

One and Zero-Shot Learning

Jeya Balasubramanian

jeya@pitt.edu

Intelligent Systems Program

20-Nov-2014

Introduction

- Can we learn a classifier that returns a class with very little evidence in the training dataset?
- One-shot learning: Optimizing classification with as little as one example for the class.
- Zero-shot learning: Optimizing classification with examples for a class omitted from the training data.

Fei-Fei et al., 2006

ONE-SHOT LEARNING OF OBJECT CATEGORIES

Image Recognition

- Recognize tens of thousands objects and categories into useful and informative taxonomies.
- It is often difficult and expensive to acquire large sets of training examples.
- Hypothesis: Once a few categories have been learned the hard way, some information can be abstracted from the process to make learning further categories more efficient.

Challenges

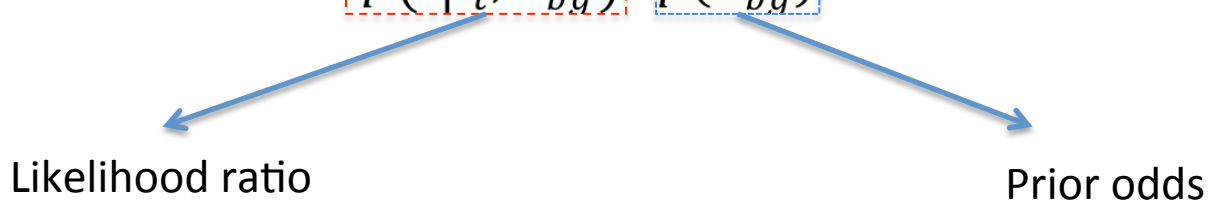
- Representation: How do we model objects and categories?
- Learning: How do we acquire such models?
- Recognition: Given a new image, how do we detect the presence of an object amongst clutter, occlusion, viewpoint, and lighting?

Requirements

- Models should be rich.
- Models should be flexible.
- Due to occlusion, not all features need to be detected.
- Learn model class variability from training examples.
- Must be computationally efficient.
- Learn with minimal supervision.

Decision: Is / an image of foreground?

$$R = \frac{p(O_{fg}|I, I_t)}{p(O_{bg}|I, I_t)}$$

$$= \frac{p(I|I_t, O_{fg}) \cdot p(O_{fg})}{p(I|I_t, O_{bg}) \cdot p(O_{bg})}$$


Likelihood ratio

Prior odds

O— Object category;

I— Image;

R— Ratio of the class posteriors;

T— Threshold for decision.

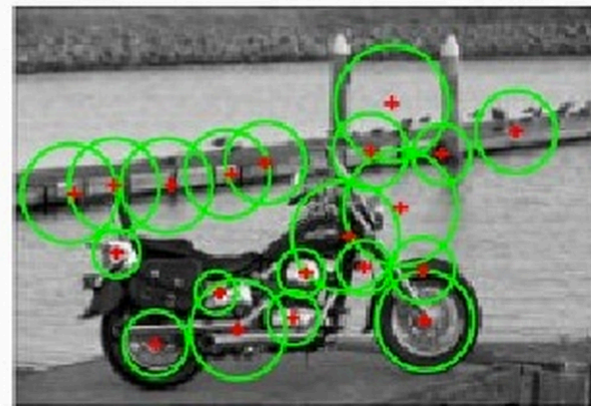
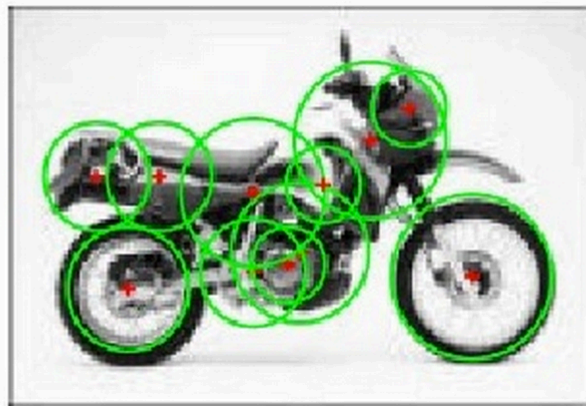
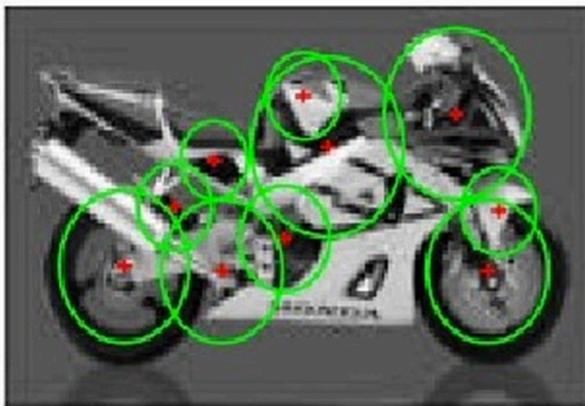
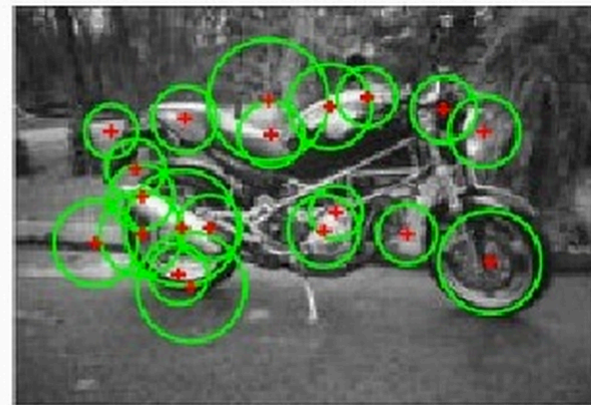
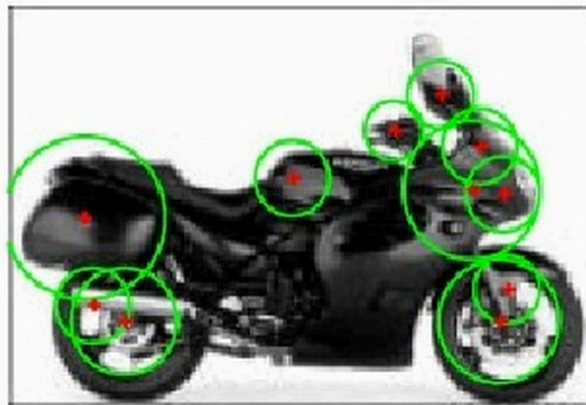
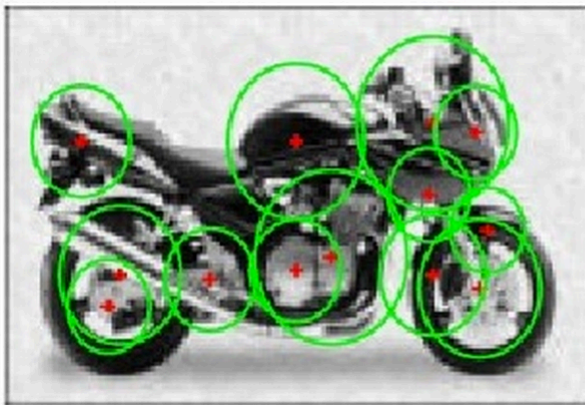
Introducing parametric models

$$R \propto \frac{\int p(I|\theta, O_{fg}) \cdot p(\theta|I_t, O_{fg}) \cdot d\theta}{\int p(I|\theta_{bg}, O_{bg}) \cdot p(\theta_{bg}|I_t, O_{bg}) \cdot d\theta_{bg}}$$
$$= \frac{\int p(I|\theta) \cdot p(\theta|I_t, O_{fg}) \cdot d\theta}{\int p(I|\theta_{bg}) \cdot p(\theta_{bg}|I_t, O_{bg}) \cdot d\theta_{bg}}$$

Learn this!

The ratio of priors is a constant and is omitted $\frac{p(O_{fg})}{p(O_{bg})}$.

Salient feature detection (Kadir and Brady)



Object category model (Constellation model)

From I we obtain N interesting regions in the image.

