# Comprendre les données visuelles à grande échelle

**ENSIMAG**
**2019-2020**

Karteek Alahari & Diane Larlus

https://project.inria.fr/bigvisdata/

# Recap

- 1st lecture
  - Big data and big visual data
  - Image annotation and datasets
  - Visual data analysis tasks
  - Image search

  - Local representations
    - Interest points (examples ?)
    - Local descriptors
    - Matching
    - Geometric verification
  - Reverse file index

# Today's lecture

- Supervised learning
- Global representations
- Hierarchical representations
- Learning features
- Compositionality of features
- Classification problem (with SVM)
- End-to-end learning

Crédits pour la majorité des transparents qui suivent: D. Batra, R. Fergus, D. Larlus, Y. LeCun, M. Renzato, HKUST

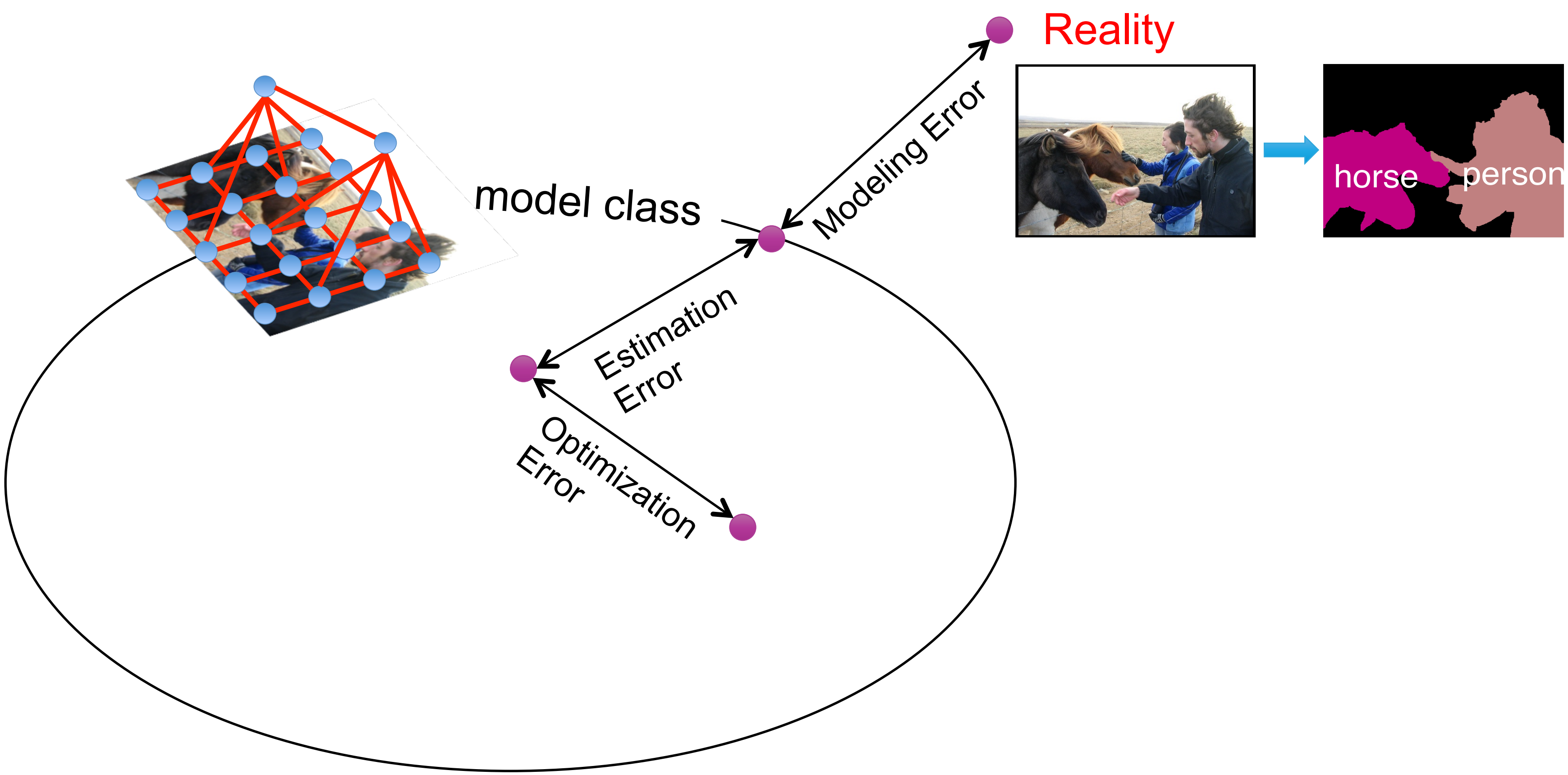
# Supervised Learning

- Input: x (images, text, emails…)
- Output: y (spam or non-spam…)
- (Unknown) Target Function
  - f: X → Y (the "true" mapping / reality)
- Data
  - $(x_1,y_1), (x_2,y_2), \ldots, (x_N,y_N)$
- Model / Hypothesis Class
  - g: X → Y
  - $y = g(x) = \text{sign}(w^T x)$
- Learning = Search in hypothesis space
  - Find best g in model class

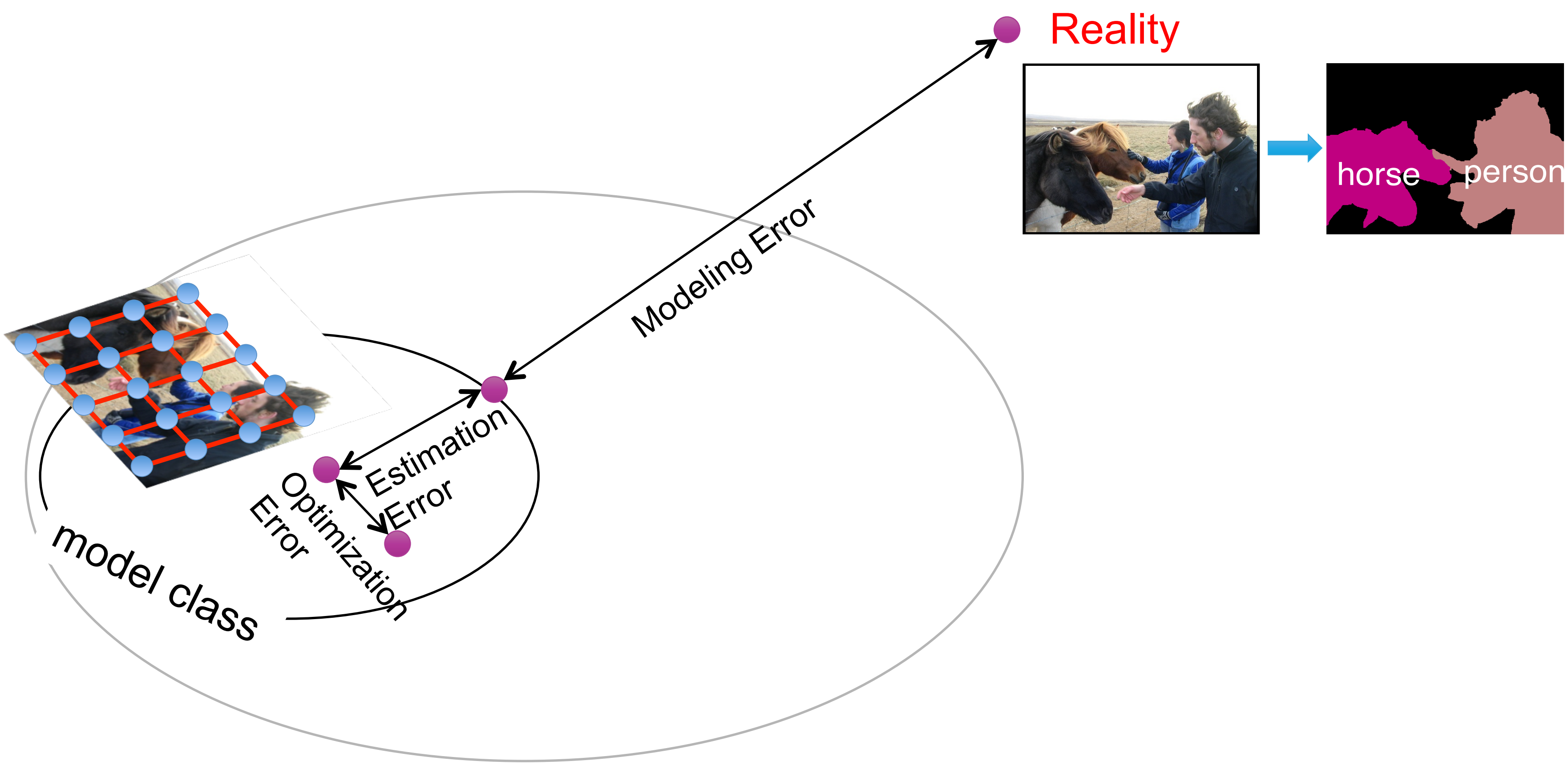
# Basic Steps of Supervised Learning

- **Set up** a supervised learning problem
- **Data collection**
  - Start with training data for which we know the correct outcome provided by a teacher or oracle
- **Representation**
  - Choose how to represent the data
- **Modeling**
  - Choose a hypothesis class: $H = \{g: X \rightarrow Y\}$
- **Learning/Estimation**
  - Find best hypothesis you can in the chosen class
- **Model Selection**
  - Try different models. Picks the best one. (More on this later)
- If happy stop
  - Else refine one or more of the above

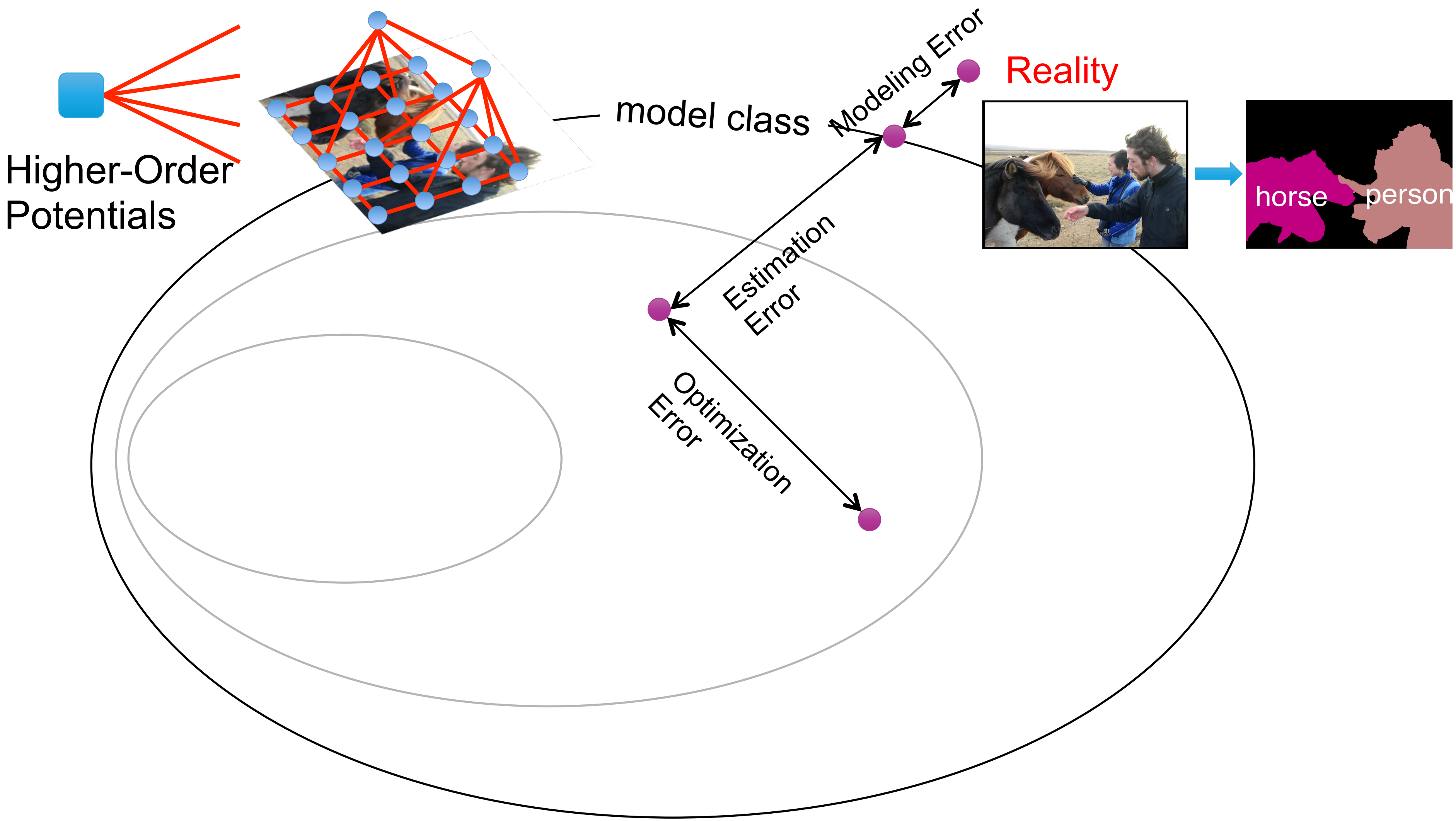
# Error Decomposition



# Error Decomposition



# Error Decomposition



# Description globale

- **Une description globale** est une représentation de l'image dans son ensemble, sous la forme d'un vecteur de taille fixe
- Caractéristiques
  - **un** vecteur de description par objet visuel
  - mesure de (dis-)similarité définie sur l'espace de ces descripteurs

