Computer Vision (7CCSMCVI / 6CCS3COV)

Recap

- **Image formation**
- **Low-level vision**
- **Mid-level vision**
- **High-level vision**
  - **Artificial**
    - template matching
    - sliding window
    - edge matching
    - model-based
    - intensity histograms
    - implicit shape model
    - SIFT feature matching
    - bag-of-words
    - geometric invariants
  - **Biological** ← Today

Today

- Theories of object recognition / categorisation:
  - object-based (3D) vs image-based (2D)
  - configural (global) vs featural (local)
  - rules vs exemplars vs prototypes
- Theories of cortical processing:
  - hierarchical neural network models
    - » Feedforward (HMAX, CNN)
    - » Recurrent
- Top-down vs Bottom-up
  - Bayesian inference

Object based vs Image based theories

Object based:

- each object represented by storing a 3D model
- object-centred reference frame

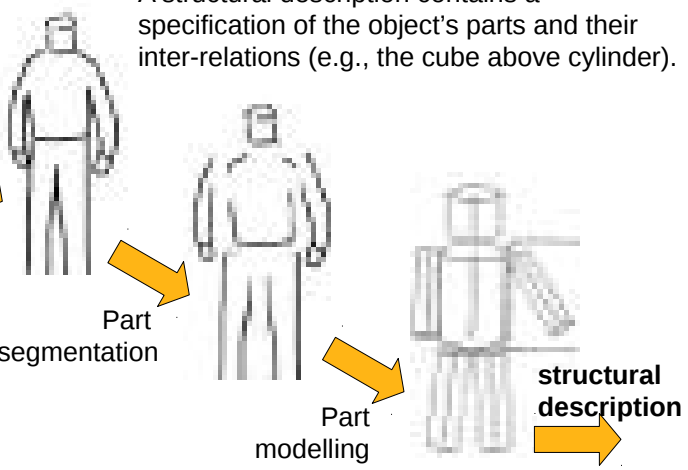
Image based:

- each object represented by storing multiple 2D views (e.g. images)
- viewer-centred reference frame

Object based: Recognition By Components

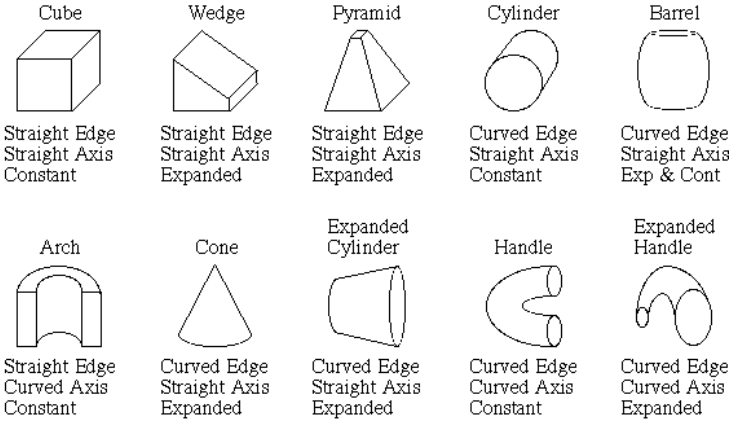
Representations of objects are stored in the brain as **structural descriptions**.

A structural description contains a specification of the object's parts and their inter-relations (e.g., the cube above cylinder).



Object based: Recognition By Components

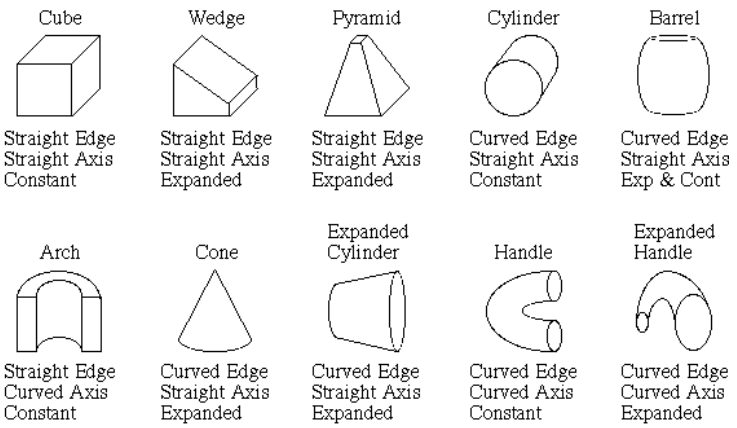
Hypothesis: there is a small number of geometric components that constitute the primitive elements of the object recognition system (like letters forming words).



Object based: Recognition By Components

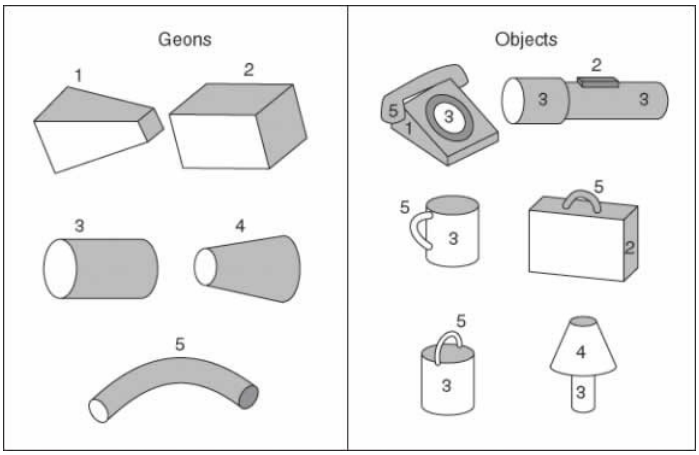
Hence, an object is an arrangement of a few simple three-dimensional shapes called *geometrical icons*, or **geons**.

Geons are simple volumes such as cubes, spheres, cylinders, and wedges.



Object based: Recognition By Components

Different combinations of geons can be used to represent a large variety of objects

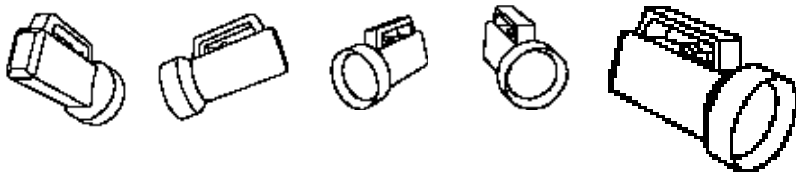


Object based: Recognition By Components

Geons are chosen to be:

- sufficiently different from each other to be easily discriminated
- robust to noise (can be identified even with parts of image missing)
- view-invariant (look similar from most viewpoints)

Different views of the same object are represented by the same set of geons, in the same arrangement. Therefore, the model achieves viewpoint invariance.

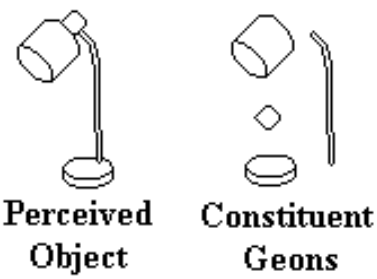


Object based: Recognition By Components

Matching

Recognition involves recognizing object elements (geons) and their configuration

The visual system parses an image of an object into its constituent geons.