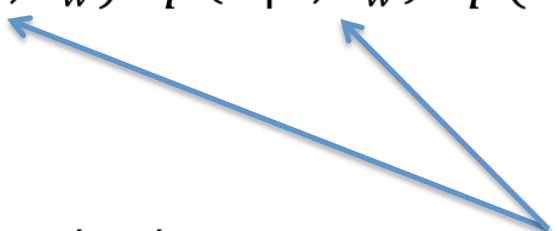
From the N regions we obtain: -

X – Locations of regions

A – Appearances of regions

$$\begin{aligned} R &\propto \frac{\int p(X, A | \theta, O_{fg}) \cdot p(\theta | X_t, A_t, O_{fg}) \cdot d\theta}{\int p(X, A | \theta_{bg}, O_{bg}) \cdot p(\theta_{bg} | X_t, A_t, O_{bg}) \cdot d\theta_{bg}} \\ &= \frac{\int p(X, A | \theta) \cdot p(\theta | X_t, A_t, O_{fg}) \cdot d\theta}{\int p(X, A | \theta_{bg}) \cdot p(\theta_{bg} | X_t, A_t, O_{bg}) \cdot d\theta_{bg}} \end{aligned}$$

Factorization of the likelihood for foreground

$$p(X, A | \theta) = \sum_{w=1}^{\Omega} \sum_{h \in H} p(X, A, h, w | \theta)$$
$$= \sum_{w=1}^{\Omega} p(w | \pi) \sum_{h \in H} \overset{\text{Appearance}}{p(A | h, \theta_w^A)} \cdot \overset{\text{Shape}}{p(X | h, \theta_w^X)} \cdot p(h | \theta_w)$$


Where $\theta = \{\pi, \theta^A, \theta^X\}$ and $p(h | \theta_w)$ are constant.
 X and A are assumed to be independent.

Flexible

Hypothesis h is an index variable for $P(3 \sim 7)$ features from $N(\text{up to } 100)$ interest points.

Factorization of the likelihood of the background

$$\begin{aligned} p(X, A | \theta_{bg}) &= p(X, A, h_0 | \theta_{bg}) \\ &= p(A | h_0, \theta_{bg}^A) \cdot p(X | h_0, \theta_{bg}^X) \cdot p(h_0 | \theta_{bg}) \end{aligned}$$

Where h_0 is the null hypothesis.

Appearance

For a given mixture component w ,
Each part p has a Gaussian density—

$$\theta_{p,w}^A = \{\mu_{p,w}^A, \Gamma_{p,w}^A\}$$

Background model—

$$\theta_{bg}^A = \{\mu_{bg}^A, \Gamma_{bg}^A\}$$

Where, μ is the mean, and Γ is the precision (inverse co-variance matrix)

Object Appearance—

$$p(A|h, \theta_w^A) = \prod_{p=1}^P G(A(h_p) | \mu_{p,w}^A, \Gamma_{p,w}^A) \prod_{j=1, j \neq h}^N G(A(j) | \mu_{bg}^A, \Gamma_{bg}^A)$$

Only background—

$$p(A|h_0, \theta_{bg}^A) = \prod_{j=1}^N G(A(j)|\mu_{bg}^A, \Gamma_{bg}^A)$$

Appearance odds—

$$\frac{p(A|h, \theta_w^A)}{p(A|h_0, \theta_{bg}^A)} = \prod_{p=1}^P \frac{G(A(h_p)|\mu_{p,w}^A, \Gamma_{p,w}^A)}{G(A(h_p)|\mu_{bg}^A, \Gamma_{bg}^A)}$$

Shape

A Gaussian density model represents shape after the locations of features are scale and translational-invariant.

Shape of object—

$$p(X|h, \theta_w^X) = \alpha^{-1} \cdot G(X(h)|\mu_w^X, \Gamma_w^X) \cdot \alpha^{-(N-P)}$$

Where $\theta_w^X = \{\alpha, \mu_w^X, \Gamma_w^X\}$

Shape of background—

$$p(X|h, \theta_{bg}^X) = \alpha^{-N}$$

Shape odds—

$$\frac{p(X|h, \theta_w^X)}{p(X|h, \theta_{bg}^X)} = \alpha^{P-1} \cdot G(X(h)|\mu_w^X, \Gamma_w^X)$$

Discussion of the model

- $X(h)$ is in $2P-2$ dimension and $A(h)$ is in kP dimension. If $k=10$, $P=4$; shape has 27 (6 mean, 21 full covariance matrix) parameters and appearance has 80 parameters (40 mean, 40 diagonal covariance matrix). A total of 107 parameters.
- Total number of hyper-parameters is 109. Both m and B have the same dimensionality as mean and covariance. In addition, there are a and b .

Discussion of the model (cntd.)

- The performance of the model depends upon the performance of the interest region detector.
- Patches may overlap leading to over-counting of evidence.
- The shape model has a data association complexity of $O(N^P)$.
- Different graph structures and co-ordinate frames may be more suited for compact objects (eg. human bodies).
- If the outline of the object is more important than texture (eg. bottle) alternative representations (eg. Curve contour) may be more suited.

Discussion of the model (cntd.)

- The background model is very simple: a uniform distribution and a single Gaussian distribution for appearance. Empirically this assumption was found to be reasonable.
- The model can be extended to perform object localization by bounding the best hypothesis.
- The paper assumes a single mixture component. Increasing the number of components can model different aspects of the object (eg. Pose variations).

Form of the parameter posterior

$$\int p(X, A | \theta) \cdot p(\theta | X_t, A_t, O_{fg}) \cdot d\theta$$

How do we estimate the parameter posterior $p(\theta | X_t, A_t, O)$?

1. Maximum Likelihood:

If we assume model distribution is highly peaked, the integral reduces to—

$$p(X, A | \theta^*)$$

$$\theta^* = \theta^{ML} = \underset{\theta}{\operatorname{argmax}} p(X_t, A_t | \theta)$$

2. Maximum a Posteriori:

If we have some idea about the parameter prior—

$$\theta^* = \theta^{MAP} = \underset{\theta}{\operatorname{argmax}} p(X_t, A_t | \theta) \cdot p(\theta)$$

Other Inference methods

- Sampling methods: Gibbs sampling or Markov chain-Monte Carlo can give accurate estimate of the integral. But this approach is computationally expensive.
- Recursive approximations: Variational approximations by sequentially processing the data points to approximate the parameter posterior.
- Conjugate densities: Use Normal-Wishart distribution which is conjugate prior to the Multivariate Gaussian distribution. The integral becomes a multivariate Student's T distribution.

Conjugate density: Parameter distribution

The mixture of constellation models—

$$p(X, A | \theta) = \sum_{w=1}^{\Omega} p(w | \pi) \sum_{h \in H} p(X(h) | \mu_w^X, \Gamma_w^X) \cdot p(A(h) | \mu_w^A, \Gamma_w^A)$$

π is the mixing coefficient of w ;

μ_w^X, μ_w^A is the mean of shape and appearance.

Γ_w^X, Γ_w^A is the precision of shape and appearance.

Parameter vector, $\theta = \{\pi, \mu^X, \mu^A, \Gamma^X, \Gamma^A\}$

$$p(\theta | X_t, A_t) = p(w) \prod_w p(\mu_w^X | \Gamma_w^X) \cdot p(\Gamma_w^X) \cdot p(\mu_w^A | \Gamma_w^A) \cdot p(\Gamma_w^A)$$

Where

$p(w)$ is a symmetric Dirichlet = $Dir(\lambda_w I_{\Omega})$

$p(\mu_w^X | \Gamma_w^X)$ is Normal = $G(\mu_w^X | m_w^X, \beta_w^X \Gamma_w^X)$

$p(\Gamma_w^X)$ is a Wishart = $W(\Gamma_w^X | a_w^X, B_w^X)$

Conjugate density: Closed-form calculation for R

$$R \propto \frac{p(X, A | X_t, A_t, O_{fg})}{p(X, A | X_t, A_t, O_{bg})}$$
$$= \frac{\int p(X, A | \theta) \cdot p(\theta | X_t, A_t, O_{fg}) \cdot d\theta}{\int p(X, A | \theta_{bg}) \cdot p(\theta_{bg} | X_t, A_t, O_{bg}) \cdot d\theta_{bg}}$$

$$p(X, A | X_t, A_t, O_{fg}) = \sum_{w=1}^{\Omega} \sum_{h=1}^{|H|} \tilde{\pi}_w \cdot S(X_h | g_w^X, m_w^X, \Lambda_w^X) \cdot S(A_h | g_w^A, m_w^A, \Lambda_w^A)$$

Where:

$S(\cdot)$ is a multimodel multivariate Student's T distribution

$$g_w = a_w + 1 - d;$$

$$\Lambda_w = \frac{\beta_w + 1}{\beta_w g_w} \cdot B_w;$$

$$\tilde{\pi}_w = \frac{\lambda_w}{\sum_{w'} \lambda_{w'}}.$$

Implementation: Feature selection/ representation

- Learning is done by Variational Bayesian Expectation Maximization (VBEM).
- Features are found using the Kadir and Brady approach.
- The co-ordinates of the center of the feature gives X .
- The region is then cropped from the image, re-scaled to 11x11 pixels (121 dimensional space), and reduced by PCA. The co-efficients from the principal components give A .