# Exemple de description globale : histogramme de couleur

- Chaque pixel est décrit par un vecteur de couleur
  - Par exemple un vecteur RGB $\in R^3$, mais le plus souvent on utilise un autre espace de couleur plus approprié
- L'ensemble des vecteurs de couleurs forme une distribution
  - Description de la distribution avec un histogramme
  - Nécessite la discrétisation de l'espace et la normalisation de l'histogramme
- Comparaison de deux histogrammes par une mesure de dissimilarité, par exemple la « distance » du Khi-2

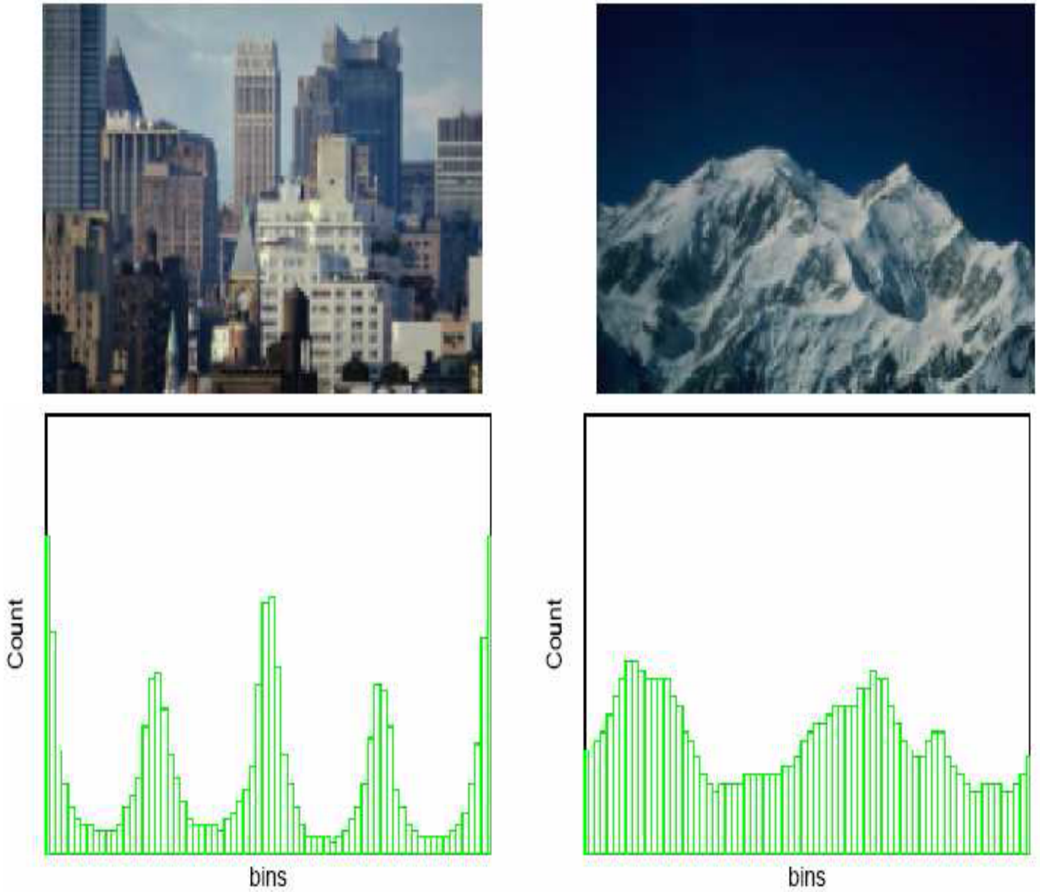
Color indexing, Swain & Ballard, IJCV 1991

The earth mover's distance, multi-dimensional scaling, and color-based image retrieval
Y Rubner, LJ Guibas, C Tomasi - Proceedings of DARPA Image, 1997

# Visualisation des distances pour une représentation basée sur la couleur

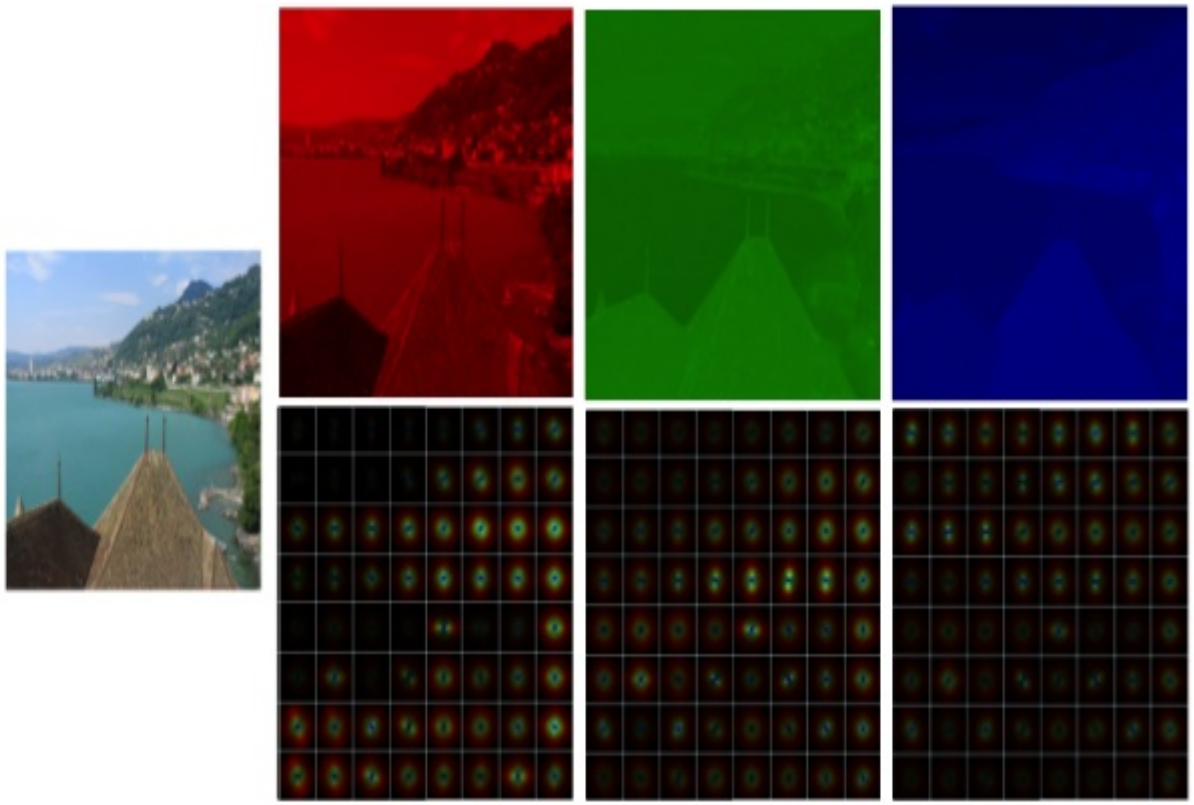
# Exemple de description globale : contours

- Exemple de descripteur de texture
- Descripteur = histogramme d'orientation des contours



# Apparence globale de l'image : descripteur GIST

- Similaire au descripteur SIFT sur une pyramide d'images, avec image = patch.
- Peut p canaux R, G, B s



- [Modeling the shape of the scene: a holistic representation of the spatial enveloppe, Aude Oliva, Antonio Torralba, IJCV 2001]

# Agrégation de descripteurs locaux

- Définir un descripteur global à partir de l'ensemble des descripteurs locaux d'une image
  - Compact, et plus facile à manipuler que les descripteurs locaux de départ
- Contraintes
  - Des images similaires doivent avoir des représentations similaires
  - Des images dissimilaires doivent avoir des représentations dissimilaires
- Nécessité d’un compromis entre ces propriétés
  - Robustes aux transformations (échelle, occultation, éclairage, etc.)
  - Informatifs (bonne description du contenu)
  - Efficaces à calculer, à stocker, à manipuler

# Agrégation de descripteurs locaux

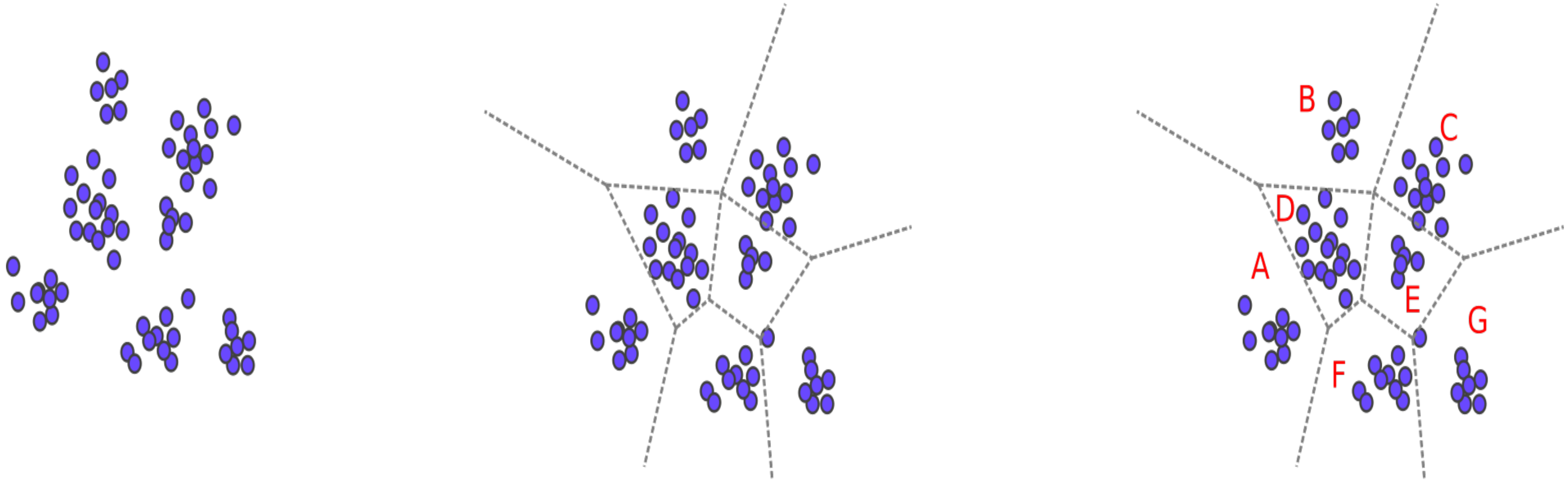
- La base: « *bag-of-features* », « *bag-of-patches* »
  - *bag* = on perd l'ordre, la géométrie.
    On utilise des ensembles non-ordonnés de descripteurs locaux
- Quantification: représentation par sac-de-mots,
  ou *bag-of-words* (BoW, aussi appelée *bag-of-visual-words* ou BoV)
  - On suppose une transformation : descripteur local -> index entier
    (par un algorithme de quantification vectorielle = *clustering*)
  - Cette transformation peut être vue comme la création d'un vocabulaire visuel
  - Histogramme de ces entiers = descripteur global

# Agrégation de descripteurs locaux

Utilisation d’un vocabulaire visuel

Etapes:

- Discrétisation de l’espace des descripteurs, par exemple avec un algorithme de *clustering*
- Chaque descripteur est associé à un (ou plusieurs) mot visuel

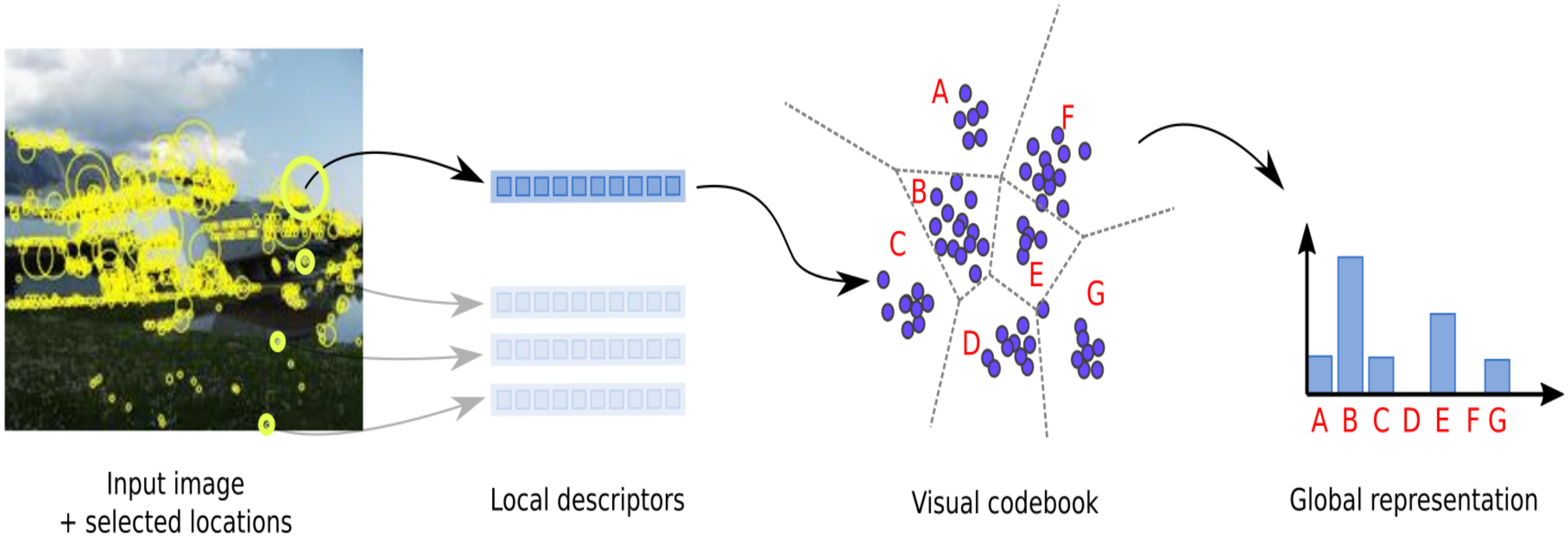


# From quantization to bag-of-visual-features

## Principle

[Sivic & Zisserman. ICCV 2003]
[Csurka et al. ECCV SLCV 2004]