Interrelations are determined, such as relative location and size (e.g., the lamp shade is left-of, below, and larger-than the fixture).

The geons and interrelations of the perceived object are matched against stored structural descriptions.

If a reasonably good match is found, then successful object recognition will occur.

Object based: Recognition By Components

Problems:

- difficult to decompose an image into components (i.e. to map an image onto a representation in geons)
- difficult to represent many natural objects using geons (may not have a simple parts-based description, e.g. a tree)
- cannot detect finer details which are necessary for identification of individuals or discrimination of similar objects. e.g.:

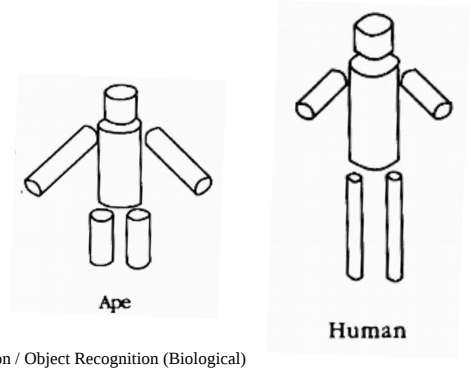


Image based

3D object represented by multiple, stored, 2D views of the object.

Object recognition occurs when a current pattern matches a stored pattern.

- Template matching
  - » An early version of the image-base approach.
  - » Too rigid to account for flexibility of human object recognition.
- Multiple Views approach
  - » More recent version of the image-based approach.
  - » Through experience, we encode multiple views of objects.
  - » These serve as the templates for recognition, but interpolation between stored views enables recognition of objects from novel viewpoints.

Configural vs Featural theories

Who is this?

Is he looking well?

Configural vs Featural theories

Configural vs Featural theories

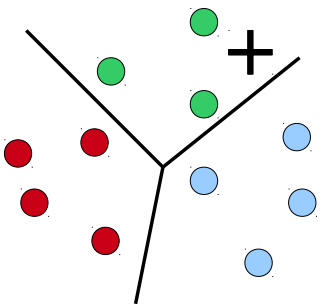
Inverted faces: featural processing

- features processed independently, relationships between features ignored.

Upright faces: configural processing

- holistic, global

Rules vs Prototypes vs Exemplars



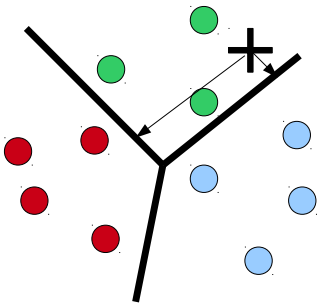
How are the boundaries between different categories defined?

How are new stimuli assigned to the closest category?

 = previous examples of stimuli from 3 different categories

 = a new stimulus from an unknown category

Rules



Category membership defined by abstract rules, e.g.

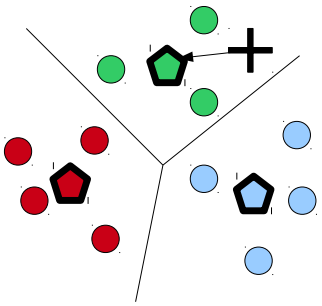
- has three sides = triangle
- has four legs and barks = dog
- has a beak and feathers = bird

Anything that satisfies the rule(s) for the category goes into that category

For: over-extension of rules of grammar, e.g. “goed” instead of went, “bitted” instead of bitten, “mouses” instead of mice.

Against: Some members are better examples of a category (graded membership), e.g. bear is a better mammal than a whale, 4 is a better even number than 106, pigeon is a better bird than penguin.

Prototypes



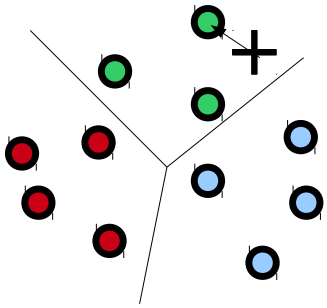
Calculate the average (or prototype) of all the individual instances from each category.

A new stimulus is compared to the stored prototypes and assigned to the category of the nearest one

For: prototypical category members are accessed more quickly and learnt more easily (e.g. pigeons vs penguins.)

Against: variations within a class can not be represented.

Exemplars



Specific individual instances of each category (“exemplars”) stored in memory.

A new stimulus is compared to the stored exemplars and assigned to the category of the nearest one

For: successfully predicts some kinds of mis-categorizations (e.g., a whale as a fish).

Against: Some members are better examples of a category (graded membership), e.g. bear is a better mammal than a whale, 4 is a better even number than 106, pigeon is a better bird than penguin.

Classifiers

Prototype and Exemplar theories in psychology correspond to standard classification methods used in pattern recognition / machine learning.

These methods use “supervised” learning:

- assumes that class for each data point in the training set is known
- new (unknown) data points assigned to appropriate class based on similarity to training examples

The alternative is unsupervised learning:

- assumes that class for each data point is unknown
- all data points assigned to appropriate class based on similarity

We previously came across unsupervised pattern recognition / machine learning methods, called clustering, when discussing image segmentation techniques (i.e. k-means clustering, agglomerative hierarchical clustering, graph cutting).

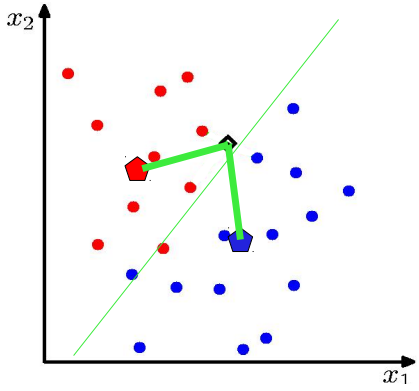
Nearest Mean Classifier (Prototype)

For each class

- calculate the mean of the feature vectors for all the training examples in that class

For each new stimulus

- find the closest class prototype and assign new stimulus to that class label



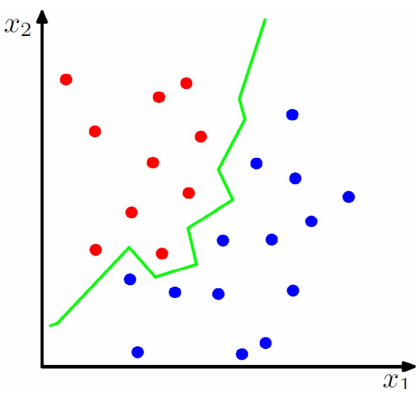
Decision boundary is linear. Hence, suitable only if data is linearly separable

Nearest Neighbour Classifier (Exemplar)

- Save the vectors for all the training examples (instead of just the mean for each class)

For each new stimulus

- find the closest training exemplar and assign new stimulus to that class label



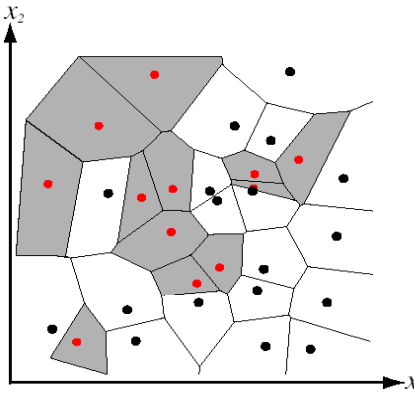
Decision boundary is non-linear (piecewise linear). Hence, suitable if data is non-linearly separable.