Implementation: Learning

- Learn a single mixture component ($\Omega = 1$). So, $\pi_{\omega} = 1$, and therefore $\lambda = 1$.
- Parameters for shape/appearance mean and variance is estimated by object models learned from three object categories (spotted cats, faces, airplanes) using ML to learn priors.
- The parameters from this existing category is used to estimate the hyper-parameters of the new category.

Details of the Bayesian One-Shot algorithm

- Shape/Appearance means are initialized by the means of the training data. Covariance is randomly chosen within reasonable range.
- Stopping criterion:
 - Largest parameter change is less than a threshold (eg. 10^{-4}).
 - Exceed maximum number of iterations (eg. 500).
- Background images are not used during learning.
- Learning a category takes less than a minute when the number of images is less than 10. The model has 4 parts. Increasing the parts improves recognition.

Experimental results

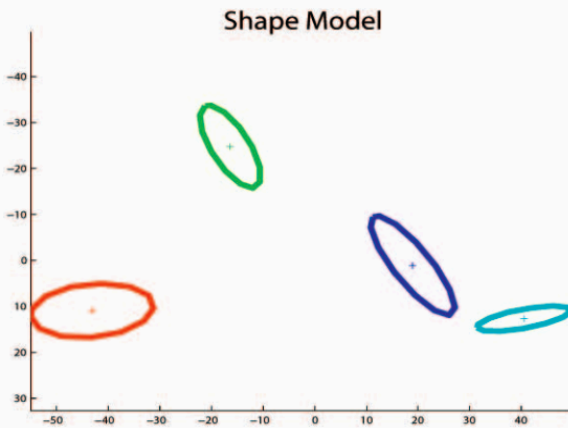
- Caltech 4 dataset: human faces, motorbikes, airplanes, and spotted cats.
- Naïve 97 dataset: 97 new dictionary term categories from Google Image Search.
- Preprocessing: Remove irrelevant images, all instances face the same direction, vertical structures are rotated.

Experimental setup

- Dataset is split into two disjoint sets. N (1-6 per category) samples are randomly drawn from the first set and assigned as training data. 50 samples are randomly drawn from the second set and assigned as test data. Additional 50 images as background.
- Number of runs = 10.
- Learning approaches: ML, MAP, and Variational Bayesian approach (Prior is obtained from three object categories).
- One mixture component.
- Parts in the model, $P = 4$.
- PCA dimensions = 10.
- Average number of interest point detections per image = 20.

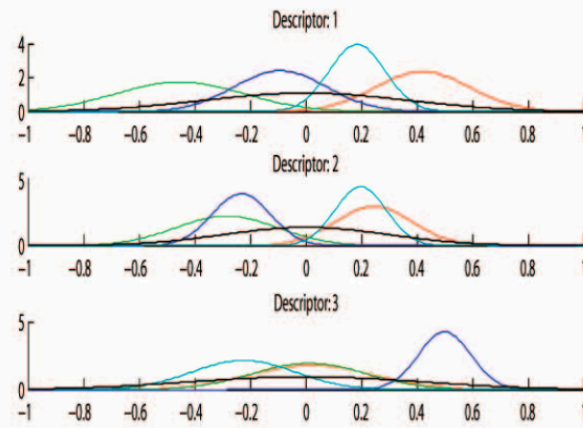
Motorbike detection

Gaussian densities

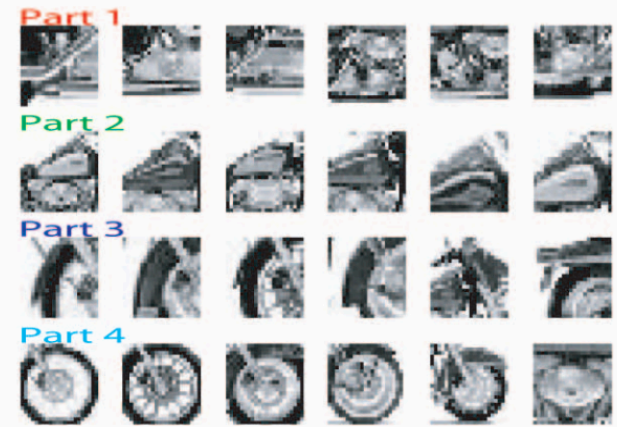


(a)

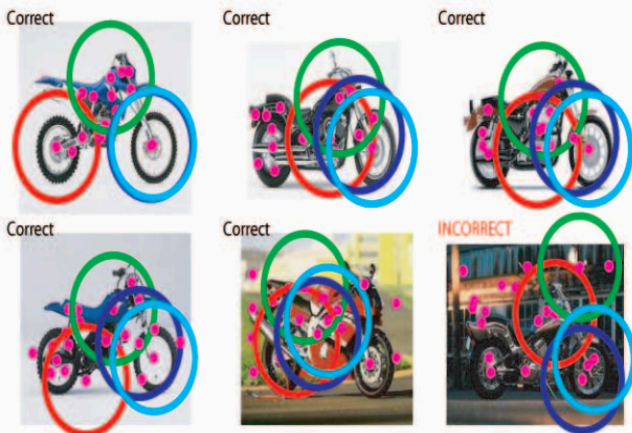
3 principal components



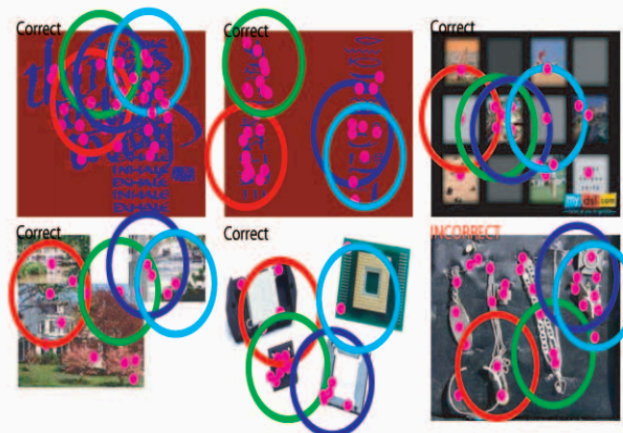
(b)



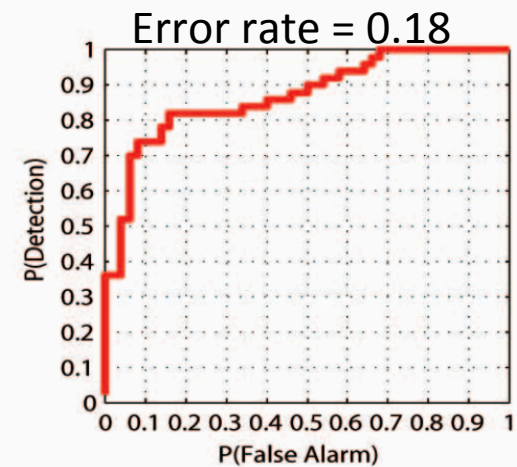
(c)



(d)

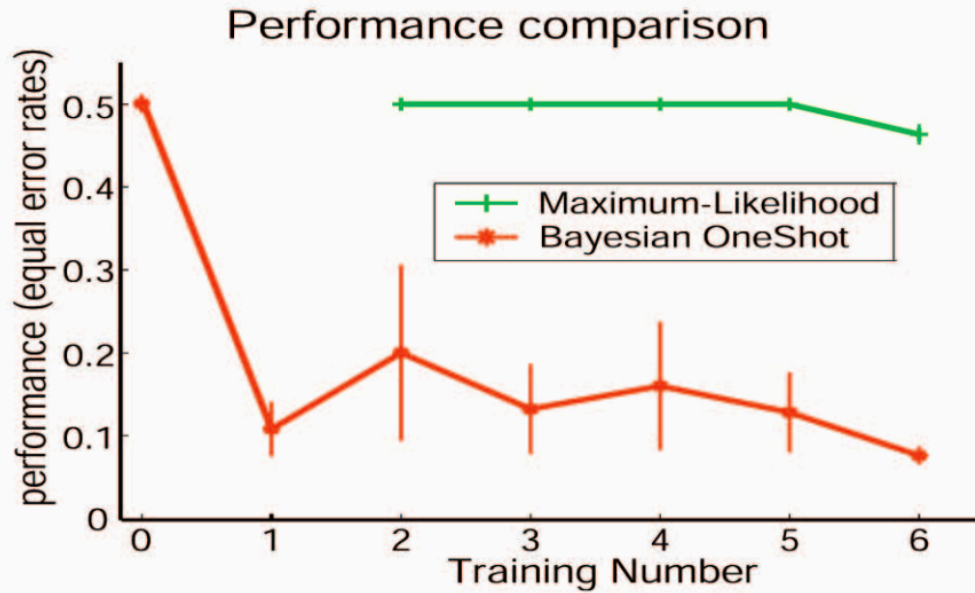


(e)