- Extract local descriptors
- Convert local descriptors into visual words, using a visual codebook
- Represent images as a histogram of occurrences



# Lien avec le cours précédent

La semaine dernière

- Construction d'un vocabulaire visuel pour la construction d'un fichier inversé pour une recherche rapide

Cette semaine

- Construction d'un vocabulaire visuel afin d'agréger les descripteurs locaux en une représentation globale
  - Les descripteurs locaux n'ont pas besoin d'être gardés en mémoire
  - Seule la représentation globale est conservée
  - Extrêmement compact, mais
  - pas de vérification géométrique possible,
  - perte totale des informatiques géométriques,
  - quantification = perte d'information -> représentation « grossière » (*coarse*)

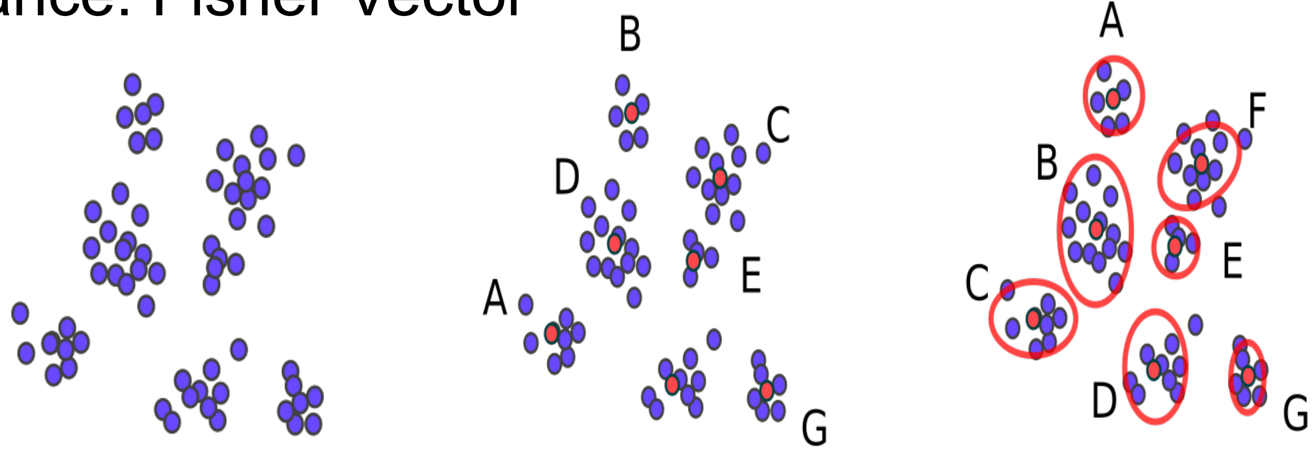
# How can we refine this description?

## Relatively coarse representation

- Solution 1: more entry in the codebook
  - drawback: significant computational cost

[Li et al. ICCV 2009]
[Yang et al. ECCV 2010]

- Solution 2: beyond counting, adding higher order statistics
  - Mean: VLAD
  - Variance: Fisher Vector

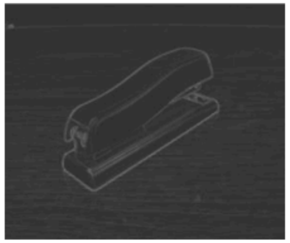


# Traditional Approaches for Recognition

Features are not learned

Input data (pixels) → feature representation (hand-crafted) → Learning Algorithm (e.g., SVM)

Image

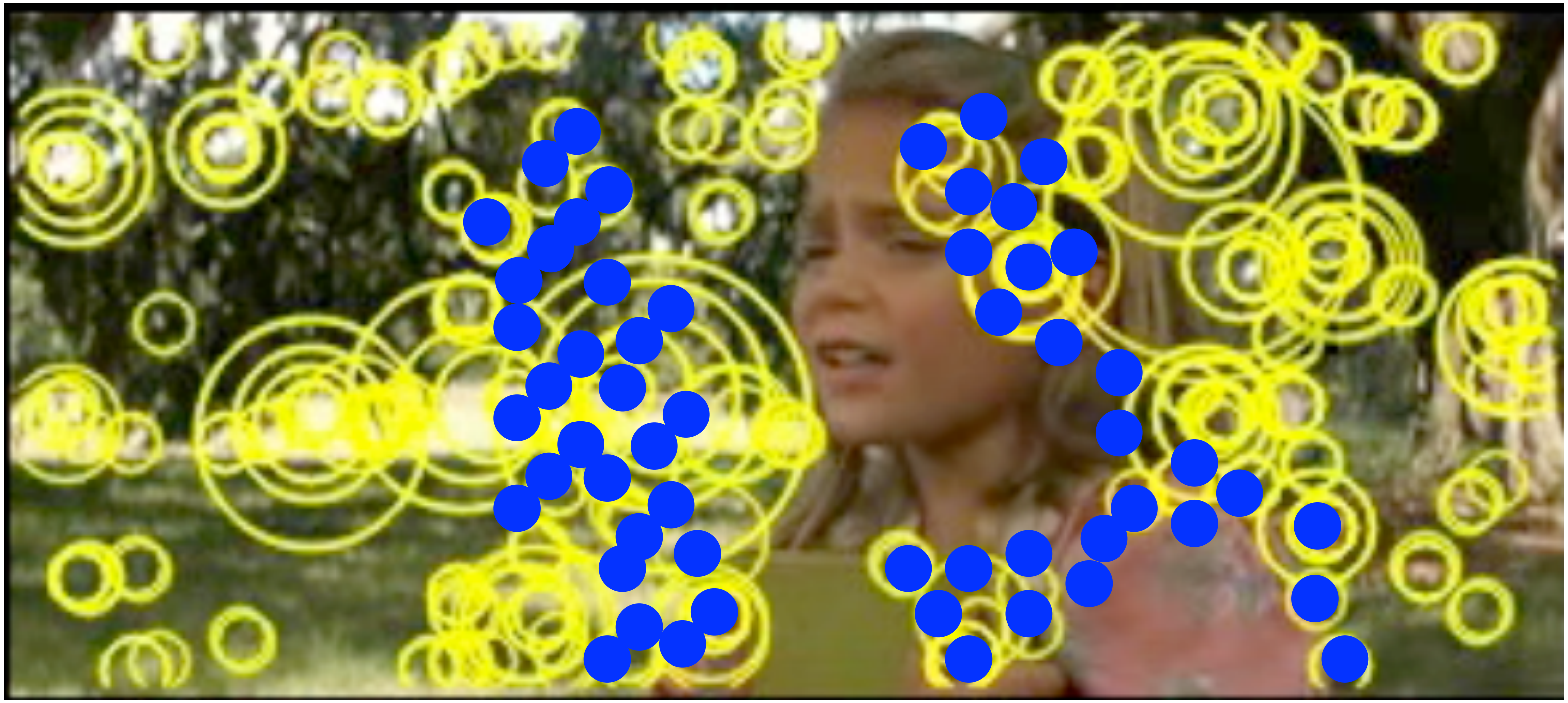


Low-level
vision features
(edges, SIFT, HOG, etc.)

Object detection
/ classification

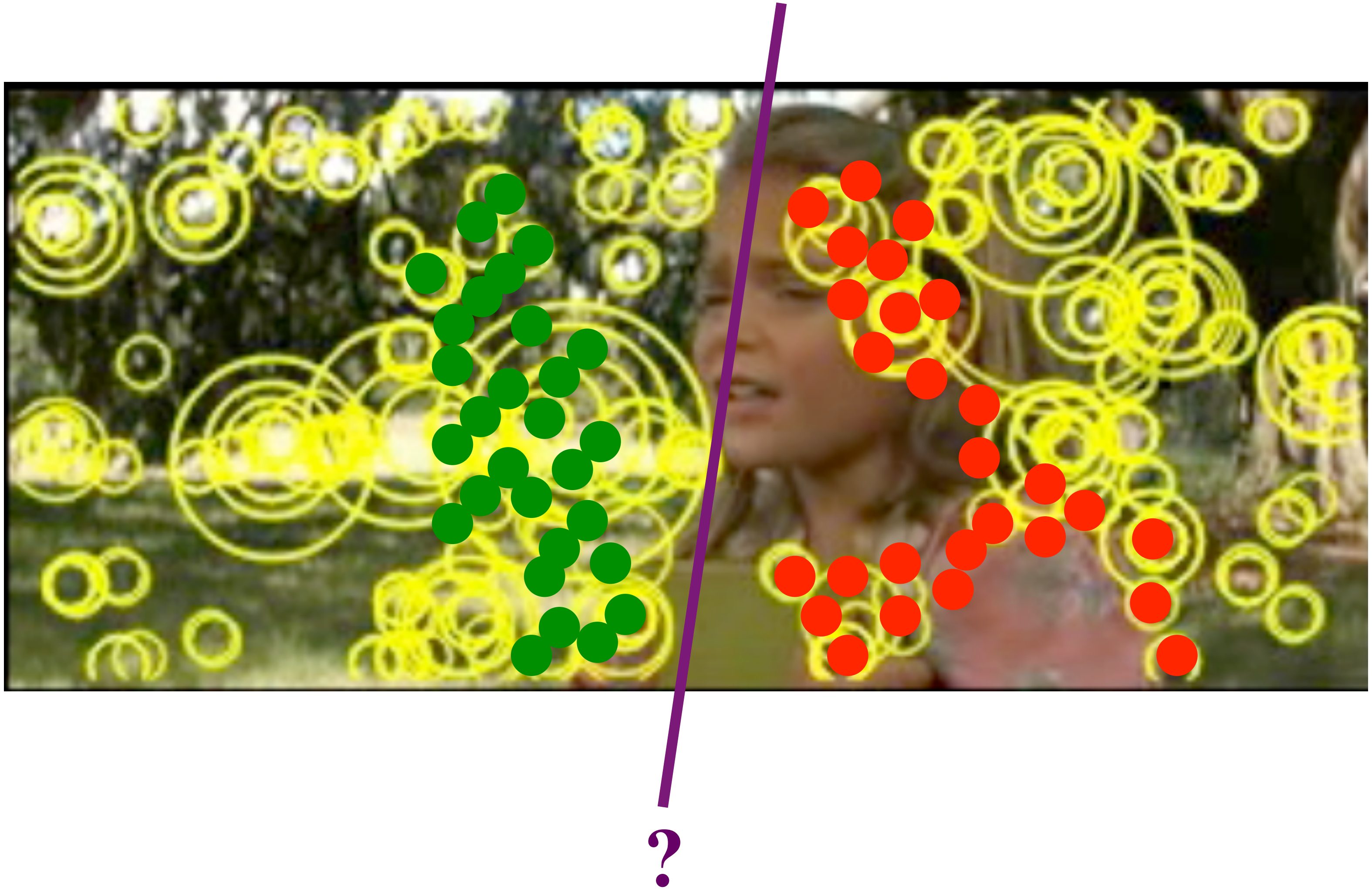
# Classification

- What can we do, given all these features ?