Nearest Neighbour Classifier (Exemplar)

- Save the vectors for all the training examples (instead of just the mean for each class)

For each new stimulus

- find the closest training exemplar and assign new stimulus to that class label



Decision boundaries form Voronoi partitioning of feature space.

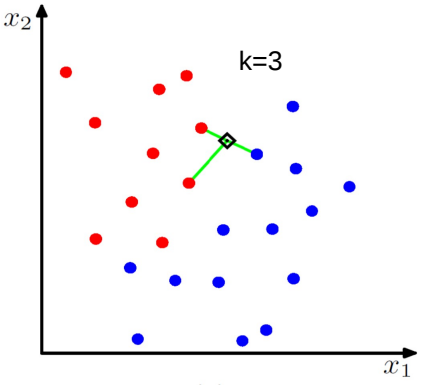
Doesn't deal with outliers.

K-Nearest Neighbours Classifier

- Save the vectors for all the training examples (instead of just the mean for each class)

For each new stimulus

- find the k closest training exemplars and assign new stimulus to the class label of the majority of these points (e.g. closest points vote on correct label)



Decision boundary is non-linear. Hence, suitable if data is non-linearly separable.

k typically small and odd (to break ties).

Increasing k reduces the effects of outliers

Similarity Measures

Determining the nearest neighbour(s) or nearest mean requires some measure of the distance between two sets of features.

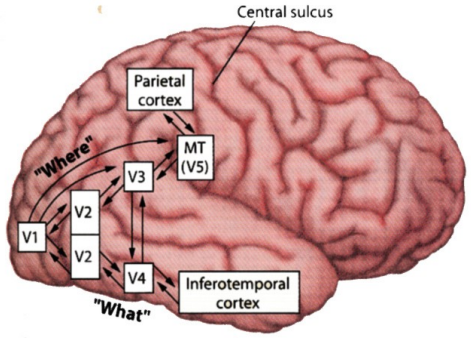
As previously, we can either find the minimum distance, e.g.:

- Sum of Squared Differences (SSD)
- Euclidean distance
- Sum of Absolute Differences (SAD) = Manhattan distance

Or, find the maximum similarity, e.g.:

- Cross-correlation
- Normalised cross-correlation
- Correlation coefficient

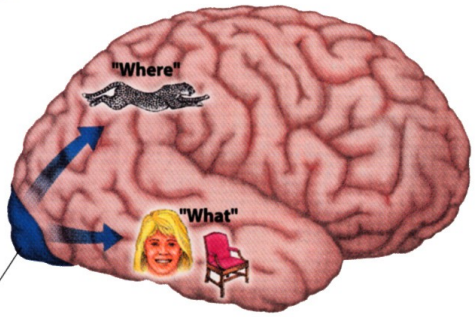
The Cortical Visual System: pathways



"What" and "Where" pathways

Hierarchically organised:

- simple, local, RFs at V1
- complex, large, RFs in higher areas



Where (or How):

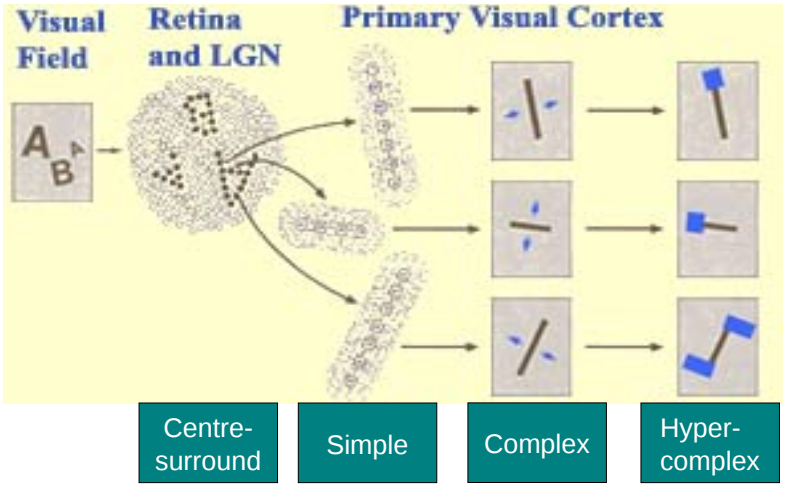
- V1 to parietal cortex
- spatial / motion information

What

- V1 to inferotemporal cortex
- identity / category information

Hierarchy of Receptive Fields

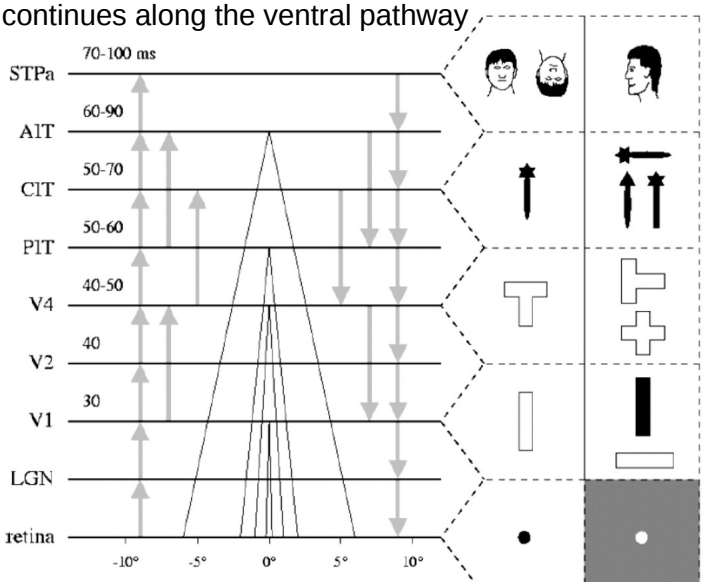
As we progress along a pathway, neurons' preferred stimuli gets more complex, receptive fields become larger, and there is greater invariance to location. e.g: LGN – V1



Hierarchy of Receptive Fields

This trend continues along the ventral pathway

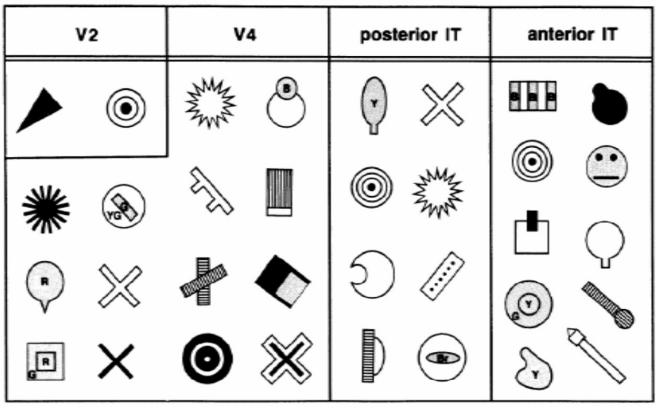
- larger RFs
- higher complexity
- higher invariance



Hierarchy of Receptive Fields

Neurons' preferred stimuli gets more complex but they have less sensitivity to location.

