid PS) +\ LS *$
Content specification	CS	$\rightarrow$	$visualcategory \mid semanticcategory \mid$ $not?\ CS\ (CB\ CS)?$
Location specification	LS	$\rightarrow$	$location \mid not?\ LS\ (CB\ LS)?$
Property specification	PS	$\rightarrow$	$shapedescriptor \mid colourdescriptor \mid$ $sizedescriptor \mid not?\ PS\ (CB\ PS)?$
Connective	CB	$\rightarrow$	$and \mid or \mid xor;$
	$\}$		

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"
  - **Binaryrelation**: e.g. "larger than", "close to", "similar size as", "above", "similar content"
  - **Visualcategory**: e.g. "water", "skin", "cloud"
  - **Semanticcategory**: Derived categories, e.g. "people", "vehicles"
  - **Location**: e.g. "background", "lower half", "top right corner"
  - **Shapedescriptor**: e.g. "straight line", "blob shaped"
  - **Colourdescriptor**: 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*:

(Sinclair, D.: "Smooth region structure: folds, domes, bowls, ridges, valleys and slopes", CVPR 2000)

1.) Full three colour edge detection

$$dT = dI_i^2 + dI_j^2 + 3.0dC$$

$$dI_i = dR_i + dG_i + dB_i$$

$$\begin{aligned} dC \quad = \quad & \sqrt{((dB_i - dG_i)^2 + (dR_i - dB_i)^2 + (dG_i - dR_i)^2} \\ + \quad & (dB_j - dG_j)^2 + (dR_j - dB_j)^2 + (dG_j - dR_j)^2) \end{aligned}$$

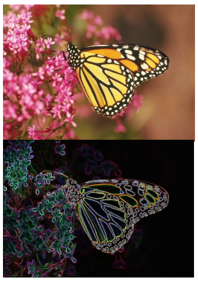
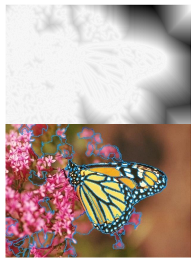


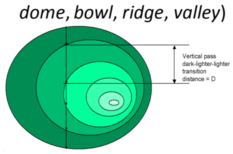
Image segmentation

2.) Voronoi transform of edge image, regions are grown agglomeratively from distance peaks

3.) Merge similar regions, find and cluster texture features, use clusters to unify textured regions

4.) Compute smooth region internal brightness structure from isobrightness contours and intensity gradients (classify into *dome*, *bowl*, *ridge*, *valley*)

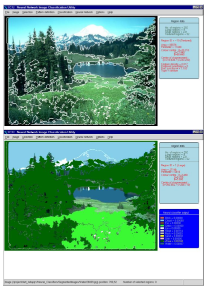


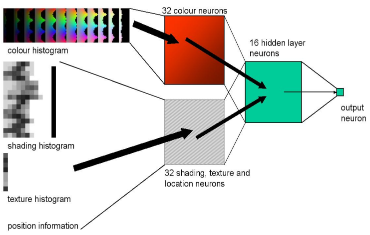


5.) Quantify other region properties: colour histogram, colour covariance, texture feature, size/colour/orientation/connectivity, shape and boundary descriptors

Region classification

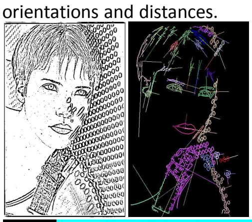
• Region shape, colour, shading, and texture properties serve as feature vectors for trained neural network (MLP and RBF) classifiers for ***visual categories:*** **Brick, Clouds, Cloth, Grass, Internal Walls, Skin, Sky, Snow, Tarmac, Trees, Water, Wood**



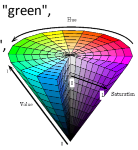


Face Detection and Colour Labelling

• **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.



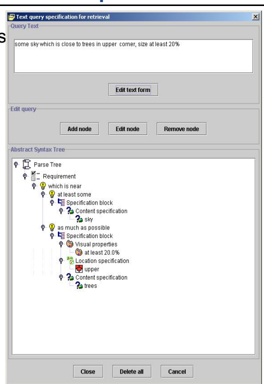
• **Colour descriptors**: Nearest-neighbour classifiers using Euclidean distances in HSV or RGB space ("black", "blue", "cyan", "grey", "green", "magenta", "orange", "pink", "red", "white", "yellow", "brown").