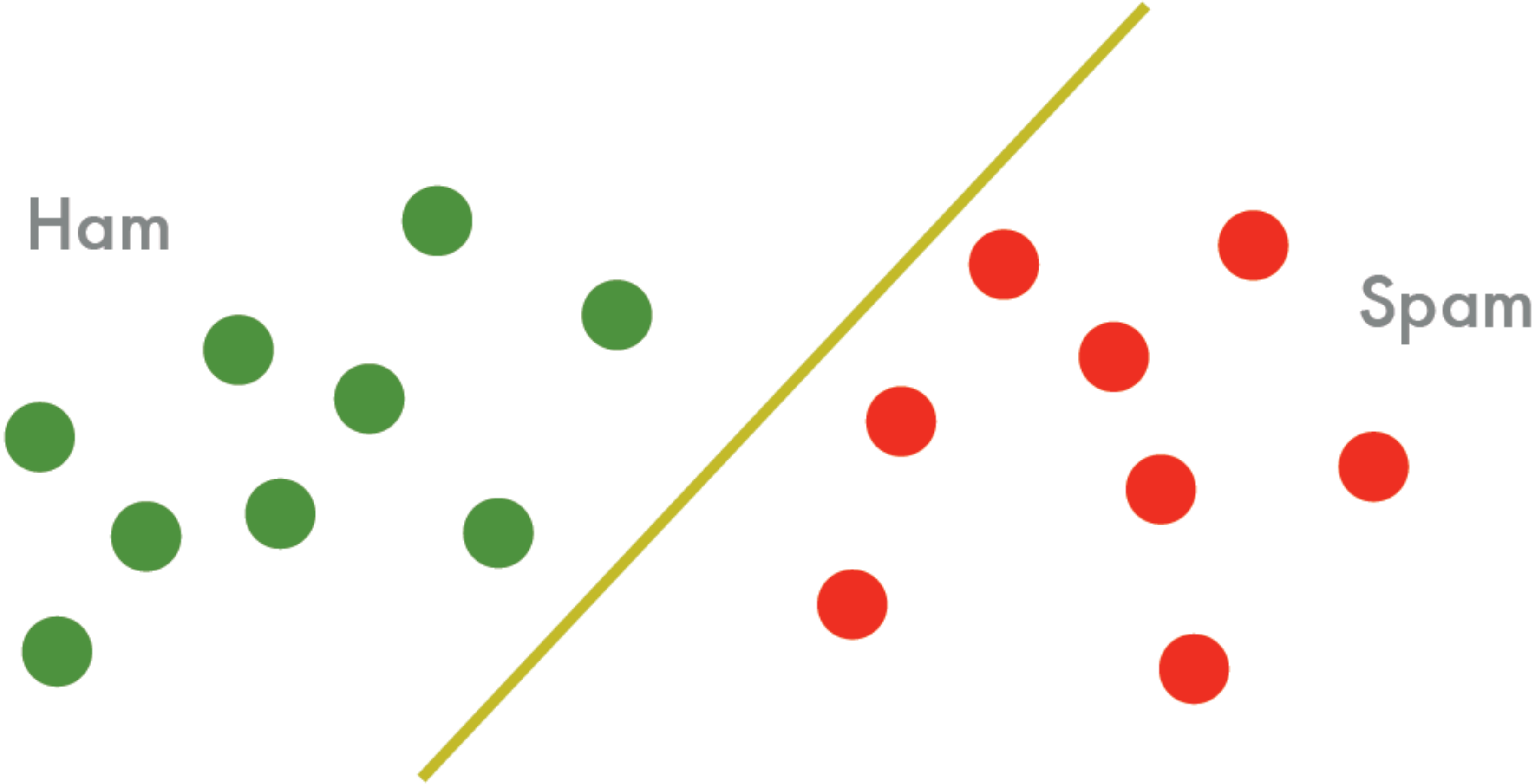


# Classification

- What can we do, given all these features ?