Neurons near the end of the ventral pathway respond to very complex stimuli, like faces.



Feedforward models of cortical hierarchy

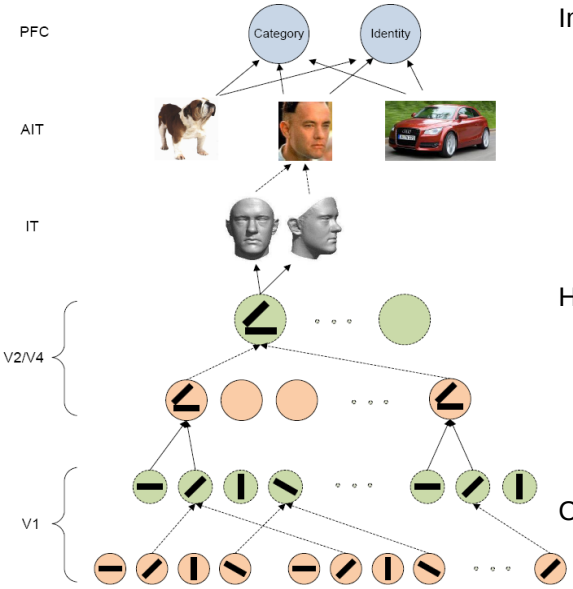
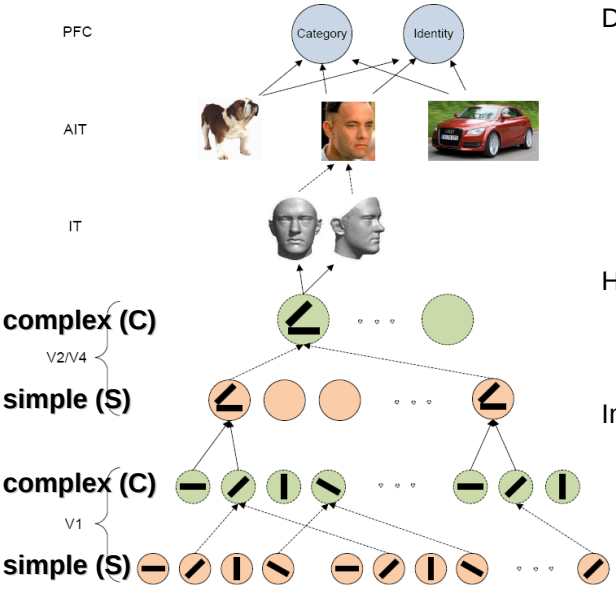


Image processed by layers of neurons with progressively more complex receptive fields at progressively less specific locations.

Hierarchical in that features at one stage are built from features at earlier stages.

Can be thought of as hierarchical template matching

Feedforward models: HMAX

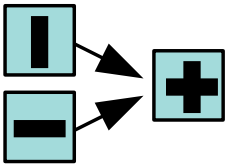
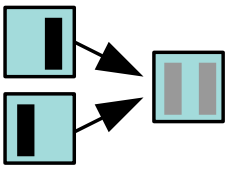


Different mathematical operations are required to increase the complexity or selectivity of RFs and to increase the degree of invariance of RFs.

Hence, several models use alternating layers of neurons with different properties.

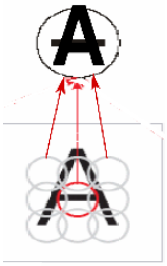
In analogy with the response properties of V1, these are called **simple (S)** and **complex (C)** cells.

Feedforward models: HMAX

Unit types		Computation	Result
Simple "S-cells"		sum "and"-like	Increased Selectivity
Complex "C-cells"		max "or"-like	Increased Invariance

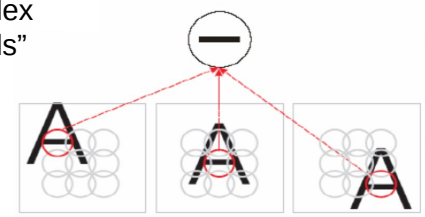
Feedforward models: HMAX

Simple "S-cells"



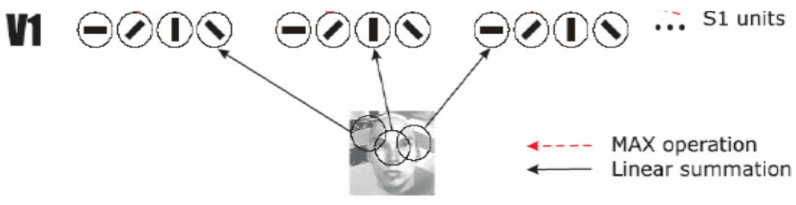
S-cells in one layer respond to conjunctions of C-cells in previous layer.

Complex "C-cells"

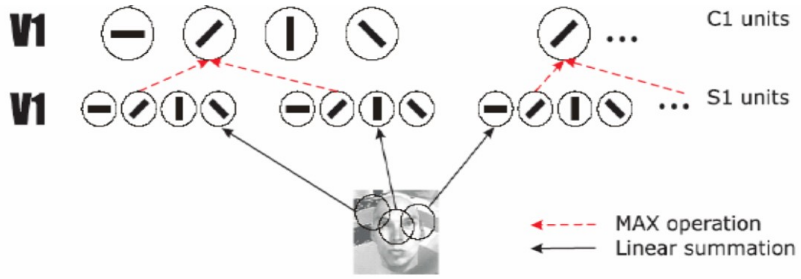


C-cells in one layer respond to any S-cell in a small neighborhood of the previous layer.