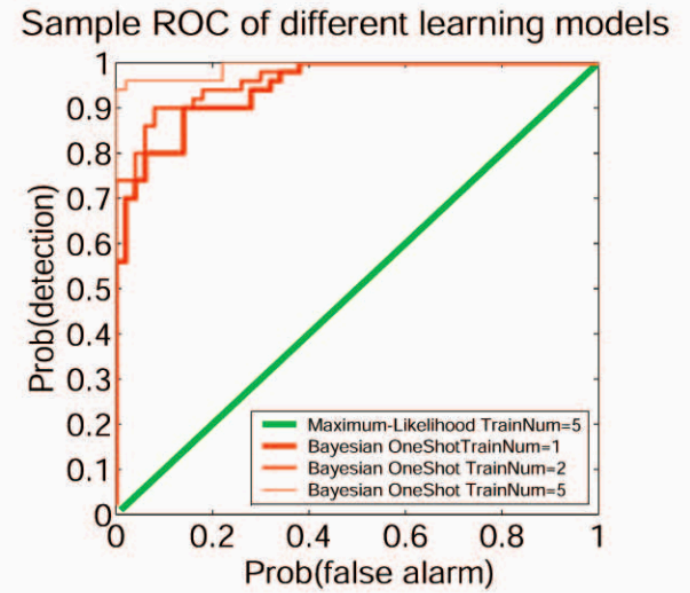


(f)

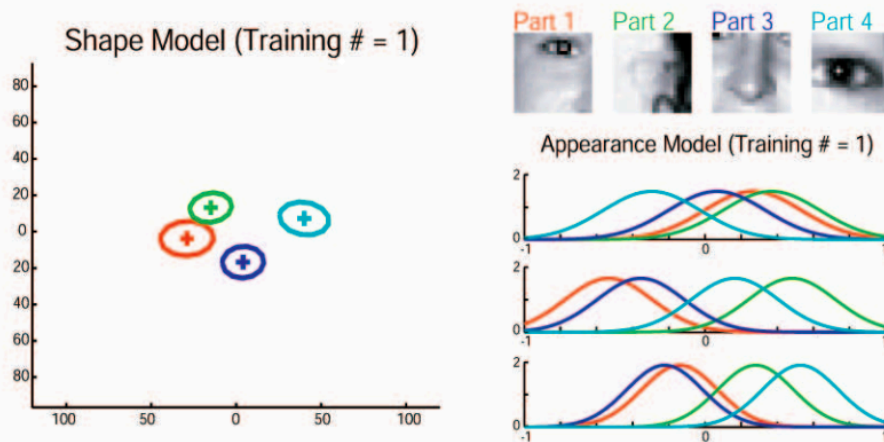
Face detection



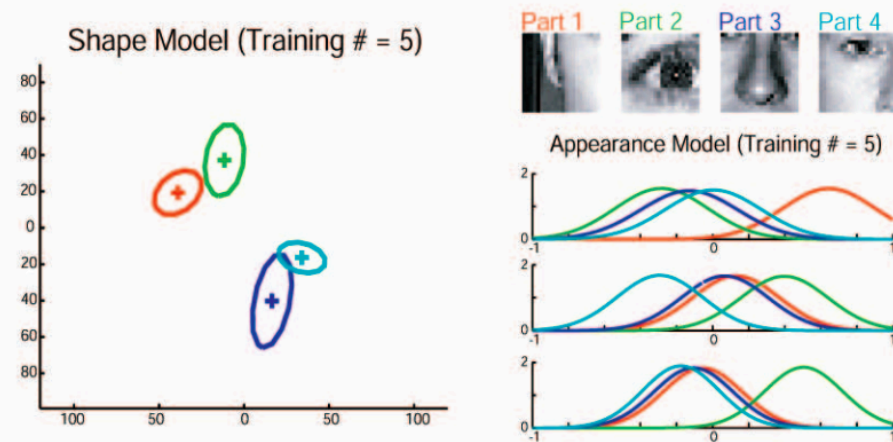
(a)



(b)



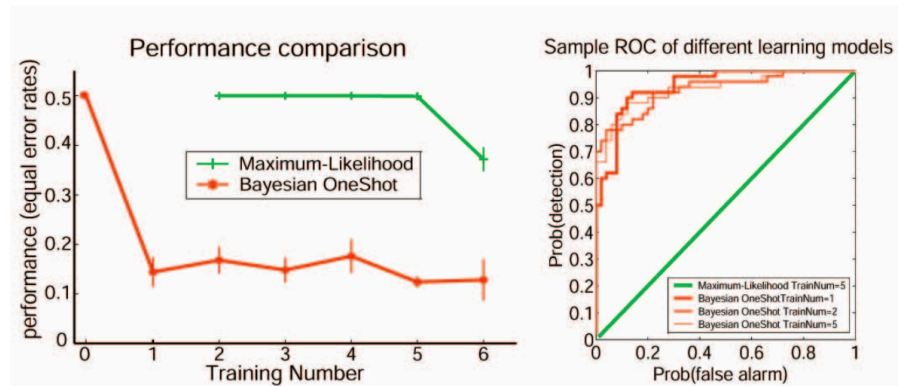
(c)



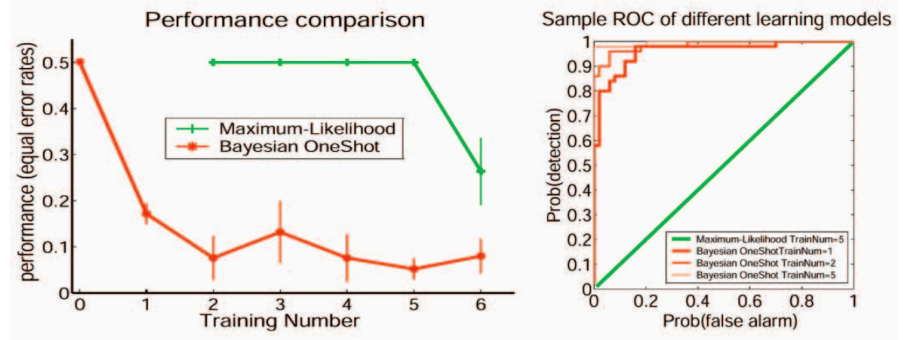
(d)