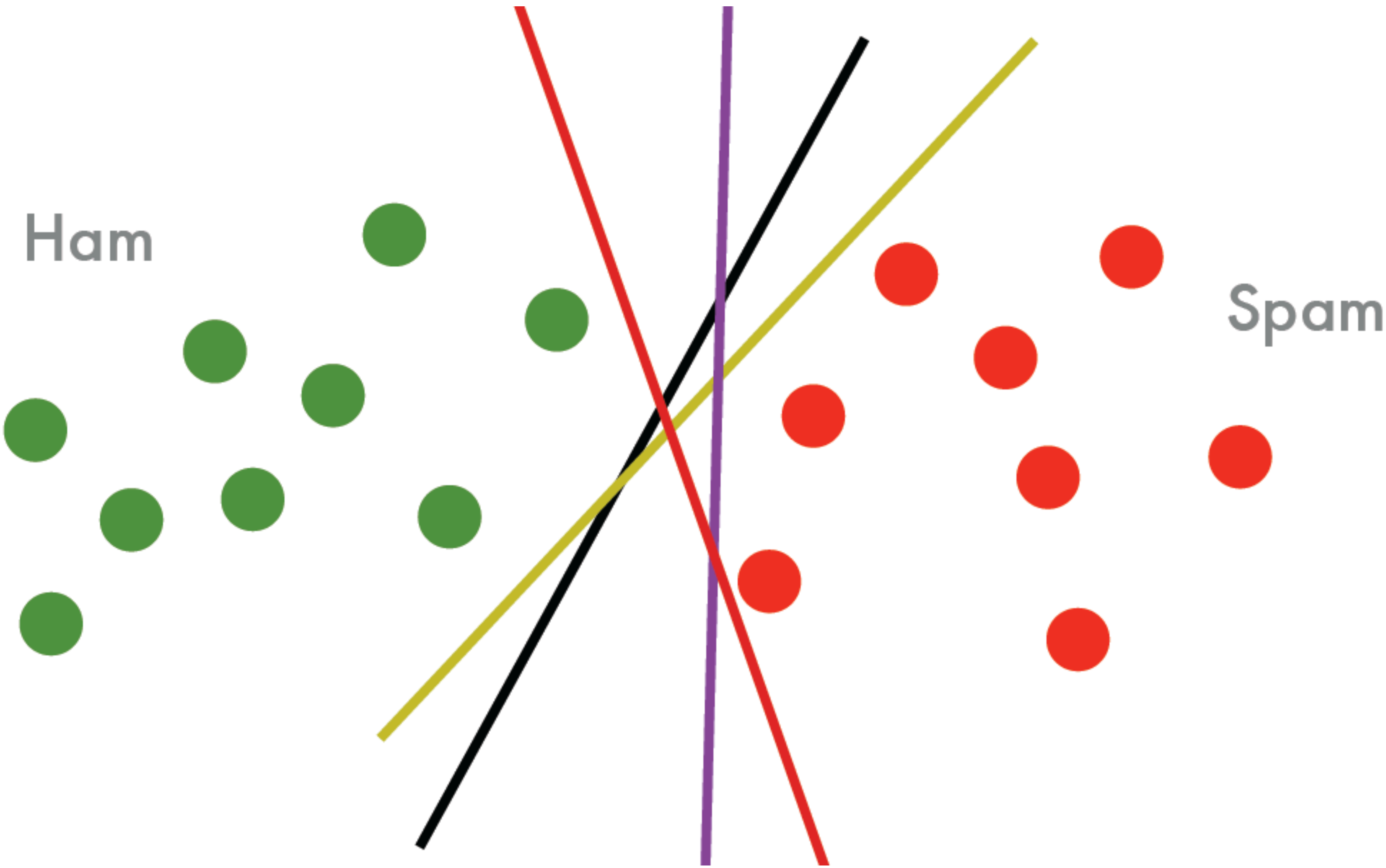


# Classification



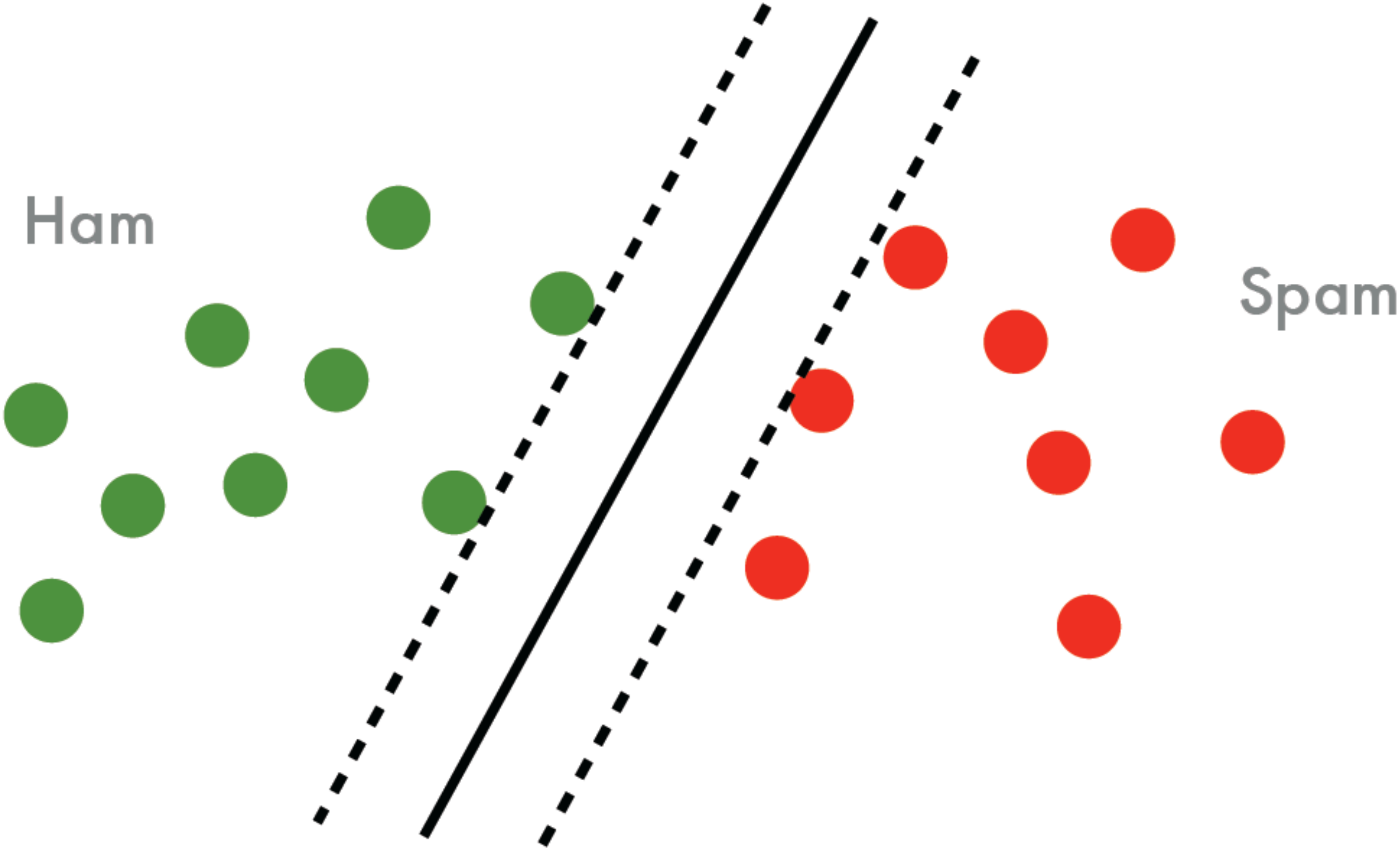

Slides courtesy Alex Smola

# Classification




Slides courtesy Alex Smola

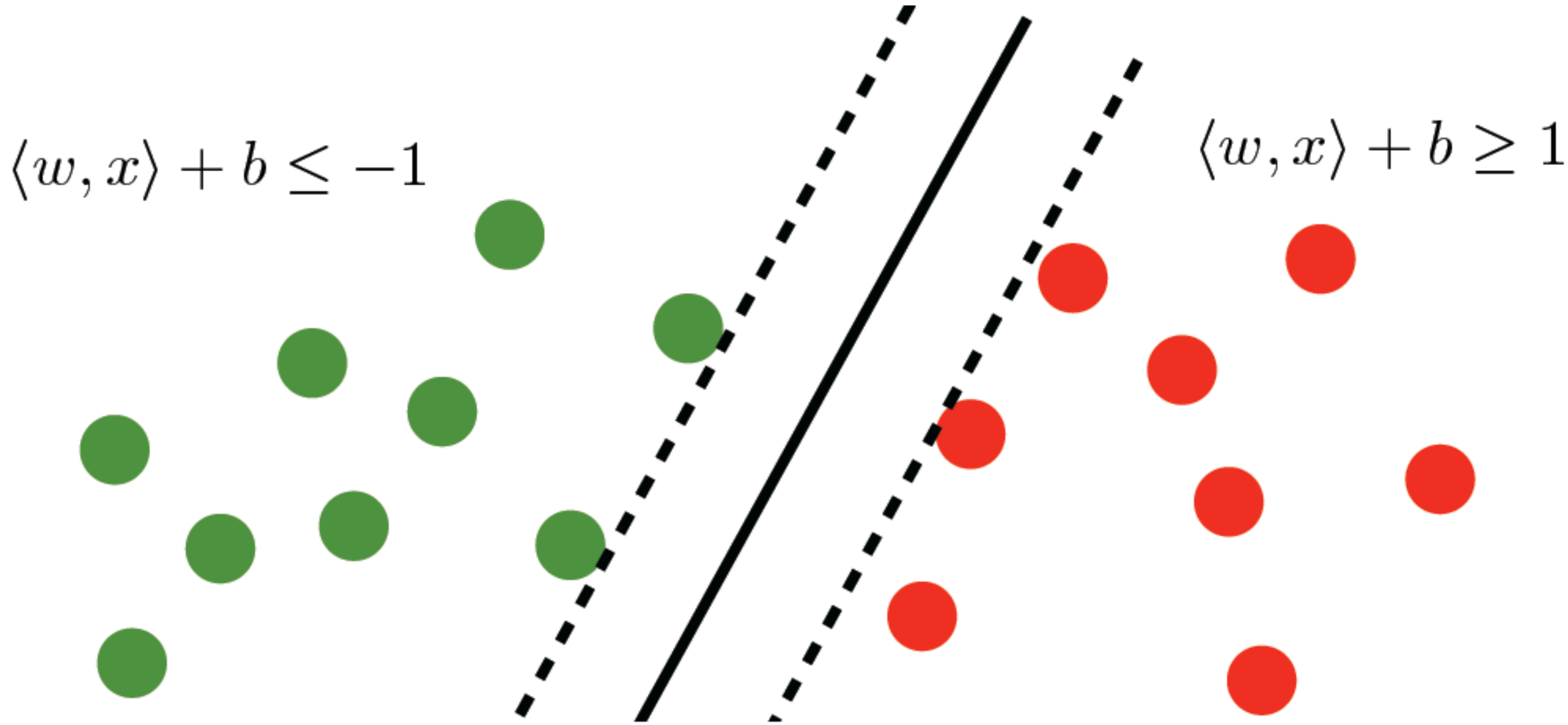
# Classification




**Large margin classifier**

Slides courtesy Alex Smola

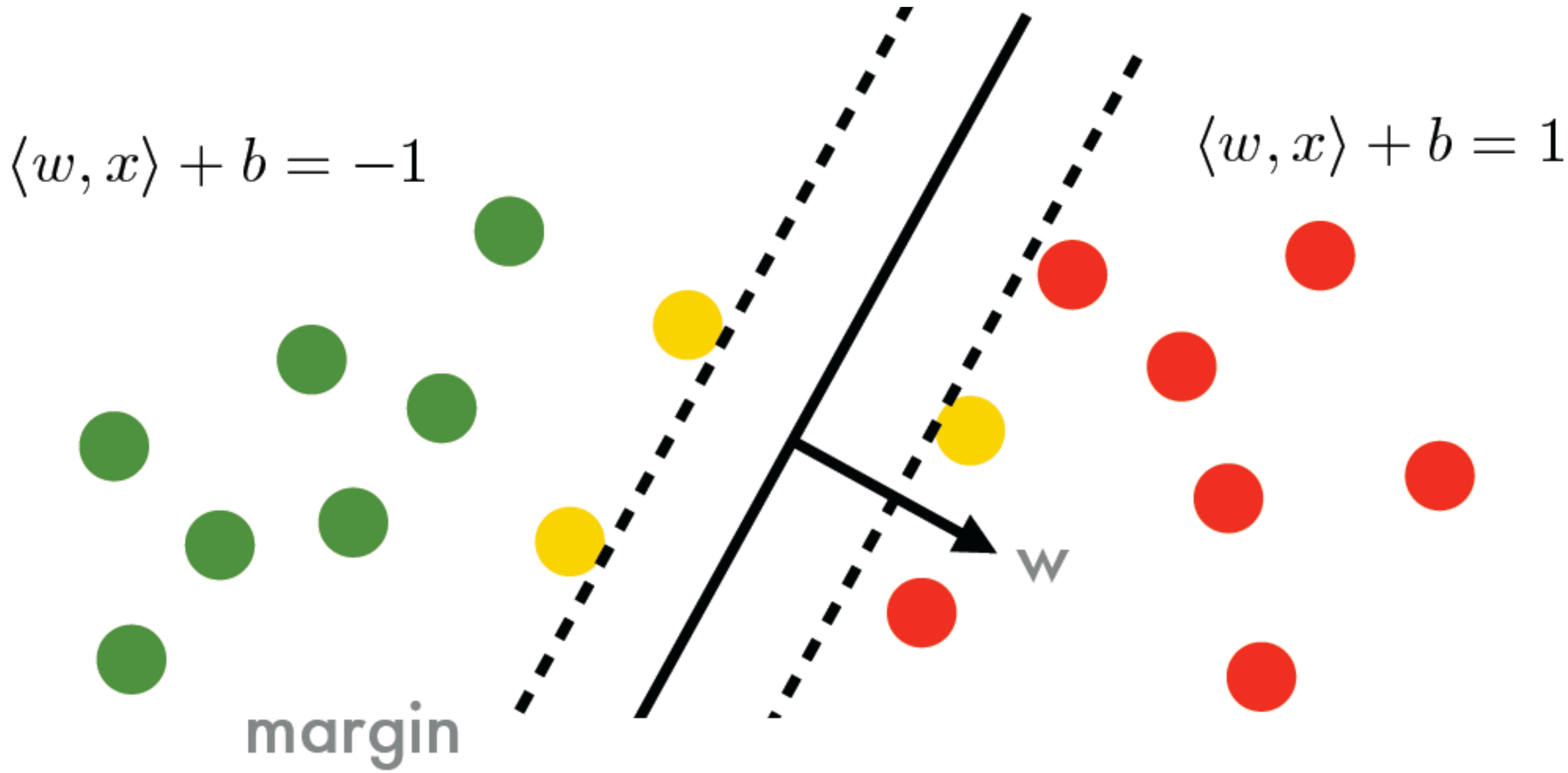
# Classification




linear function

$$f(x) = \langle w, x \rangle + b$$

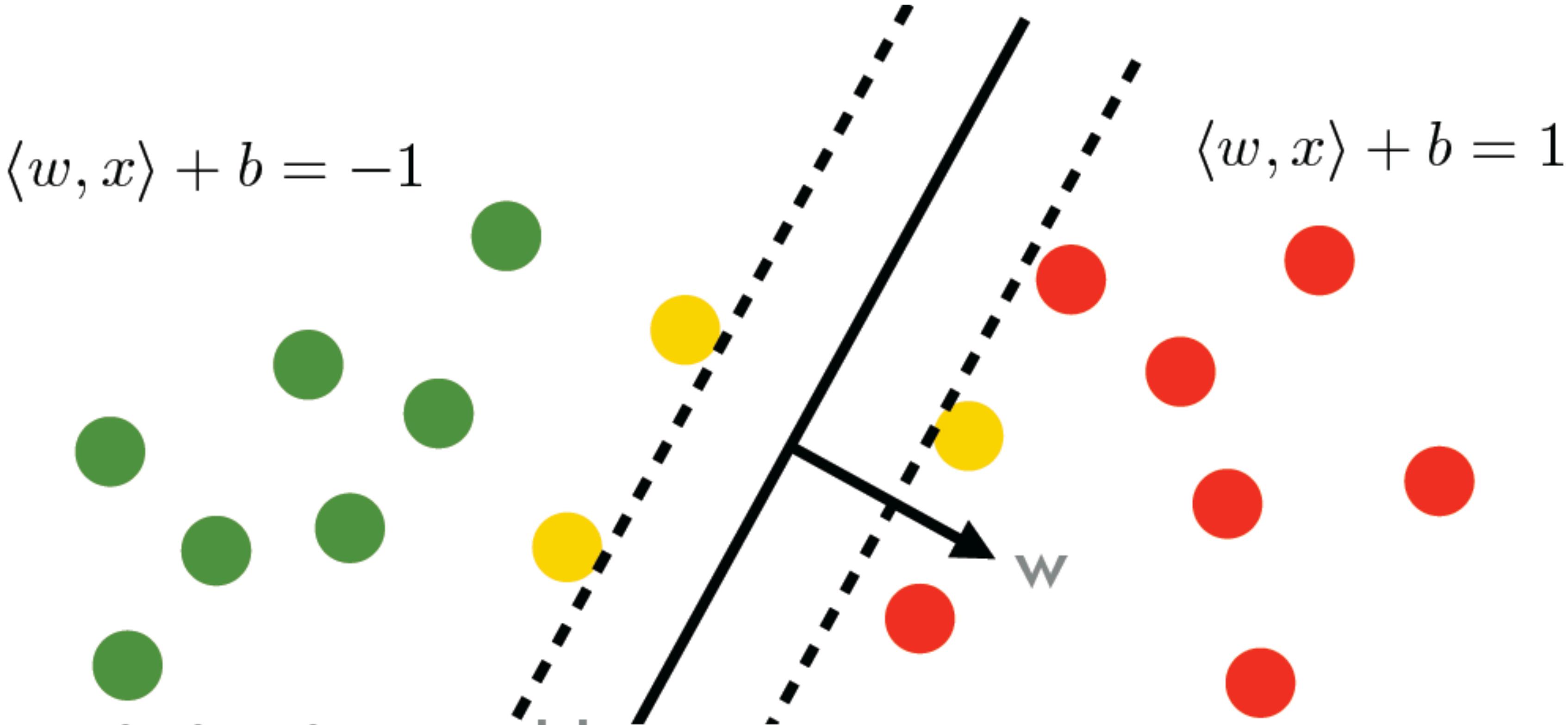
# Classification




$$\frac{\langle x_+ - x_-, w \rangle}{2\,\|w\|} = \frac{1}{2\,\|w\|}\left[\left[\langle x_+, w \rangle + b\right] - \left[\langle x_-, w \rangle + b\right]\right] = \frac{1}{\|w\|}$$

Slides courtesy Alex Smola

# Classification




optimization problem

$$\underset{w,b}{\text{maximize}} \frac{1}{\|w\|} \text{ subject to } y_i \left[ \langle x_i, w \rangle + b \right] \geq 1$$

# Support Vector Machines

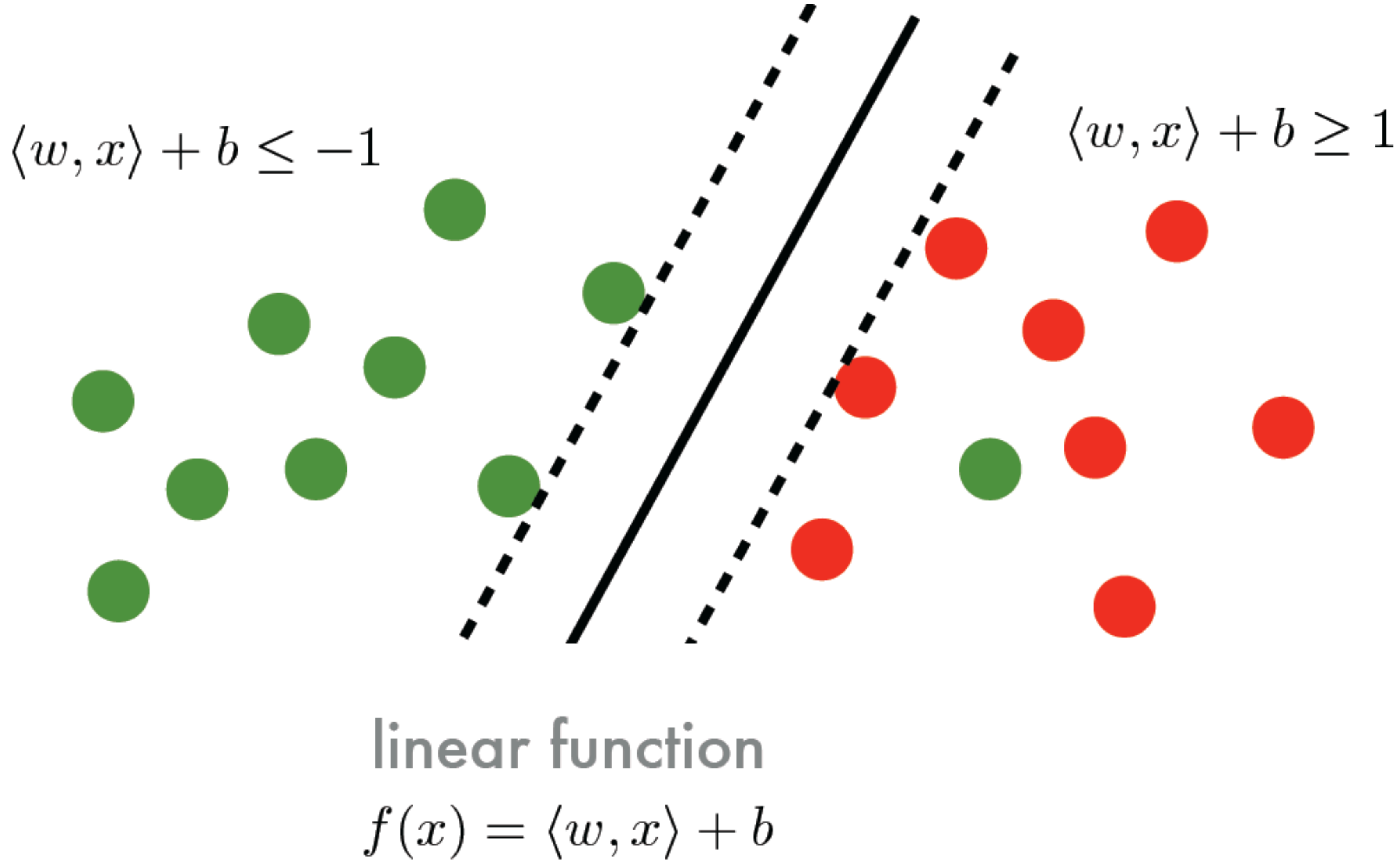
$$\underset{w,b}{\text{minimize}} \frac{1}{2} \|w\|^2 \text{ subject to } y_i \left[\langle x_i, w \rangle + b\right] \geq 1$$

- Many optimization techinques to solve this
  - e.g., Stochastic Gradient Descent (SGD)

- Implementations available
  - SVM$^{\text{light}}$ (Thorsten Joachims)
  - SGD-SVM (Léon Bottou)