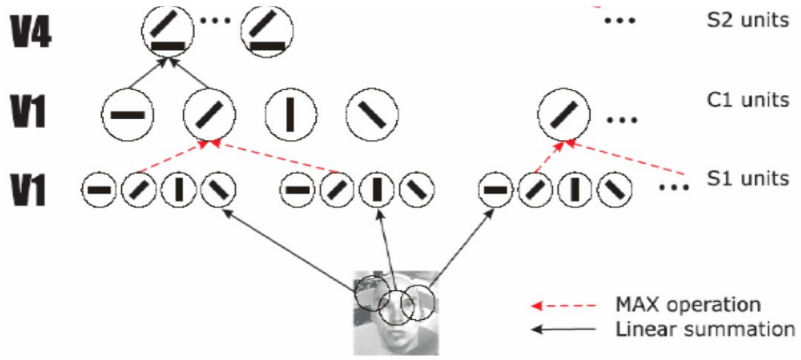
Feedforward models: HMAX



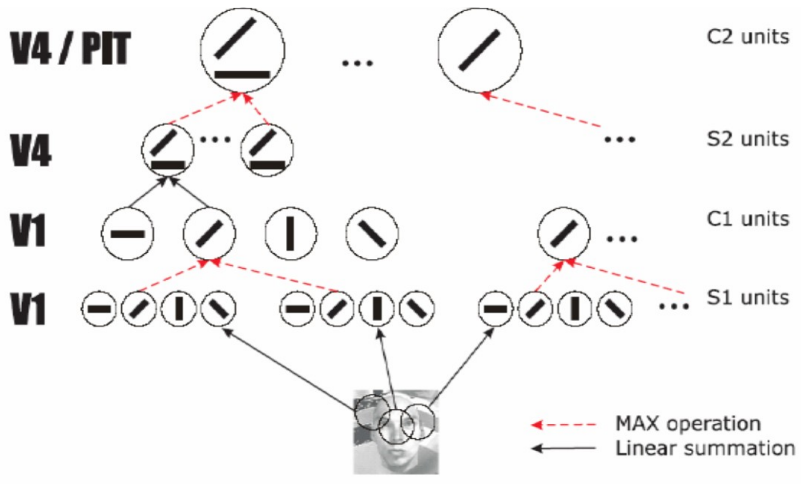
Feedforward models: HMAX



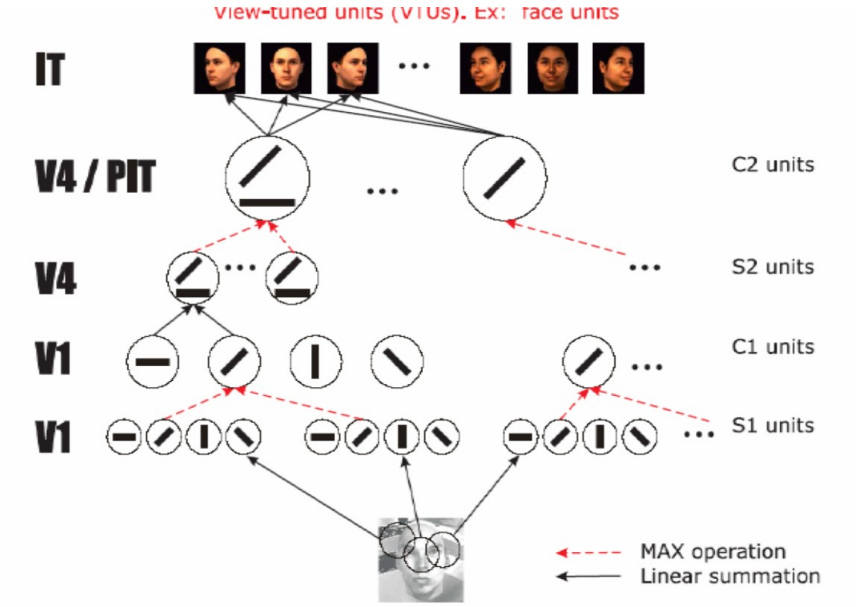
Feedforward models: HMAX



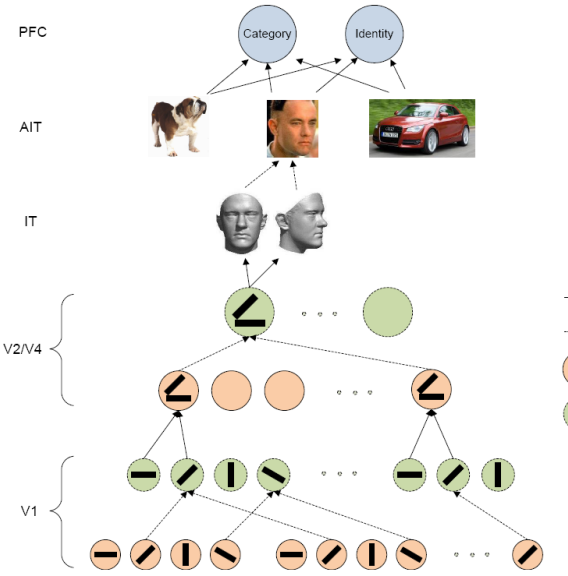
Feedforward models: HMAX



Feedforward models: HMAX



Feedforward models: HMAX



From IT to PFC:

Task-specific

Trained using supervised learning (i.e. a classifier)

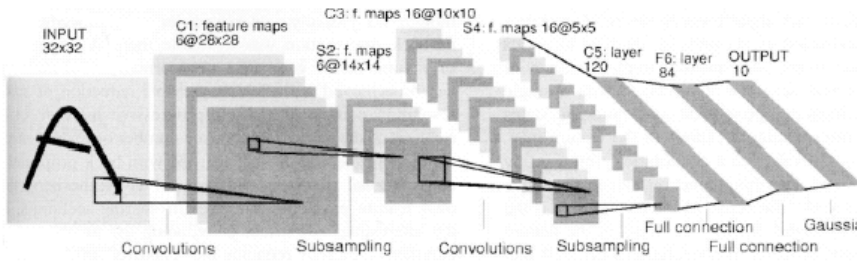
From V1 to IT:

Generic, overcomplete dictionary of reusable shape components.

"Hard-wired."

Feedforward models: CNN

CNN = convolutional neural network

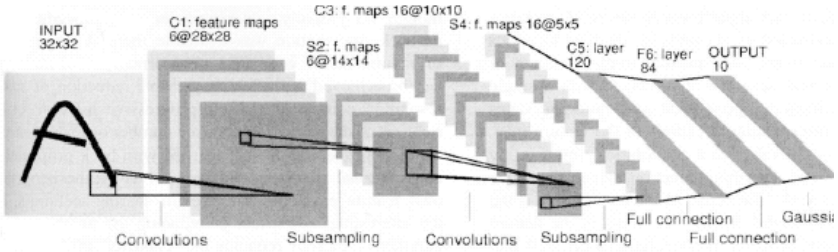


A hierarchical model similar to HMAX can be implemented using standard image processing techniques: convolution and sub-sampling.

It consists of alternating layers of

- convolution (equivalent to responding to conjunctions), and
- sub-sampling (equivalent to responding to any input in a small neighbourhood, to reduce location specificity.

Feedforward models: CNN

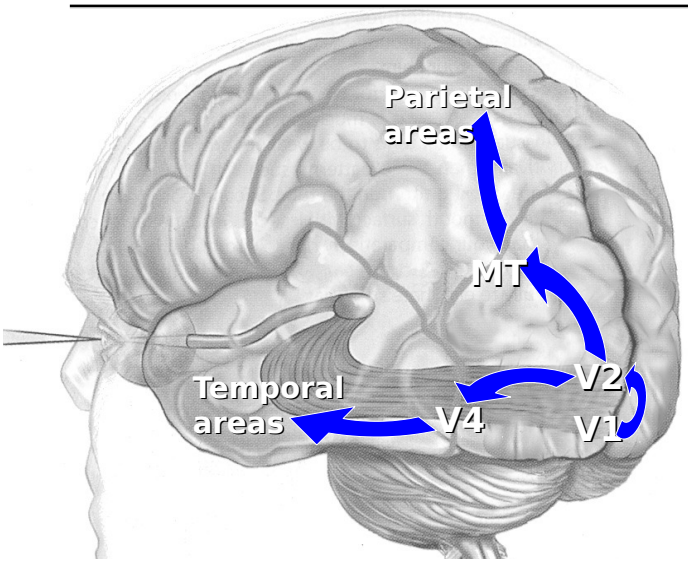


Confusingly:

- convolution layers are called “C layers” but are equivalent to S layers in HMAX, and
- sub-sampling layers are called “S layers” but are equivalent to C layers in HMAX.