Performance on motorbike, spotted cat, and airplane

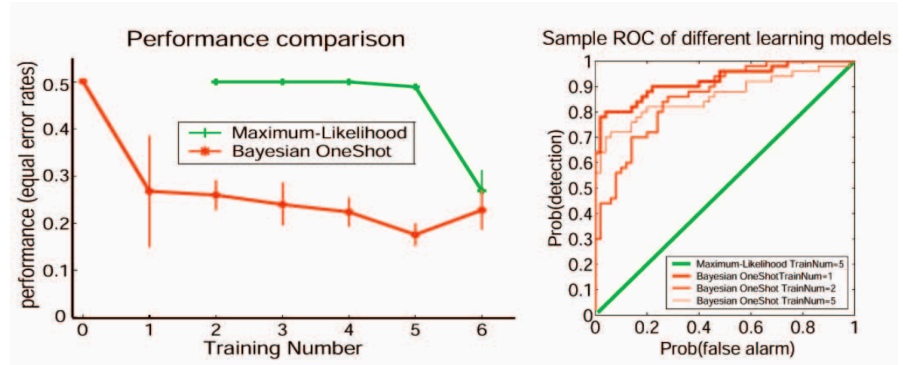
Motorbike



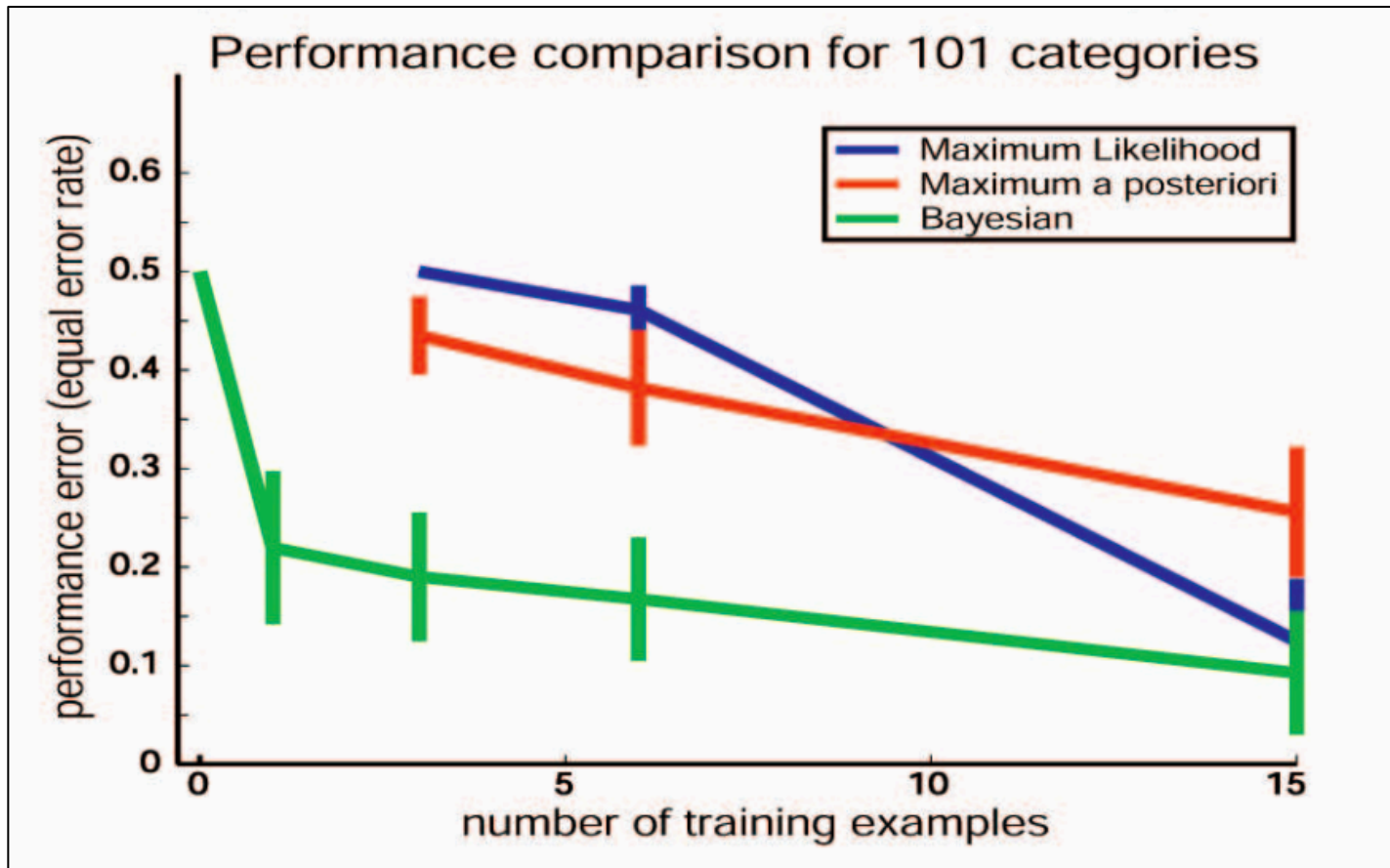
Spotted cats



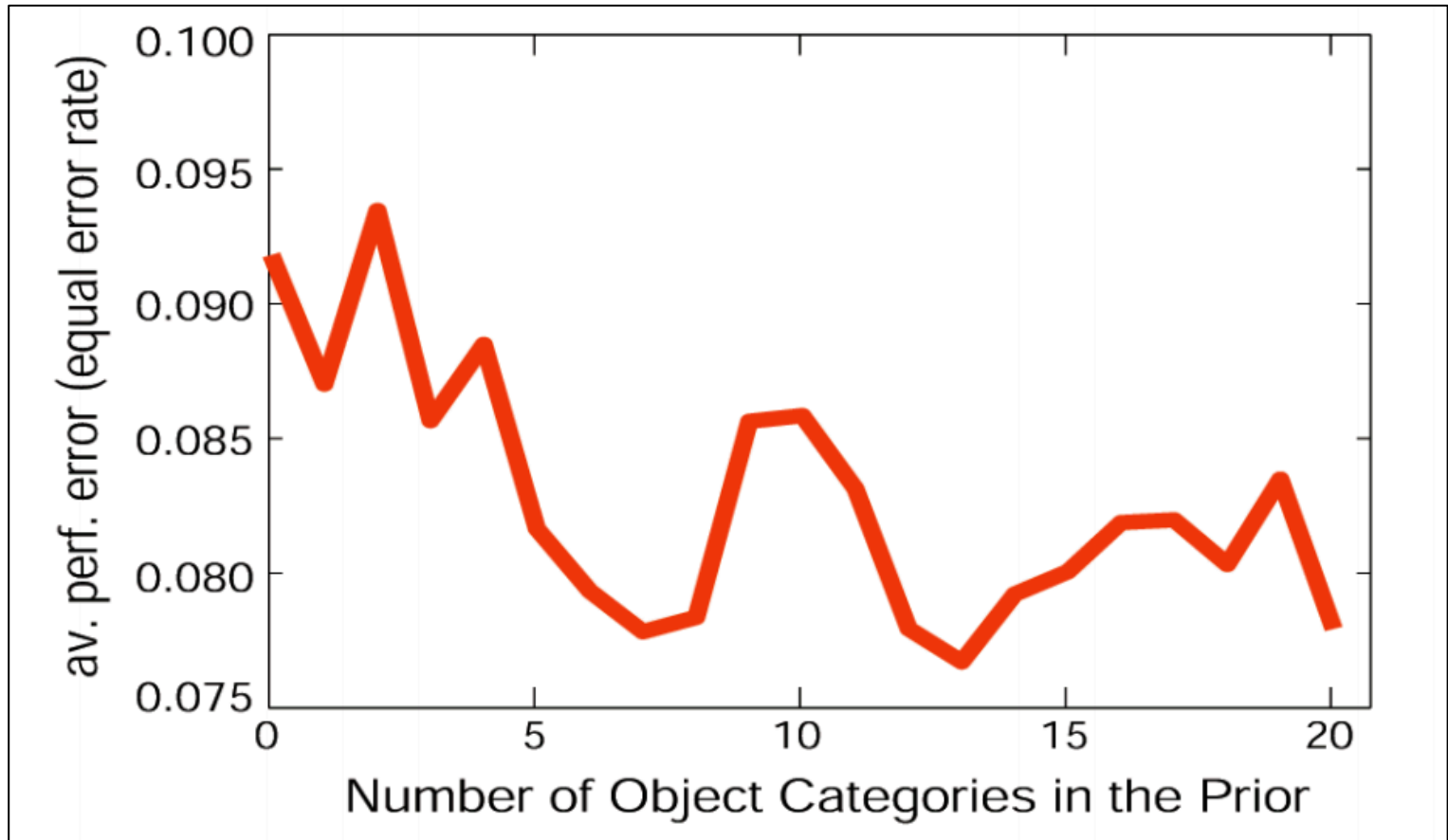
Airplane



Impact of priors: ML, MAP, Bayes in 101 categories



Effect of the number of categories used to learn the prior model

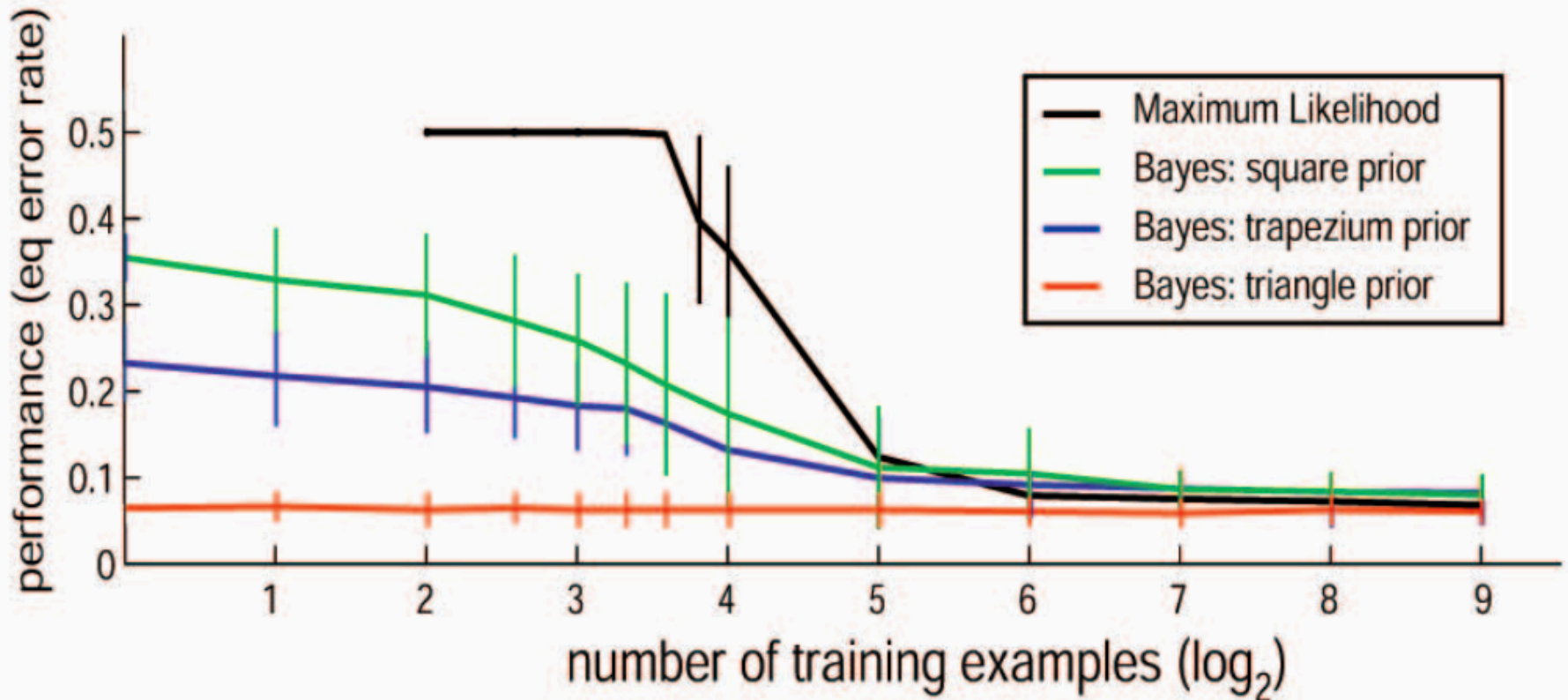


A few experimental observations

- With larger number of training examples per category, the shape was more well defined.
- The performance of the recognition algorithm is sensitive to the feature detection algorithm.
- Shape-only, Appearance-only approaches work for certain category models. Overall performance is best when Shape and Appearance is combined.

Impact of different priors

performance comparison for learning a triangular model



Limitations

- Model does not account for occlusion.