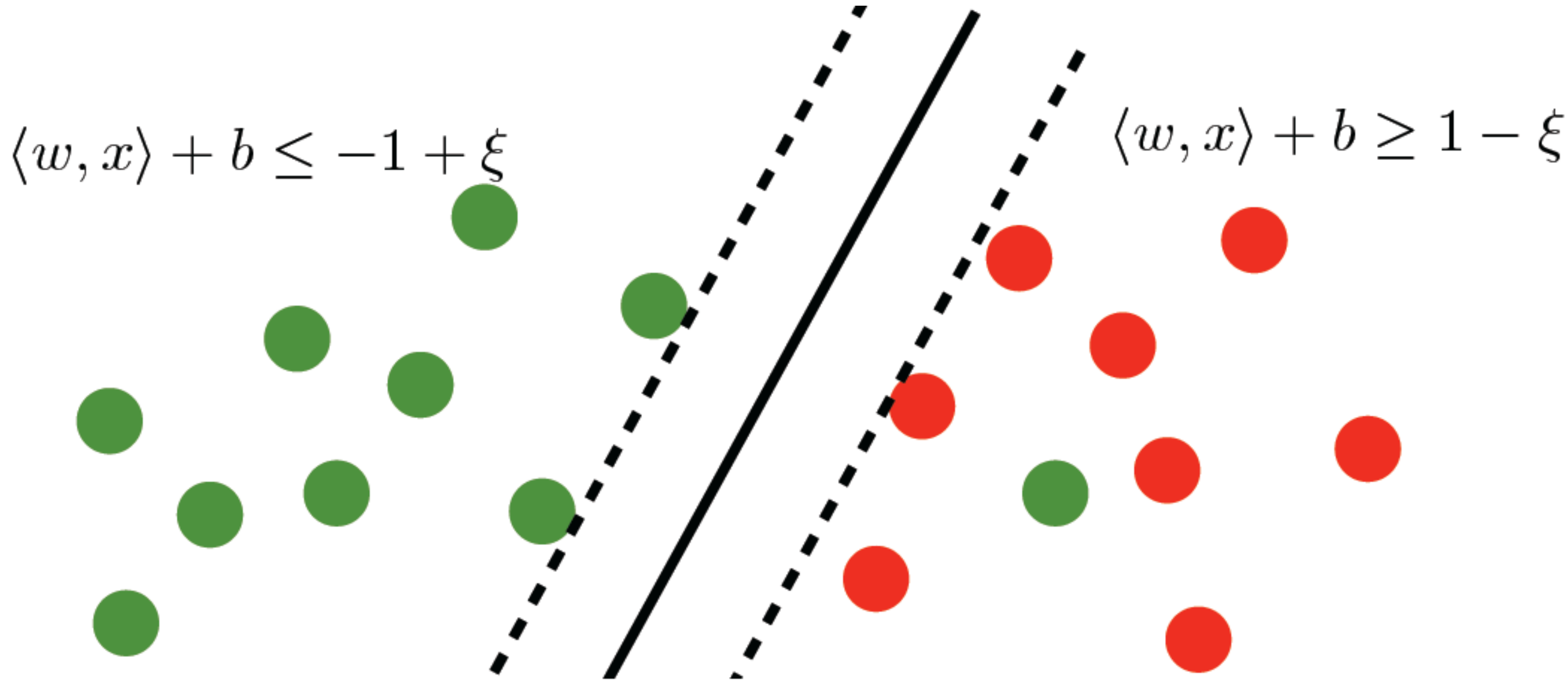
# Support Vector Machines

- What about linearly inseparable cases ?



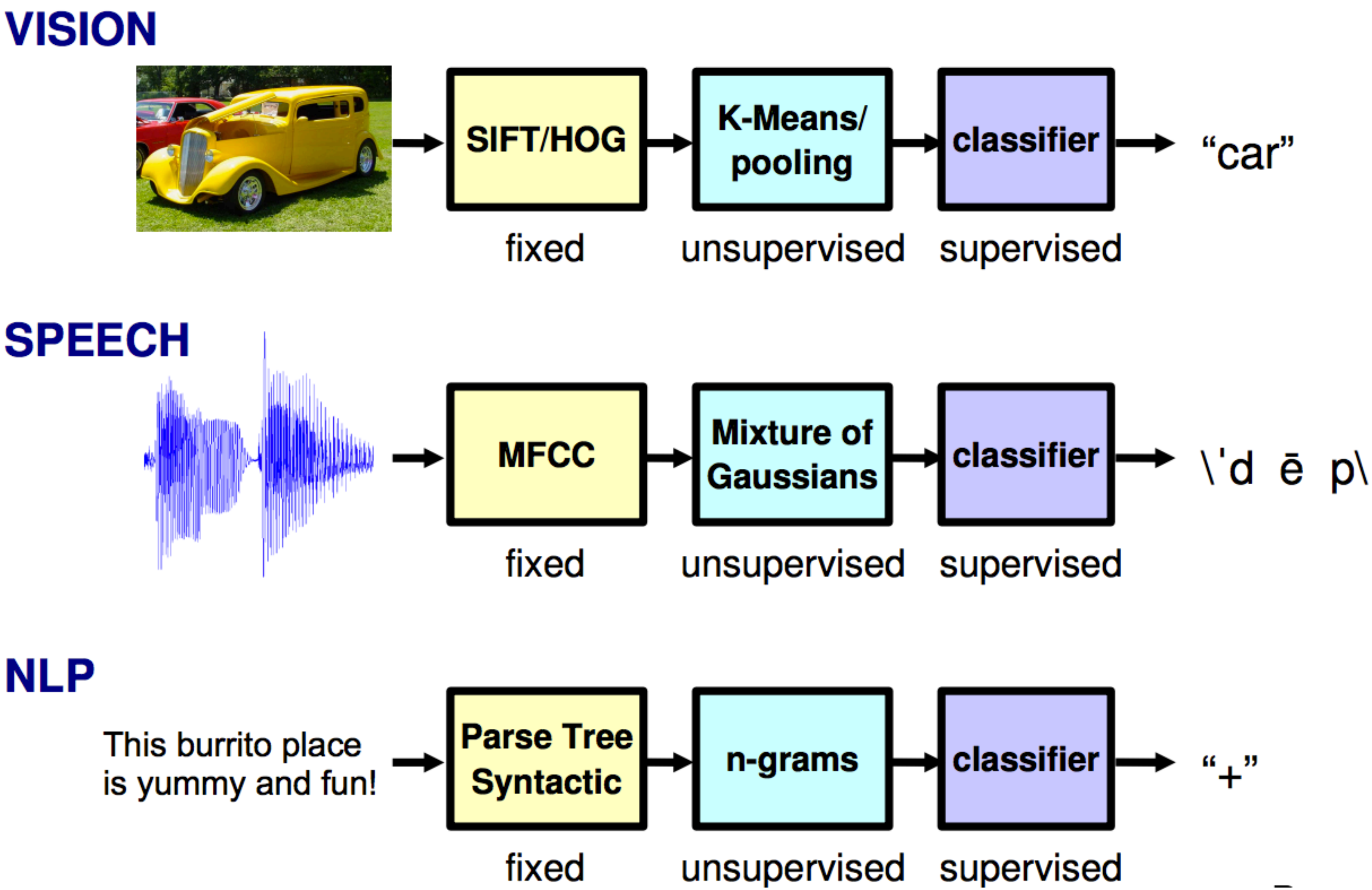
# Support Vector Machines

- What about linearly inseparable cases ?

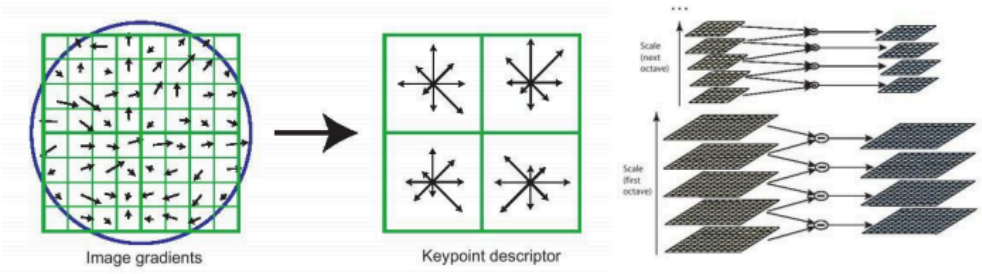



Slides courtesy Alex Smola

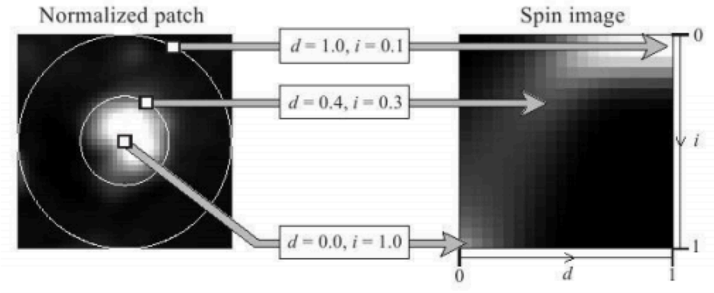
# Traditional Approaches for Recognition



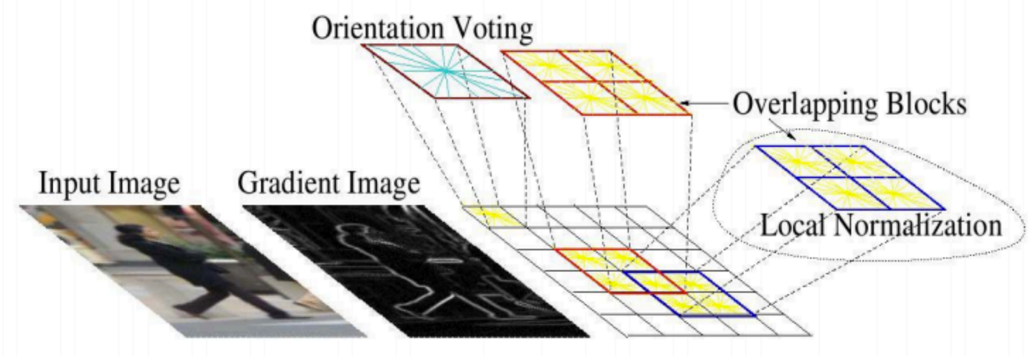
# Computer Vision Features



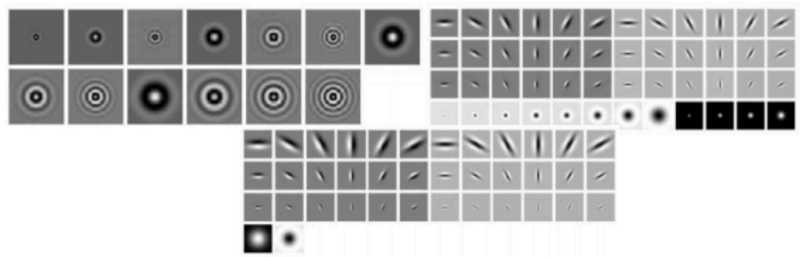

SIFT




Spin image




HoG



Textons

and many others:

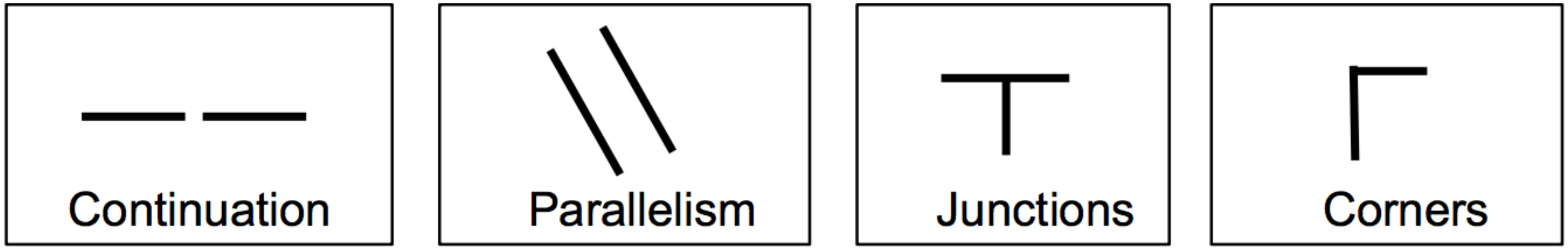
SURF, MSER, LBP, Color-SIFT, Color histogram, GLOH, …..

# Computer Vision Features

- Features are key to progress
- Have led to impressive results in various competitions (e.g., PASCAL VOC)
- Where do we go from here? Better features? Better classifiers?

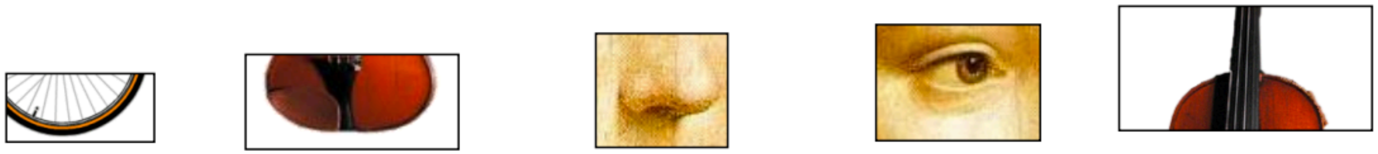
# Mid-level Representations

- Mid-level cues




"Tokens" from Vision by D.Marr: 

- Object parts:



# Mid-level Representations

**VISION**

pixels → edge → texton → motif → part → object

**SPEECH**

sample → spectral band → formant → motif → phone → word

**NLP**

character → word → NP/VP/.. → clause → sentence → story

Difficult to hand-engineer → What about learning them?