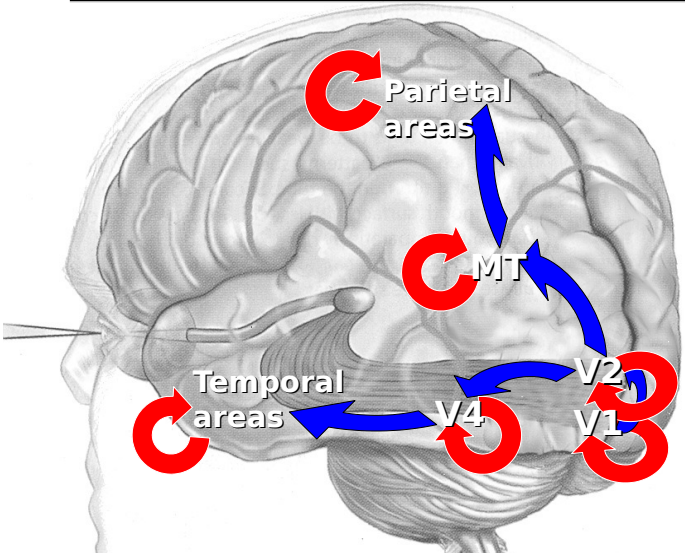
Deep Neural Networks (like HMAX and CNN) produce state-of-the-art image recognition performance.

Feedforward models of cortical hierarchy



HMAX and CNN are two examples of several models that propose a purely serial, feedforward, sequence of cortical information processing.

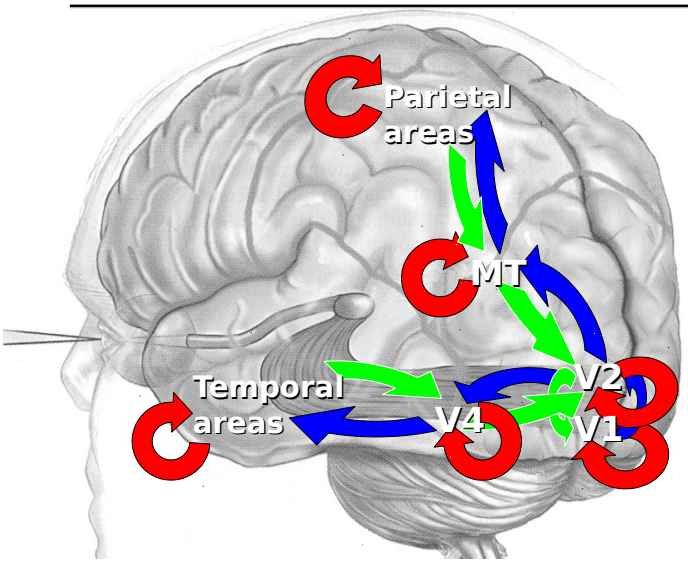
Recurrent models of cortical hierarchy



However, there are two types of recurrent connections.

(1) Within each region, lateral connections (both excitatory and inhibitory) enable neurons within the same population to interact (see descriptions of V1 and V2 in earlier lectures).

Recurrent models of cortical hierarchy



In addition,

(2) feedback connections convey information from higher cortical regions to primary sensory areas.

Bottom-up and top-down information interacts to affect perception.

Feedback models

Allow bottom-up and top-down information to be combined.

- **Bottom-Up processes:**
  - Using the information in the stimulus itself to aid in identification.
  - Stimulus driven.
- **Top-Down processes:**
  - Using context, previous knowledge, and expectation to aid in identification.
  - Knowledge driven.

Bayesian Inference

Bayes' Theorem describes an optimum method of combing bottom-up and top-down information.

Bayes' Theorem:

$$p(A|B)p(B) = p(B|A)p(A)$$

or

$$p(A|B) = p(B|A)p(A)/p(B)$$

p(A|B) is the **conditional probability** of A given B (vertical bar "|" reads as "given").

Bayesian Inference

An example of conditional probabilities.

The conditional probability that it is raining given that the pavement is wet is:

p(rain | wet pavement) < 1

because a wet pavement can be caused by many things (leaking pipes, dropped water bottles, etc).

The conditional probability that the pavement is wet given that it is raining is:

p(wet pavement | rain) = 1

because rain always wets the pavement.

Therefore, the two conditional probabilities are not necessarily equal

p(rain | wet pavement) ≠ p(wet pavement | rain)

Bayes' theorem gives the relationship between conditional probabilities.

Bayesian Inference

Bayes' theorem can be considered as a method for obtaining the information you need from the information you have.

In vision, we want to know **p(object$_j$ | Image$_i$)**: the probability that object$_j$ is present in the world given that image$_i$ is on the retina.

Solving this is hard – it is an inverse problem

Bayesian Inference

Bayes' theorem can be considered as a method for obtaining the information you need from the information you have.

In vision, we want to know **p(object$_j$ | Image$_i$)**: the probability that object$_j$ is present in the world given that image$_i$ is on the retina.

Solving this is hard – it is an inverse problem

However, what we know is **p(Image$_i$ | object$_j$)**: the probability of observing image$_i$ given the 3D object$_j$.

Solving this is easier – it is a forward problem

Bayes' theorem provides a means of calculating p(object$_j$ | Image$_i$) since:

$$p(\text{object}_j \mid \text{Image}_i) = p(\text{Image}_i \mid \text{object}_j)\, p(\text{object}_j) / p(\text{Image}_i)$$

Bayesian Inference: nomenclature

$$\underbrace{p(object_j \mid Image_i)}_{\text{posterior}} = \underbrace{p(Image_i \mid object_j)}_{\text{likelihood}} \underbrace{p(object_j)}_{\text{prior}} / \underbrace{p(Image_i)}_{\text{evidence}}$$

posterior: the thing we want to know (the probability of a particular object being present given the image).

likelihood: the thing we already know (the probability of the particular image being a projection of the particular object).

prior: the thing we know from prior experience (the probability that the particular object will be present in the environment)

evidence: the thing we can ignore, as it is the same for all possible interpretations of this image.

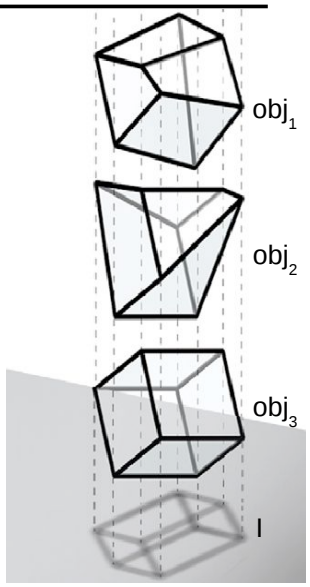
Bayesian Inference: example

Each of N=3 possible objects can generate the observed image.

The probability of observing this image, I, is the samefor all three possible objects, obj_j (make p(I)=1 for simplicity).