- Number of parts in the object is limited to 4.
- Priors are learned from only three categories.
- Only one component constellation model was evaluated.

Palatucci et al., 2011

ZERO-SHOT LEARNING WITH SEMANTIC OUTPUT CODES

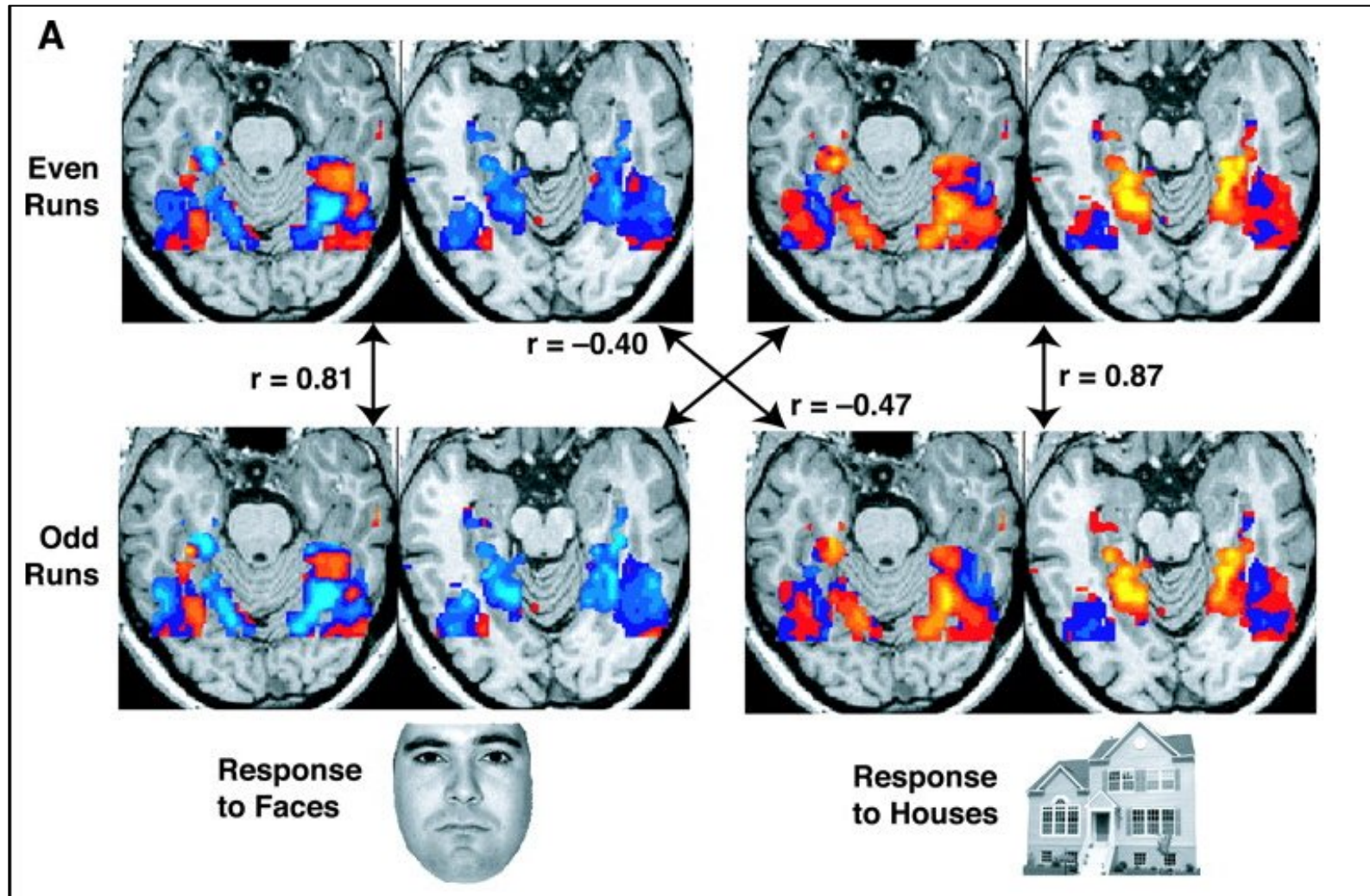
Zero-shot learning

- Goal: Learn a classifier $f: X \rightarrow Y$, which can predict novel values of Y omitted from the training set.
- Application:
 - Y takes very large number of values.
 - Labeling all of Y is expensive.

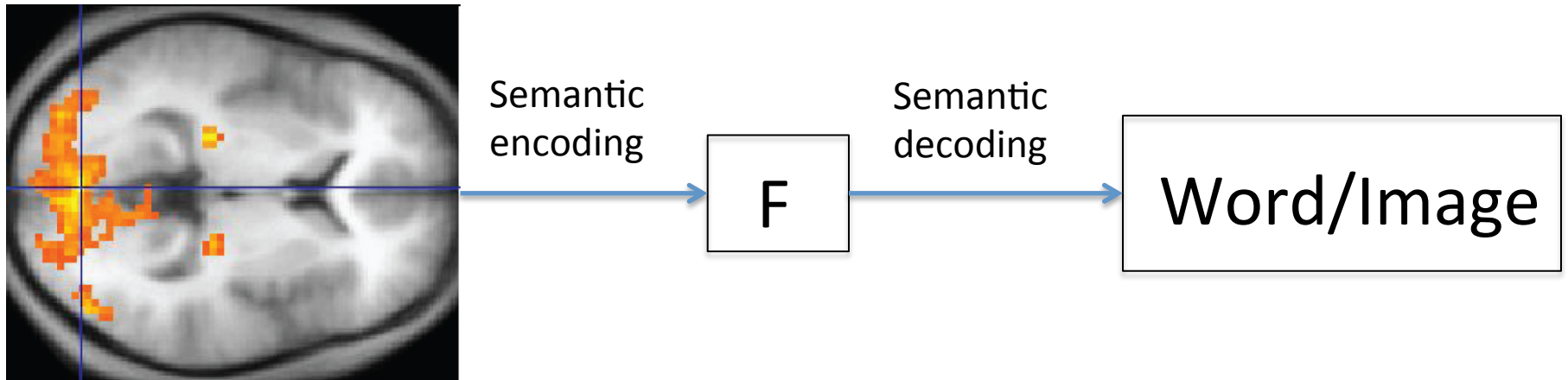
Problem domain

- Object recognition: Several tens of thousands of objects to recognize.
- Automatic speech recognition: Recognize words without training for all words.
Vocabulary independence can be achieved by a phoneme-based recognition strategy.
- Neural activity decoder: Determine the word or object based on neural activity.