# Recall: Basic Steps of Supervised Learning

- **Set up** a supervised learning problem

- **Data collection**
  - Start with training data for which we know the correct outcome provided by a teacher or oracle

- **Representation**
  - Choose how to represent the data

- **Modeling**
  - Choose a hypothesis class: H = {g: X → Y}

- **Learning/Estimation**
  - Find best hypothesis you can in the chosen class

- **Model Selection**
  - Try different models. Picks the best one. (More on this later)

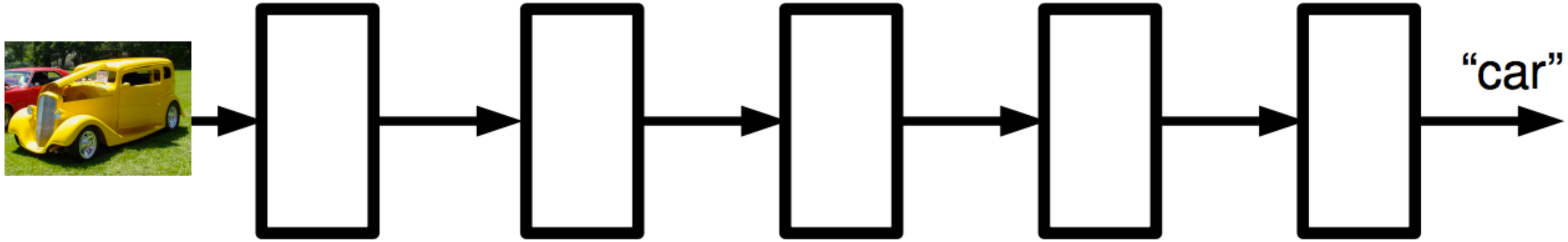
- If happy stop
  - Else refine one or more of the above

# Learning Feature Hierarchy

- Learn hierarchy
- All the way from pixels → classifier
- One layer extracts features from output of previous layer


- Train all layers jointly

# Deep Learning




**What is Deep Learning**

- Cascade of non-linear transformations
- End to end learning
- General framework (any hierarchical model is deep)

# So what is Deep (Machine) Learning?

- A few different ideas:

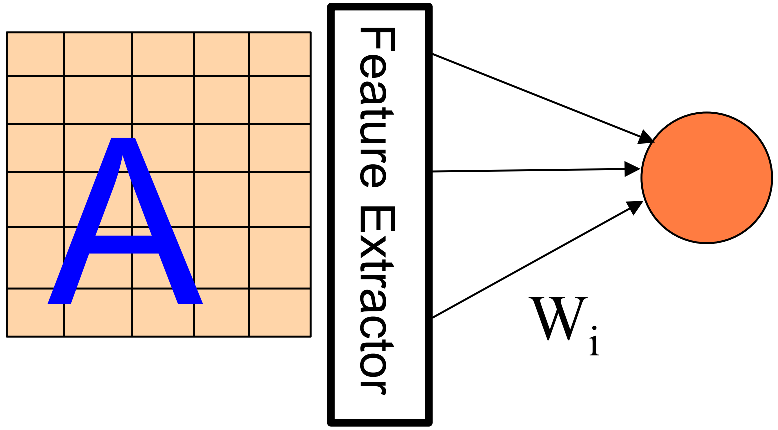
- (Hierarchical) Compositionality
  - Cascade of non-linear transformations
  - Multiple layers of representations

- End-to-End Learning
  - Learning (goal-driven) representations
  - Learning to feature extraction

- Distributed Representations
  - No single neuron "encodes" everything
  - Groups of neurons work together

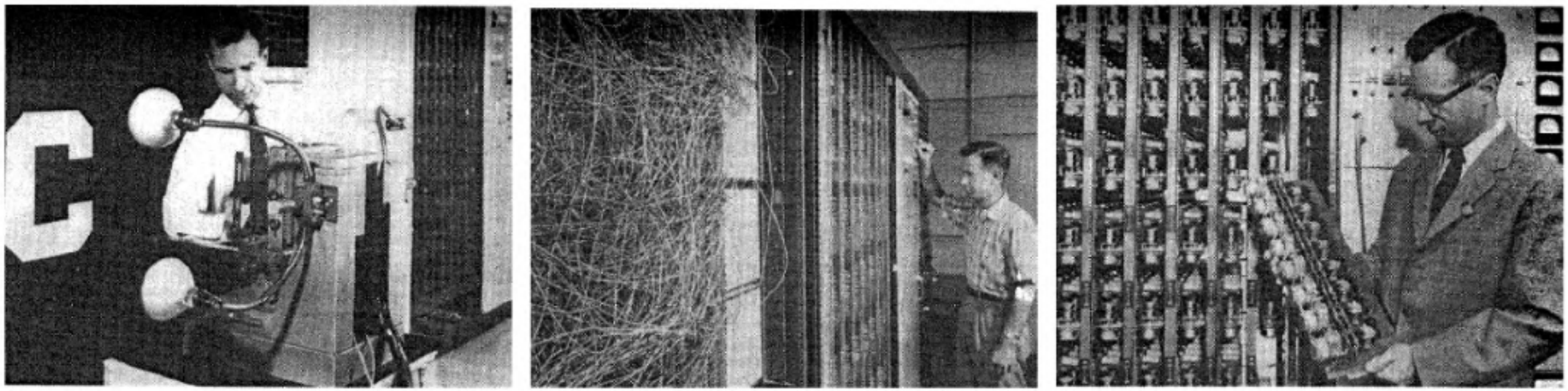
# It’s an old paradigm

- The first learning machine: the Perceptron
  - Built at Cornell in 1960
- The Perceptron was a linear classifier on top of a simple feature extractor




$$y = sign\left(\sum_{i=1}^{N} W_i F_i(X) + b\right)$$

- The vast majority of practical applications of ML today use glorified linear classifiers or glorified template matching.
- Designing a feature extractor requires considerable efforts by experts.



# It's only linear? What about hierarchy?

**VISION**

pixels → edge → texton → motif → part → object

**SPEECH**

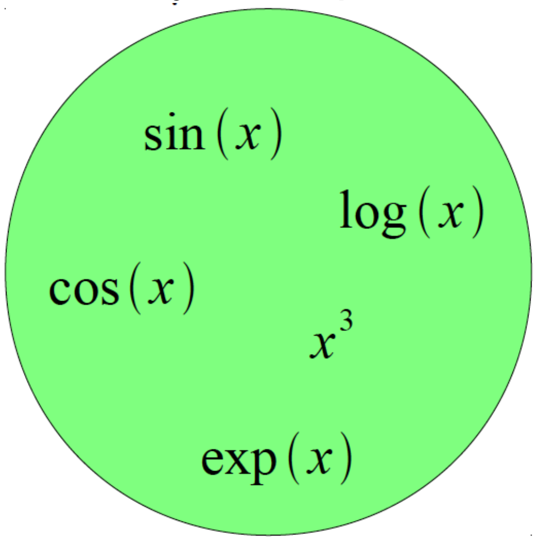
sample → spectral band → formant → motif → phone → word

**NLP**

character → word → NP/VP/.. → clause → sentence → story

# Building a Complicated Function

Given a library of simple functions



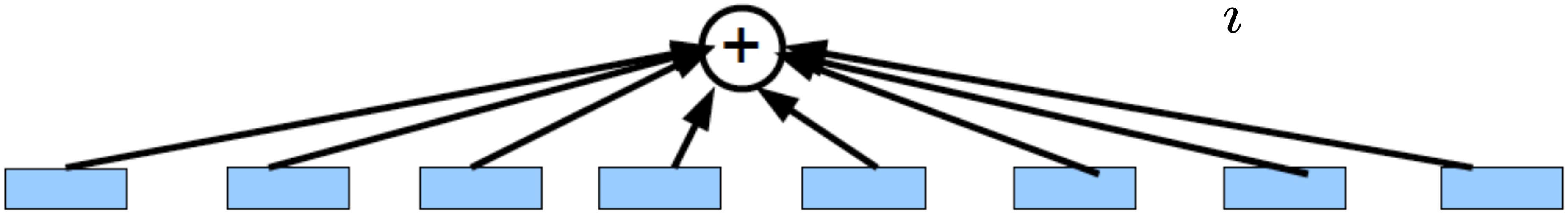

Compose into a

complicated function

Idea 1: Linear Combinations

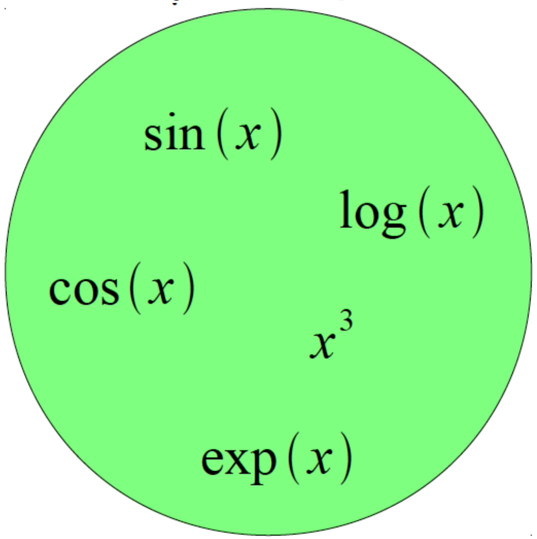
- Boosting
- Kernels
- …

$$f(x) = \sum_i \alpha_i g_i(x)$$



# Building a Complicated Function

Given a library of simple functions

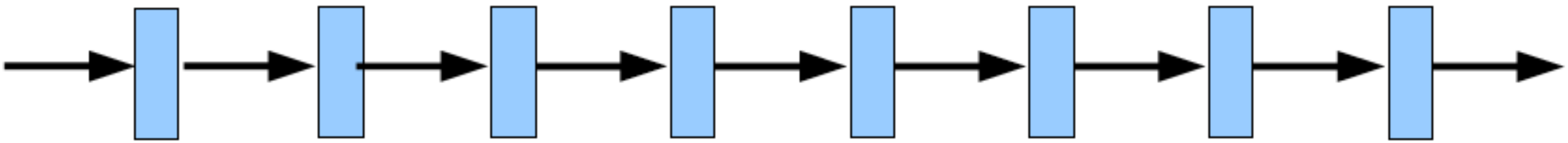



Compose into a complicated function

Idea 2: Compositions

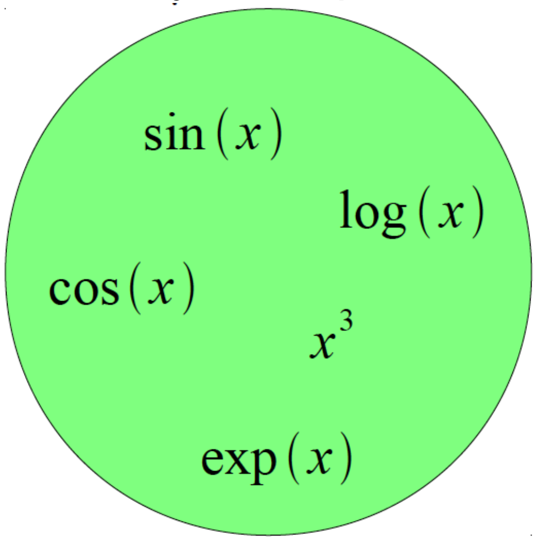
- Deep Learning
- Grammar models
- Scattering transforms…