The likelihood $p(I|obj_j)$ is: “the probability of observing image I, given the 3D object obj_j”.

If all N=3 objects could produce the same image with equal probability, their likelihoods are the same:
$p(I|obj_1) = p(I|obj_2) = p(I|obj_3) = 0.09$



Bayesian Inference: example

Thus, the image alone cannot be used to decide which of the three possible objects produced the image.

However, if our prior experience of 3D objects produces a higher expectation of cubes than irregular shapes, then the priors will be different:
e.g. $p(obj_3) = 0.1$, $p(obj_2) = 0.01$, $p(obj_1) = 0.01$

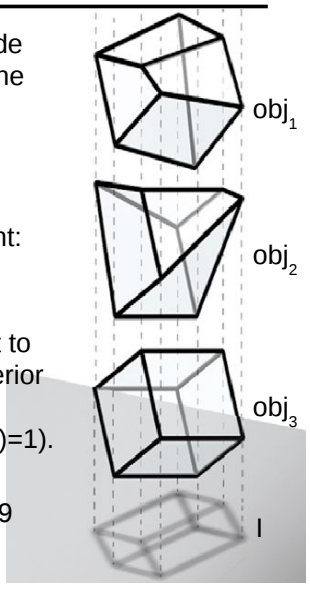
We can use the prior probability of each object to weight the known likelihood to obtain the posterior probability:

$p(obj_j|I) = p(I|obj_j)\ p(obj_j)$ (assuming p(I)=1).

Hence,

$p(obj_1|I) = p(obj_2|I) = 0.09x0.01 = 0.0009$

$p(obj_3|I) = 0.09x0.1 = 0.009$

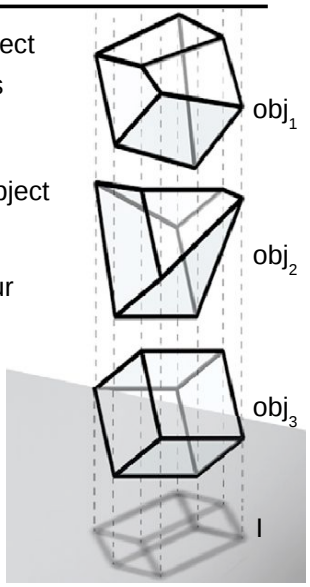


Bayesian Inference: example

The posterior $p(obj_j|I)$ is the probability that object obj_j is present in the world given that image I is on the retina.

The posterior probabilities thus tell us which object is most likely to have yielded image I.

In this example, the prior experience biases our interpretation of the image, so that we tend to interpret the image I as object obj_3.



Bayesian Inference

Bayes rule shows how to combine current evidence, I, with knowledge gained from prior experience, $p(obj_j)$, to estimate the posterior probability $p(obj_j|I)$ that the hypothesis (obj_j) under consideration is true (e.g. that obj_j is the correct 3D object).

Need to compute posterior $p(obj_j|I)$ for all possible hypotheses in order to select that hypothesis with the largest posterior.

If we assume p(I)=1 then

posterior = likelihood * prior

Bayesian Inference

Alternatively, if we just want to determine the probability that an image contains a particular object or not, we can use the following formulation:

$$\underbrace{\frac{p(object_j|image_i)}{p(not\,object_j|image_i)}}_{\text{posterior ratio}} = \underbrace{\frac{p(image_i|object_j)}{p(image_i|not\,object_j)}}_{\text{likelihood ratio}} \cdot \underbrace{\frac{p(object_j)}{p(not\,object_j)}}_{\text{prior ratio}}$$

Bayesian Inference: example

p(image | zebra) = 0.07
p(zebra) = 0.01

p(image | no zebra) = 0.0005
p(no zebra) = 0.99

$$\frac{p(zebra|image)}{p(no\ zebra|image)} = \frac{p(image|zebra)}{p(image|no\ zebra)} \cdot \frac{p(zebra)}{p(no\ zebra)}$$

$$= \frac{0.07}{0.0005}\frac{0.01}{0.99} = 1.41$$

>1 so zebra

Bayesian Inference: example

p(image | zebra) = 0.003
p(zebra) = 0.01

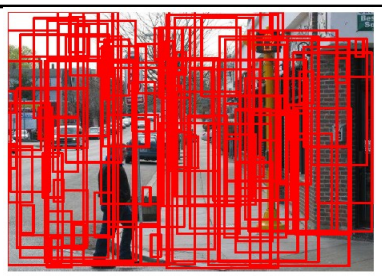
p(image | no zebra) = 0.85
p(no zebra) = 0.99

$$\frac{p(zebra|image)}{p(no\ zebra|image)} = \frac{p(image|zebra)}{p(image|no\ zebra)} \cdot \frac{p(zebra)}{p(no\ zebra)}$$

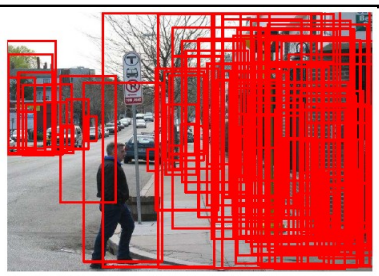
$$= \frac{0.003}{0.85}\frac{0.01}{0.99} = 0.000036$$

<1 so not zebra

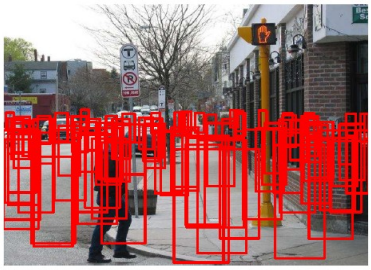
Bayesian Inference: example



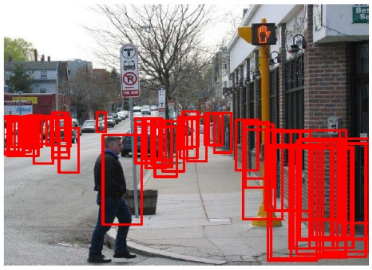
(b) P(person) = uniform



(d) P(person | geometry)



(f) P(person | viewpoint)



(g) P(person|viewpoint,geometry)

Bayesian Inference

Bayesian inference can be seen as a mathematical implementation of Helmholtz' Likelihood Principle (see lecture on segmentation):

The preferred perceptual organization of a sensory pattern reflects the most likely object or event.

i.e What we see is inferred from both the sensory input data and our prior experience.

Bayesian inference is the basis for many algorithms in Machine Learning and Pattern Recognition

Bayesian Inference

Bayesian inference can be seen as a method of solving the ill-posed, inverse problem of vision (see introductory lecture)

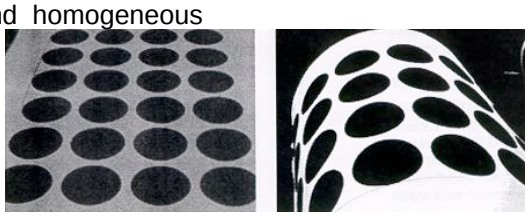
Vision is an inverse problem – we know the pixel intensities (the outcomes) and want to infer the causes (i.e. the objects in the scene, etc.).