fMRI response to words/images

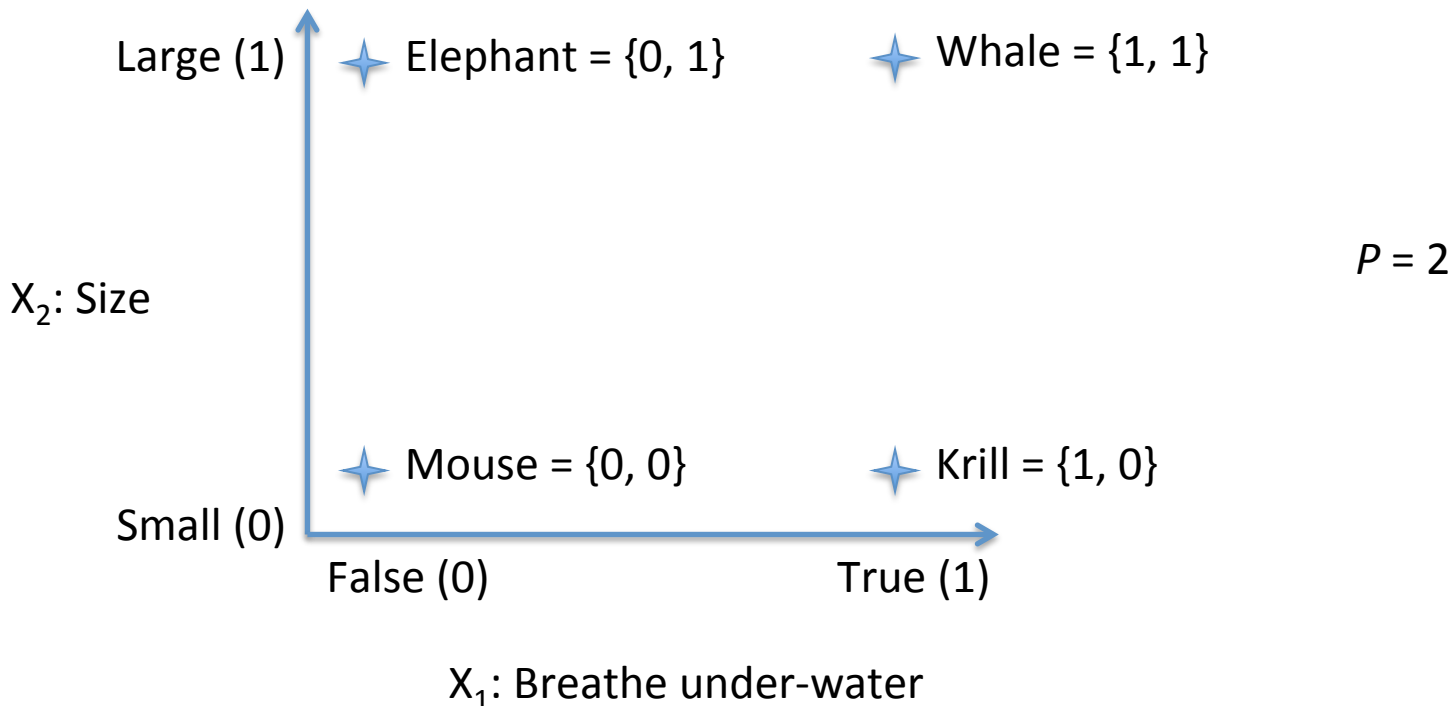


Semantic encoding/decoding



Semantic feature space

- Metric space, F^p , of p dimensions that encodes the values of the semantic properties.

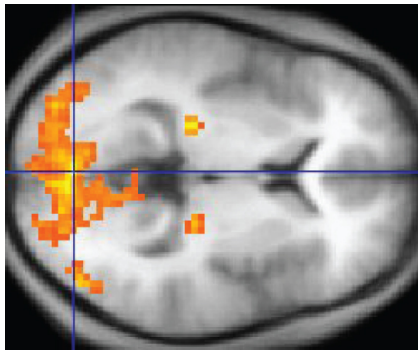


Semantic knowledge base

- Knowledge base K : M examples of pairs $\{f, y\}$
1:M.
- Point f is in semantic feature space F^p .
- Class label y is from set Y .

Semantic output code classifier, H

Input: X^d



Semantic
encoding

\mathcal{S}

Feature space:

F^p

Is it furry?
Does it have a tail?
Can it breathe under-
water?
Is it a carnivore?
Is it slow moving?

Semantic
decoding

\mathcal{L}

Label: y

- Dog
- Whale
- Elephant

$$\mathcal{H} = \mathcal{L}(\mathcal{S}(\cdot))$$

$$\mathcal{S} : X^d \rightarrow F^p$$

$$\mathcal{L} : F^p \rightarrow Y$$

Zero-shot: Semantic Output Code Classification

- Input: Training data D a d -dimensional matrix with N instances of form $\{x, y\}_{1:N}$. Semantic encoding of a large set of concept classes, K with M examples of form $\{f, y\}_{1:M}$. Typically, $M \gg N$.
- Output: Classifier to recognize classes including those omitted from the training dataset.

Model

Training data, $D = \{X, Y\}$

$X_{N \times d}$ with N words and d dimensions.

Dimensions are reduced by *voxel stability criterion*.

$Y_{N \times p}$ with N words and p semantic features.

Learning:

$$\hat{W}_{d \times p} = (X^T X + \lambda I)^{-1} \cdot X^T Y$$

Predict:

Stage 1: $\hat{f} = x \cdot \hat{W}$

Stage 2: $\mathcal{L}(\hat{f})$ 1-Nearest Neighbor
with Euclidean distance

Under what conditions does the SOC classifier predict labels omitted in the training data?

$$d(q, r_y) = \text{distance between } q = S(x) \text{ and the semantic representation } r_y \text{ of some class } y \quad (1)$$

$$R_q(z) = P(d(q, r_y) \leq z), \text{ z some arbitrary distance} \quad (2)$$

$$\eta_q = \text{distance of q to its nearest neighbor} \quad (3)$$

$$G_q(z) = P(\eta_q \leq z) = 1(1 - R_q(z))^n, \text{ given n points in F that are considered} \quad (4)$$

$$\tau_q = \text{distance of q to its true class representation} \quad (5)$$

$$P(\eta_q \leq \tau_q) \leq \gamma = \text{maximum risk of another class being equidistant or closer then the true class} \quad (6)$$

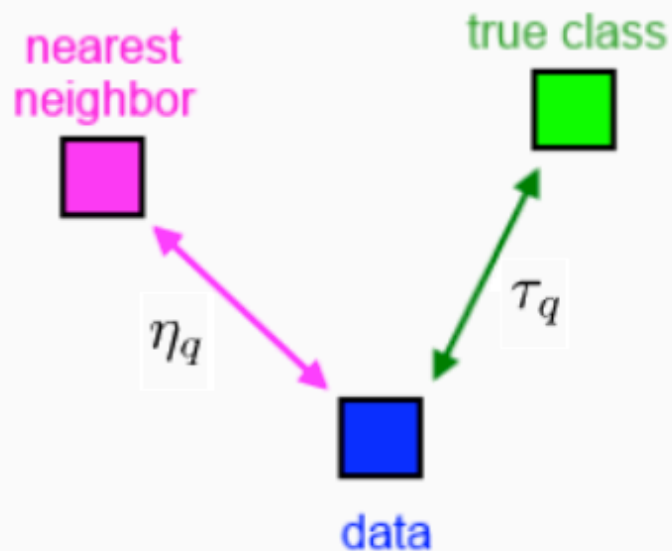
$$\rightarrow G_q(\tau_q) \leq \gamma \quad (7)$$

$G_q(.)$ not necessarily invertible, so define a pseudo inverse:

$$G_q^{-1}(\gamma) = \underset{\tau_q}{arg \max} (G_q(\tau_q) \leq \gamma) = \tau_q^{max} \quad (8)$$

semantic feature space

$$P(\eta_q \leq \tau_q) \leq \gamma$$



τ_q^{max} being the maximum number of errors in the prediction $S(x)$ to achieve the bound γ .