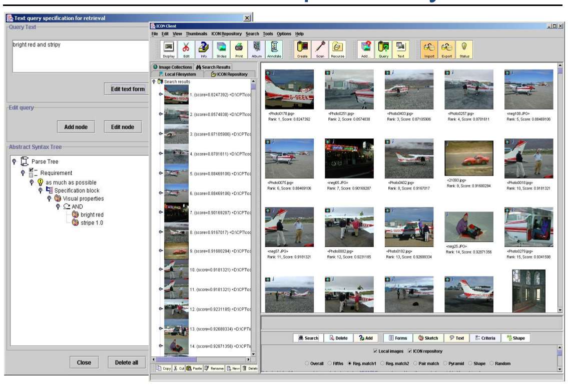
OQUEL – Examples

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

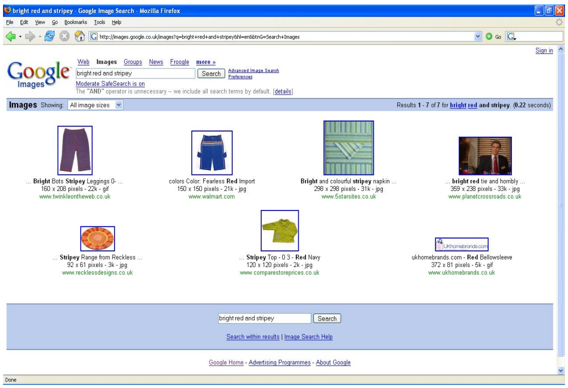
Parse Tree
- AND
  - Requirement
    - as much as possible
      - Specification block
        - Location specification
          - foreground
        - Content specification
          - people
  - Requirement
    - as much as possible
      - indoors



OQUEL – Sample Query A

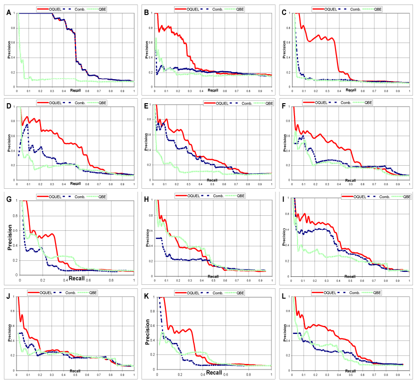


Google: “bright red and stripy”(stripey)



Dr Chris Town

OQUEL - Results

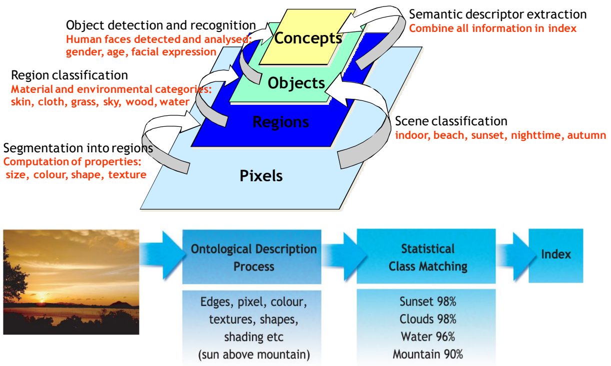


$$Rank^{\sim} = \frac{1}{NN_{rel}}\{\sum_{i=1}^{N_{rel}} R_i - \frac{N_{rel}\{N_{rel}+1\}}{2}\}$$

$Rank^{\sim}$			
Query	OQUEL	Comb	QBE
A	0.2176	0.2175	0.3983
B	0.2915	0.3072	0.3684
C	0.2628	0.3149	0.3521
D	0.1935	0.2573	0.2577
E	0.2152	0.2418	0.3324
F	0.1969	0.1816	0.2475
G	0.3147	0.3766	0.2831
H	0.3312	0.2947	0.2952
I	0.1863	0.2123	0.2105
J	0.2170	0.2113	0.2020
K	0.3151	0.4078	0.3377
L	0.2367	0.2558	0.3351

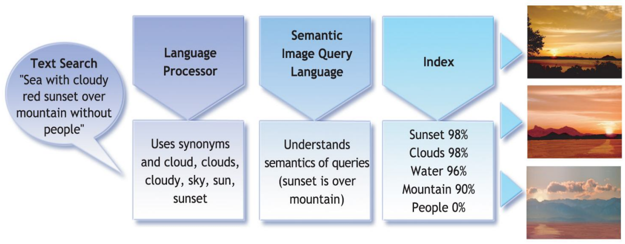
Dr Chris Town

“Imense” - Image Analysis



Dr Chris Town

“Imense” - Image Retrieval



Dr Chris Town