Vision is ill-posed as there are usually multiple solutions (i.e. multiple causes that could give rise to the same outcomes).

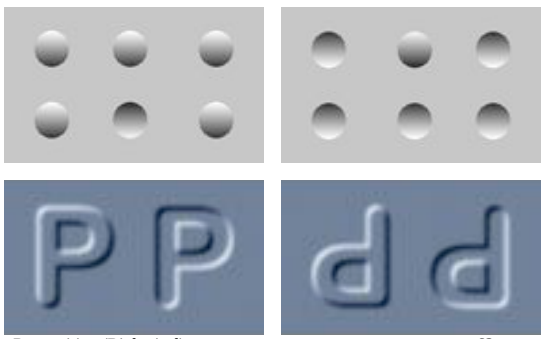
In order to compensate, the perceptual systems make use of assumptions, constraints or priors about the nature of the physical world.

Bayesian Inference

Prior: Texture is circular and homogeneous
Infer: shape/depth



Prior: Light from above
Infer: depth



Bayesian Inference

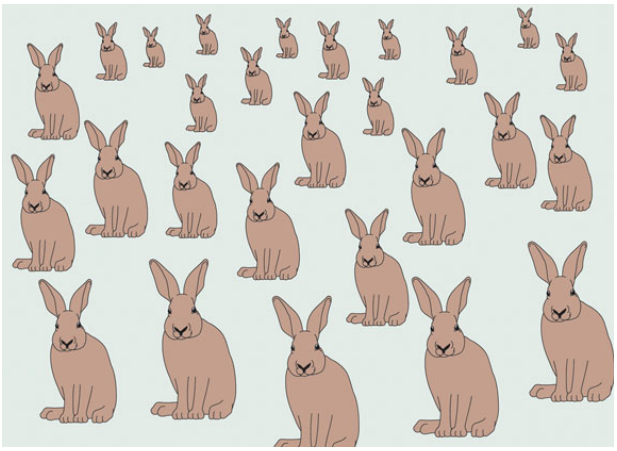
Prior: faces are convex
Infer: shape/depth

Convex face appears convex if assume light comes from above

Concave face still appears convex, but only if assume light now comes from below

Bayesian Inference

Prior: size is constant
Infer: depth



Bayesian Inference

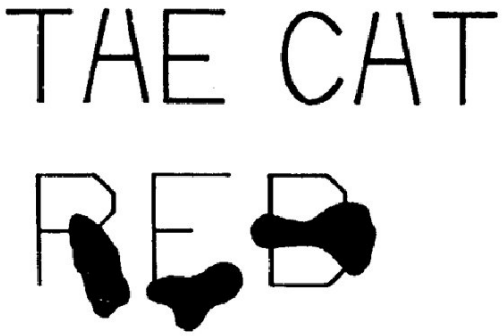
Prior: neighbouring features are related
Infer: grouping

Prior: similar features are related
Infer: grouping

Prior: connected features are related
Infer: grouping

Bayesian Inference

Prior: strings of letters form words
Infer: letter identity



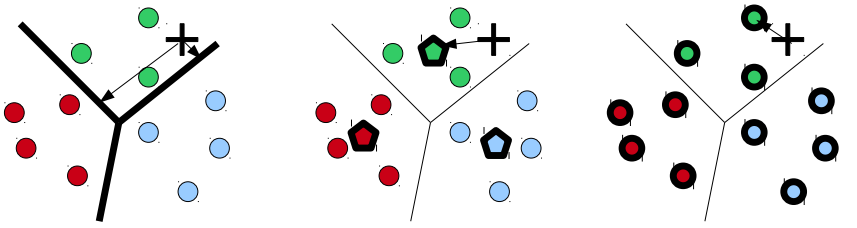
Bayesian Inference

Prior: knowledge about image content
Infer: object identity

We are back where we started in lecture 1!

Summary

rules vs prototypes vs exemplars



Decision Trees

Nearest Mean Classifier

Nearest Neighbour Classifier

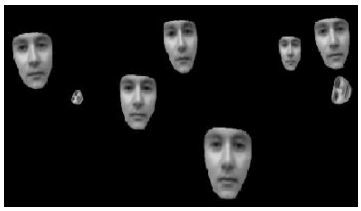
K-Nearest Neighbours Classifier

Summary

object-based (3D)
e.g. recognition by components

vs

image-based (2D)
e.g. template matching



configural (global)

vs

featural (local)

Summary

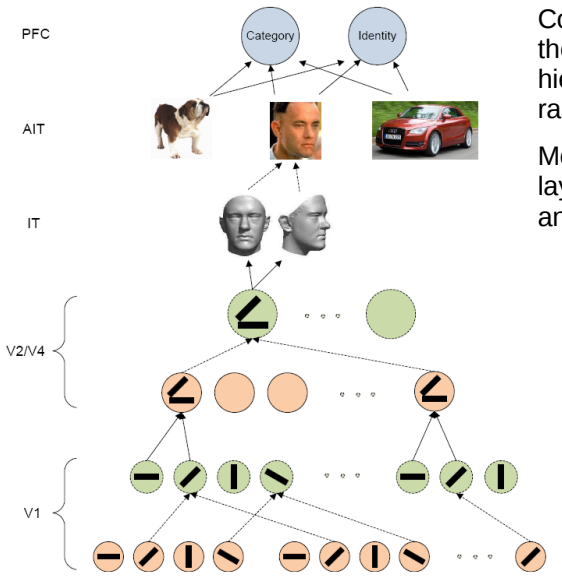
Local (featural) and global (configural) representations have complementary advantages and disadvantages

- simple (local) features generate many false positives:
  - fail to distinguish objects with similar features in different arrangements,
  - fail to deal with clutter
- complex (global) features generate many false negatives
  - fail to deal with occlusion
  - fail to deal with viewpoint changes and within class variation

Solutions:

1. use features of intermediate complexity
2. use a hierarchy of features with a range of complexities

Summary



Cortex seems to employ the latter approach: a hierarchy of features with a range of complexities.

Modelled using alternate layers increasing selectivity and increasing invariance.

Summary

- **Bottom-Up processes**
  - Using the information in the stimulus itself to aid in identification
  - Stimulus driven
  - Discriminative
- **Top-Down processes**
  - Using context, previous knowledge, and expectation to aid in identification
  - Knowledge driven
  - Generative

Summary

$$\underbrace{p(object_j \mid image_i)}_{\text{posterior}} = \underbrace{p(image_i \mid object_j)}_{\text{likelihood}} \underbrace{p(object_j)}_{\text{prior}} / \underbrace{p(image_i)}_{\text{evidence}}$$

$$\underbrace{\frac{p(object_j \mid image_i)}{p(not\ object_j \mid image_i)}}_{\text{posterior ratio}} = \underbrace{\frac{p(image_i \mid object_j)}{p(image_i \mid not\ object_j)}}_{\text{likelihood ratio}} \cdot \underbrace{\frac{p(object_j)}{p(not\ object_j)}}_{\text{prior ratio}}$$

- Discriminative methods model the posterior
- Generative methods model the likelihood and prior