$$f(x) = g_1(g_2(\ldots(g_n(x)\ldots))$$



# Building a Complicated Function

Given a library of simple functions

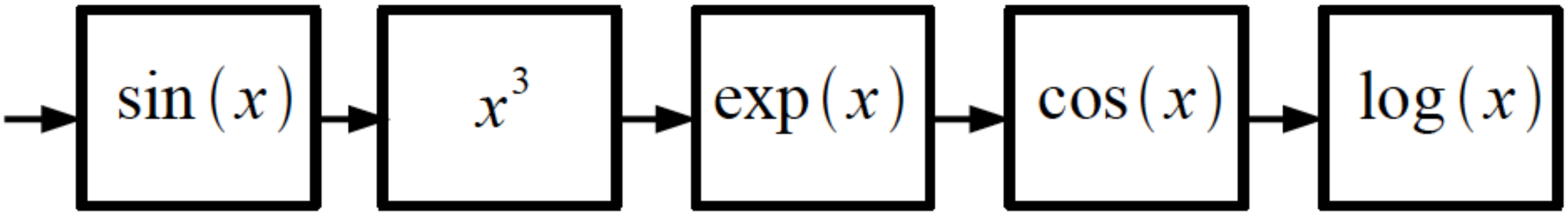



Compose into a complicated function

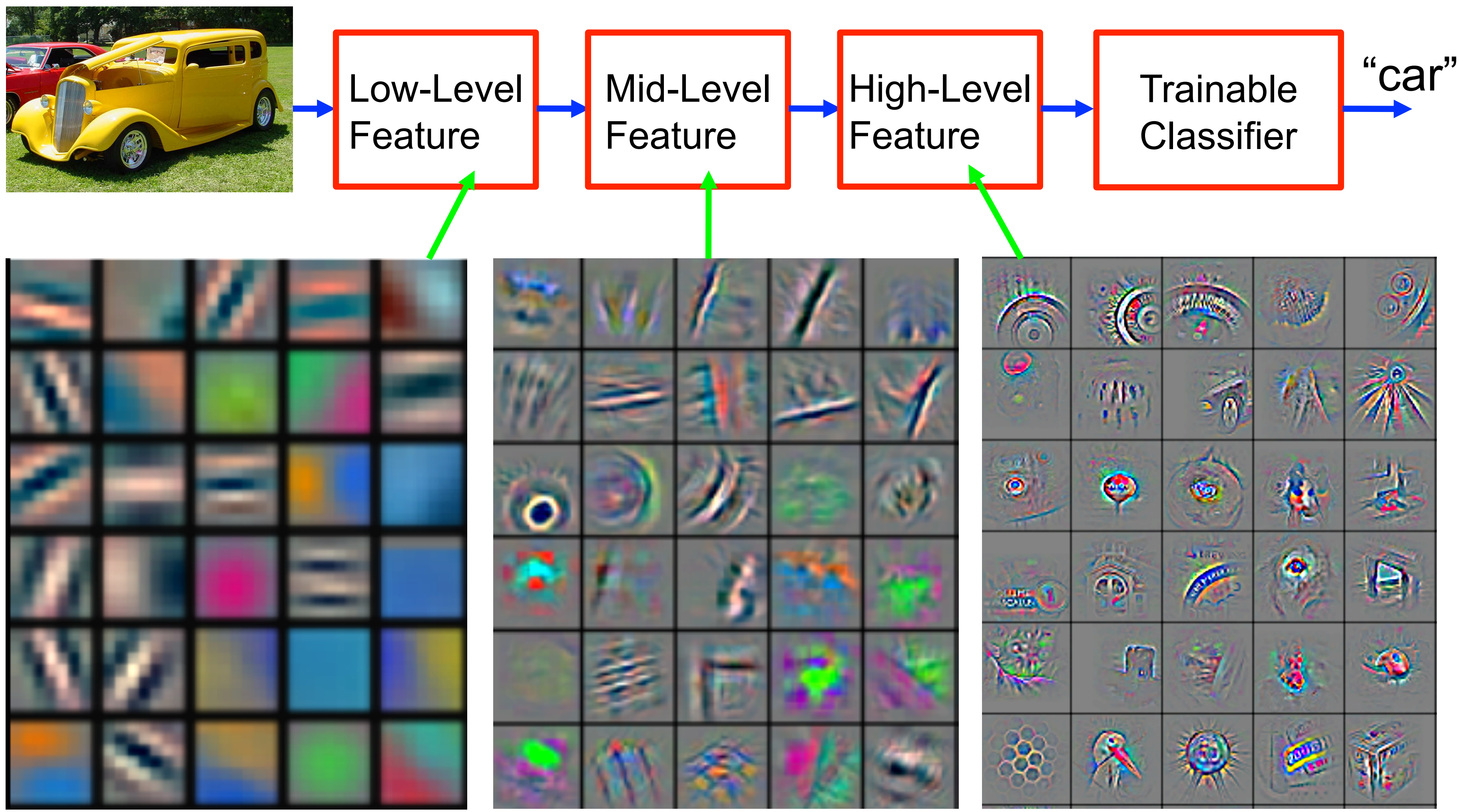
**Idea 2: Compositions**

- Deep Learning
- Grammar models
- Scattering transforms…

$$f(x) = \log(\cos(\exp(\sin^3(x))))$$



# Deep Learning = Hierarchical Compositionality




Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

# So what is Deep (Machine) Learning?

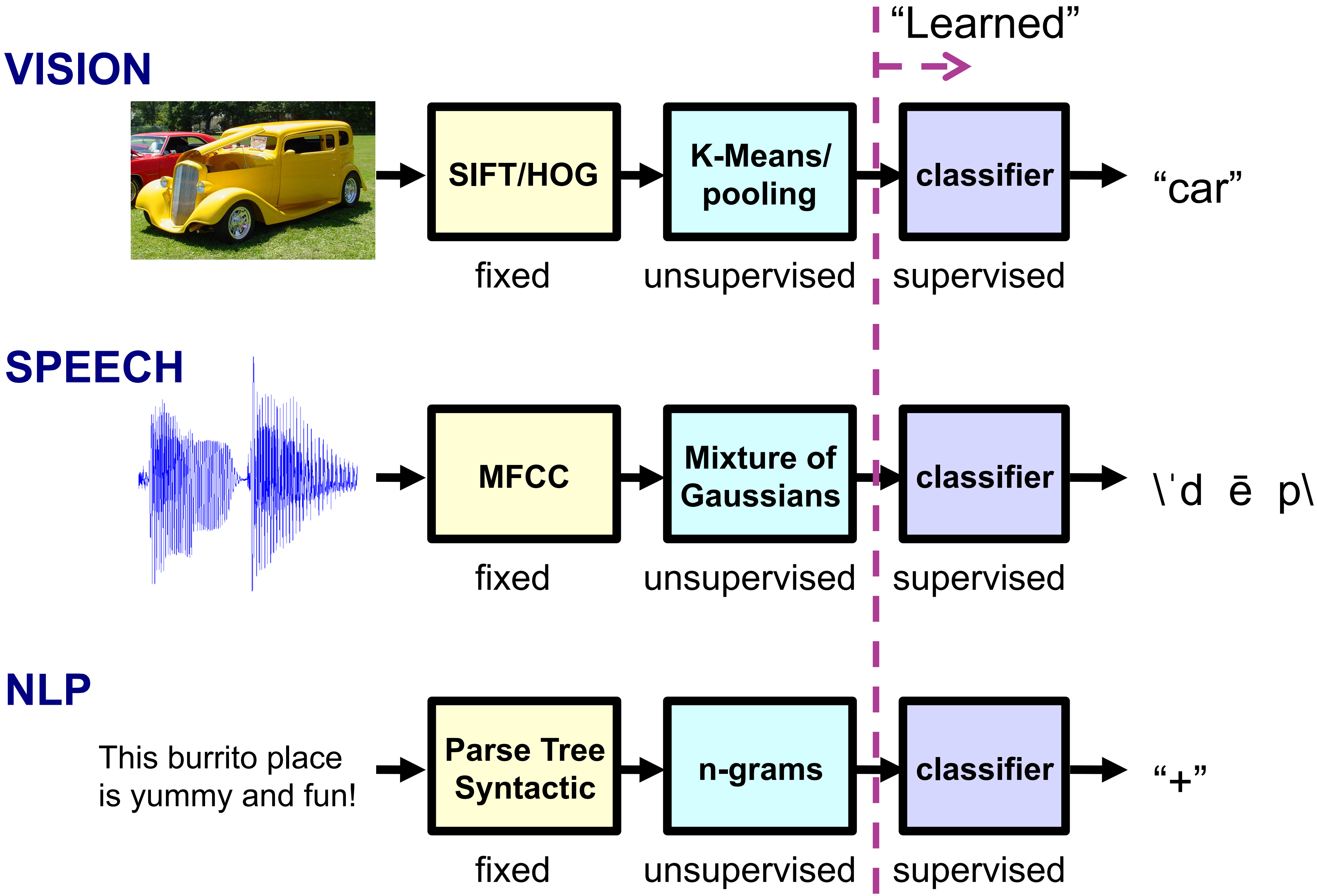
- A few different ideas:

- (Hierarchical) Compositionality
  - Cascade of non-linear transformations
  - Multiple layers of representations

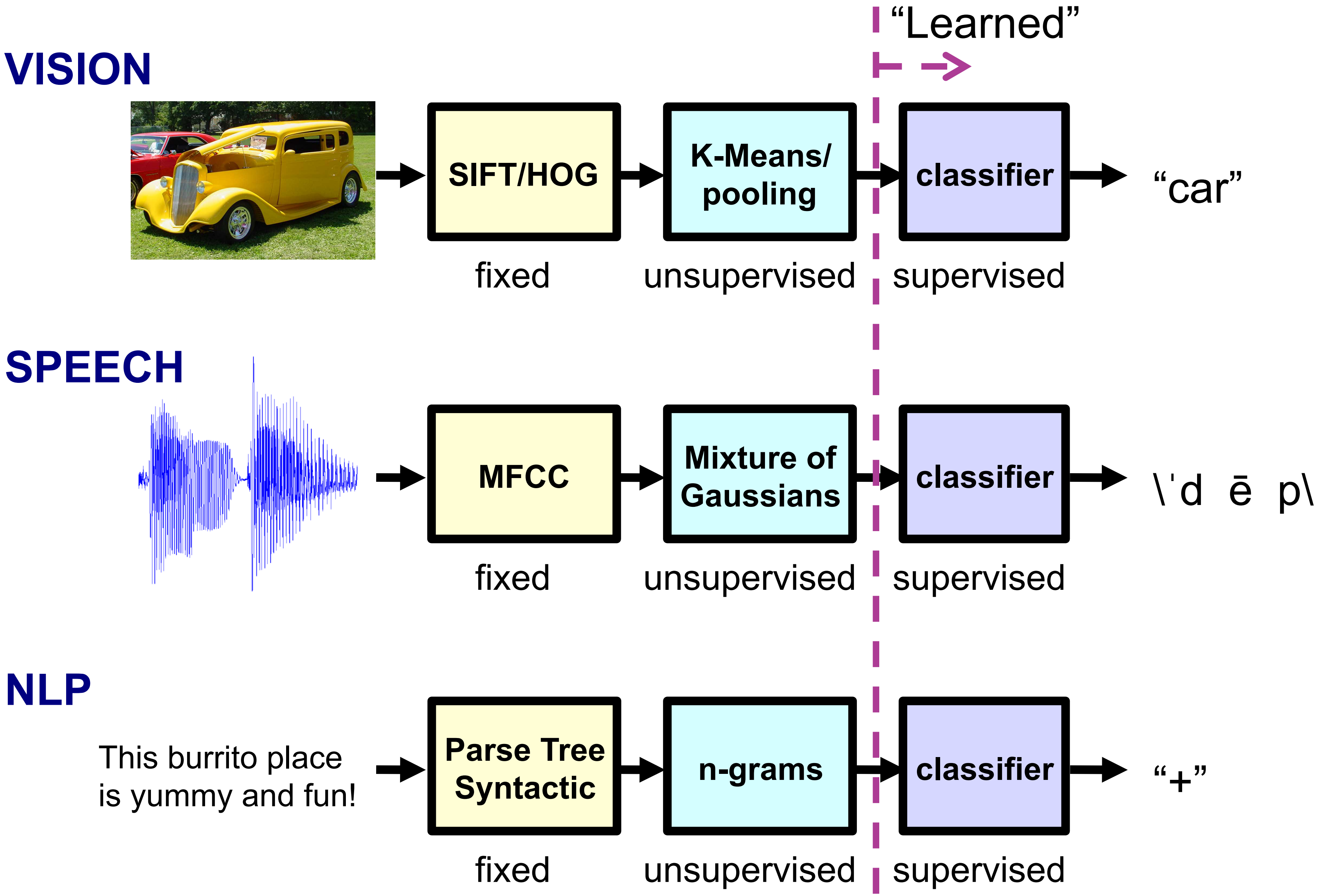
- End-to-End Learning
  - Learning (goal-driven) representations
  - Learning to feature extraction

- Distributed Representations
  - No single neuron "encodes" everything
  - Groups of neurons work together

# Traditional Machine Learning

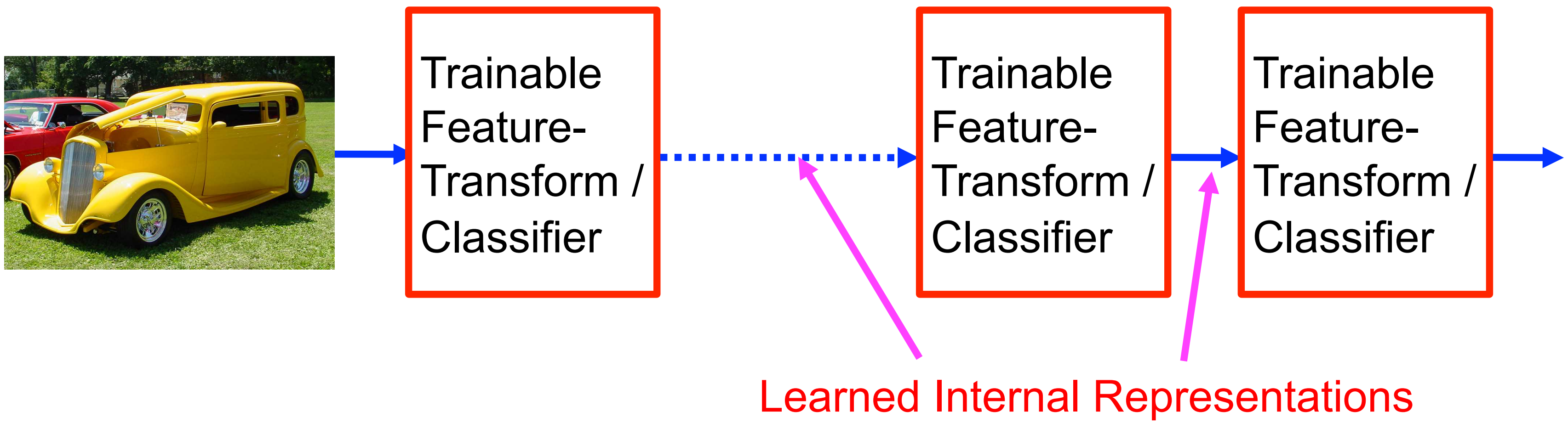


# Deep Learning = End-to-End Learning



# Deep Learning = End-to-End Learning

- A hierarchy of trainable feature transforms
  - Each module transforms its input representation into a higher-level one
  - High-level features are more global and more invariant
  - Low-level features are shared among categories



# Today's lecture

- Supervised learning
- Global representations
- Hierarchical representations
- Learning features
- Compositionality of features
- Classification problem (with SVM)
- End-to-end learning

Crédits pour la majorité des transparents qui suivent: D. Batra, R. Fergus, D. Larlus, Y. LeCun, M. Renzato, HKUST