Given p independently learned classifiers, each is supposed to have true error $\epsilon = \frac{\tau_q^{max}}{p}$

$$P(\text{errors in } S(x) \leq \tau_q^{max}) = BinoCDF(\tau_q^{max}; p, \epsilon) \quad (9)$$

$$P(\text{classifier has true error} \leq \epsilon) = 1 - \delta \quad (10)$$

All sums up in the number M of needed examples per classifier to feasibly assure the bounds:

$$M \geq \tau_q^{max} \left[4 \log \left(\frac{2}{\delta} \right) + 8(d+1) \log \left(\frac{13p}{\tau_q^{max}} \right) \right] \quad (11)$$

with d and p the dimensions of x and $q = S(x)$ respectively.

$$P(\text{predict the right } S(x)) = (1 - \delta)^p BinoCDF(\tau_q^{max}; p, \epsilon) (1 - \gamma) \quad (12)$$

Experiments

- Dataset: fMRI dataset with 9 human participants. Words representing 12 categories (5 examples each) to a total of 60 words.
- Neural activity is measured in 20,000 locations (voxels) in the brain. Six MRIs are taken for each word. These are time-averaged.
- There are two Knowledge base—
 - corpus5000 (Google Trillion-Word-Corpus) containing co-occurrence vector for words.
 - Human218 Mturk annotated 218 features for the 60 words, scaled from 1-5.

Can the classifier discriminate between two classes, neither of which is in the training data?

Table 1: Percent accuracies for leave-two-out-cross-validation for 9 fMRI participants (labeled P1-P9). The values represent classifier percentage accuracy over 3,540 trials when discriminating between two fMRI images, both of which were omitted from the training set.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	Mean
corpus5000	79.6	67.0	69.5	56.2	77.7	65.5	71.2	72.9	67.9	69.7
human218	90.3	82.9	86.6	71.9	89.5	75.3	78.0	77.7	76.2	80.9

How does the classifier discriminate closely related novel classes?

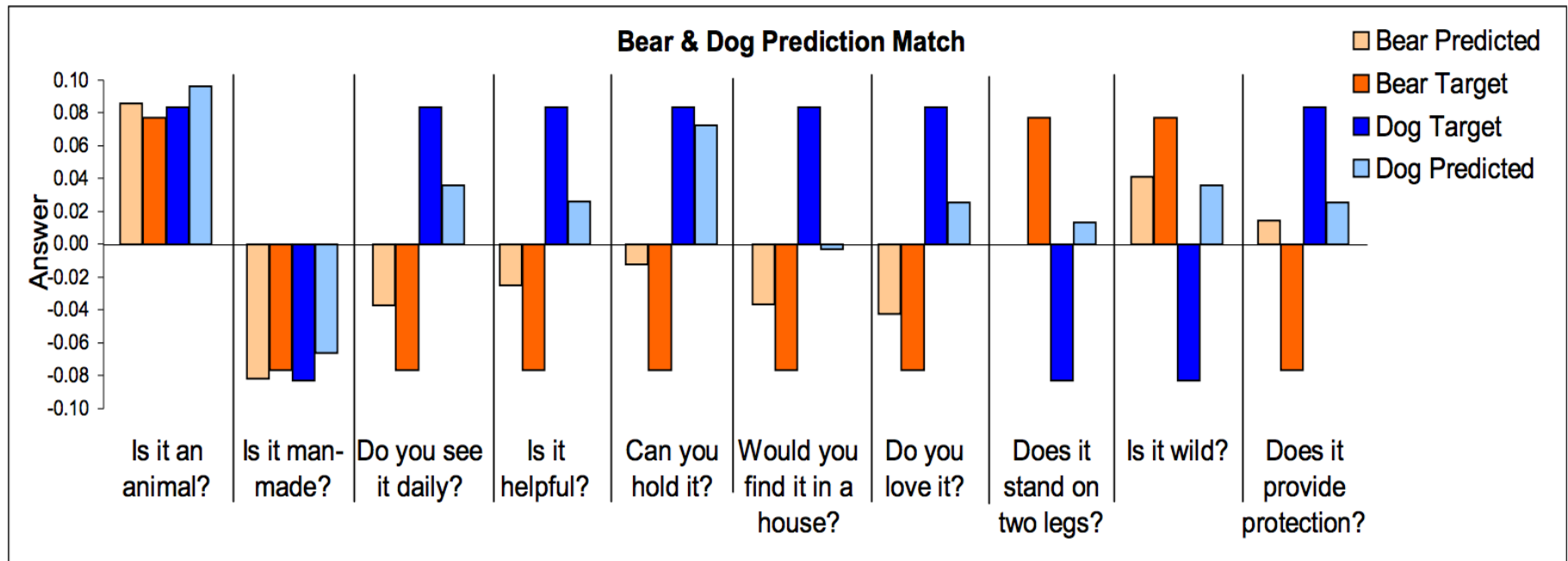


Figure 1: Ten semantic features from the human218 knowledge base for the words *bear* and *dog*. The true encoding is shown along with the predicted encoding when fMRI images for bear and dog were left out of the training set.

Can we decode the word from a large set of words?

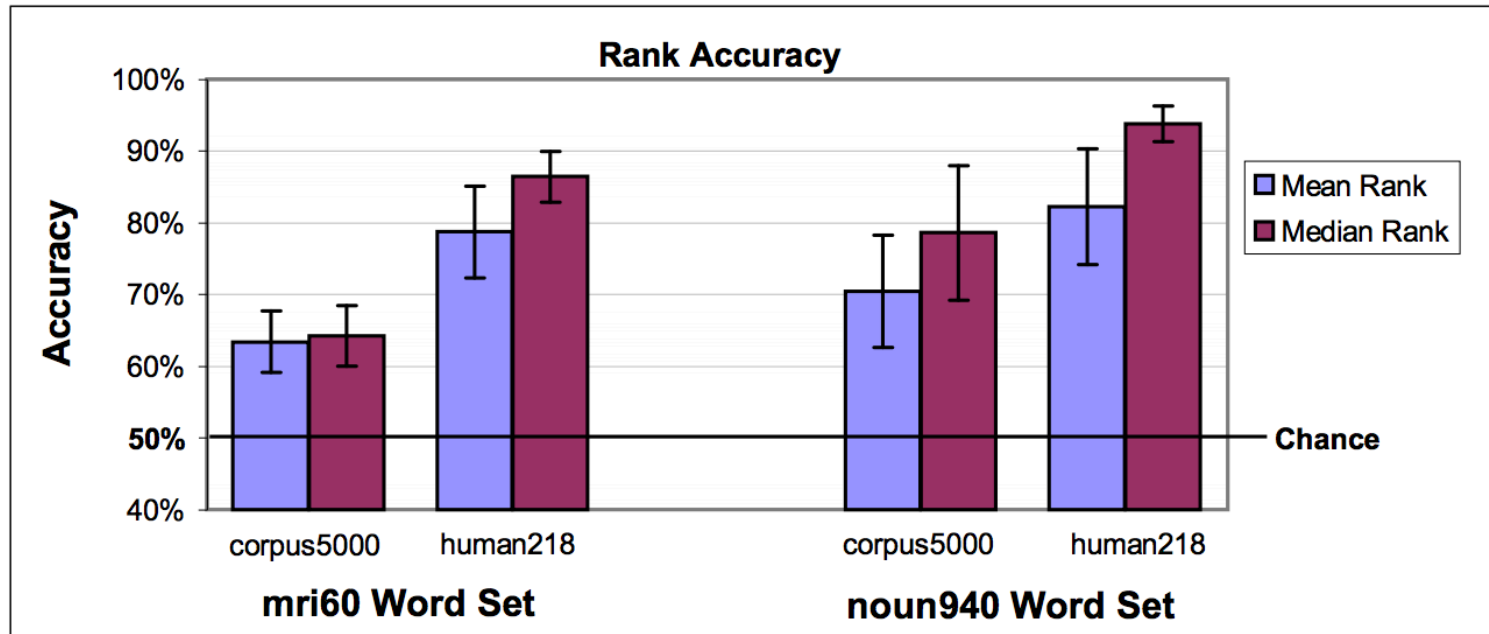


Figure 2: The mean and median rank accuracies across nine participants for two different semantic feature sets. Both the original 60 fMRI words and a set of 940 nouns were considered.

Table 2: The top five predicted words for a novel fMRI image taken for the word in bold (all fMRI images taken from participant P1). The number in the parentheses contains the rank of the correct word selected from 941 concrete nouns in English.

Bear	Foot	Screwdriver	Train	Truck	Celery	House	Pants
(1)	(1)	(1)	(1)	(2)	(5)	(6)	(21)
<i>bear</i>	<i>foot</i>	<i>screwdriver</i>	<i>train</i>	jeep	beet	supermarket	clothing
fox	feet	pin	jet	<i>truck</i>	artichoke	hotel	vest
wolf	ankle	nail	jail	minivan	grape	theater	t-shirt
yak	knee	wrench	factory	bus	cabbage	school	clothes
gorilla	face	dagger	bus	sedan	<i>celery</i>	factory	panties

END