

Image and Region Categorization

Computer Vision
CS 543 / ECE 549
University of Illinois

Derek Hoiem

Last classes

- Object instance recognition: localizing an object instance in an image
- Face recognition: matching one face image to another
- This week and next: mapping images and regions to categories

This week: image categorization

- Overview of image and region categorization
 - Task description
 - What is a category
- Example of spatial pyramids bag-of-words scene categorizer
- Key concepts: features and classification
- Deep convolutional neural networks (CNNs)

What do you see in this image?



Forest

Categorization lets us describe, predict, or interact with the object based on visual cues



Is it **dangerous**?

How **fast** does it run?

Is it **alive**?

Is it **soft**?

Does it have a **tail**?

Can I **poke with it**?

Theory of categorization

How do we determine if something is a member of a particular category?

- Definitional approach
- Prototype approach
- Exemplar approach

Definitional approach: classical view of categories

- Plato & Aristotle
 - Categories are defined by a list of properties shared by all elements in a category
 - Category membership is binary
 - Every member in the category is equal



Aristotle by Francesco Hayez

The Categories (Aristotle)

Prototype Model

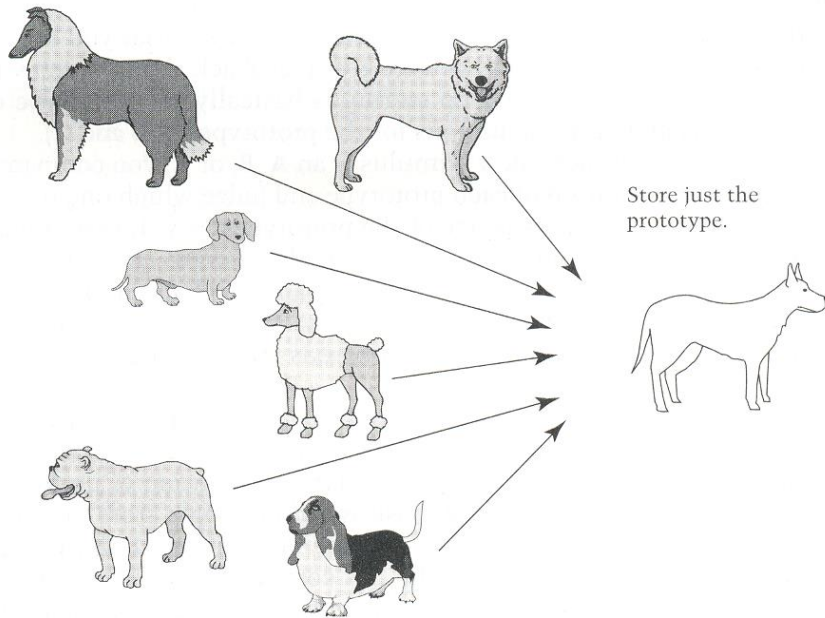


Figure 7.3. Schematic of the prototype model. Although many exemplars are seen, only the prototype is stored. The prototype is updated continually to incorporate more experience with new exemplars.

Category judgments are made by comparing a new exemplar to the prototype.

Exemplars Model

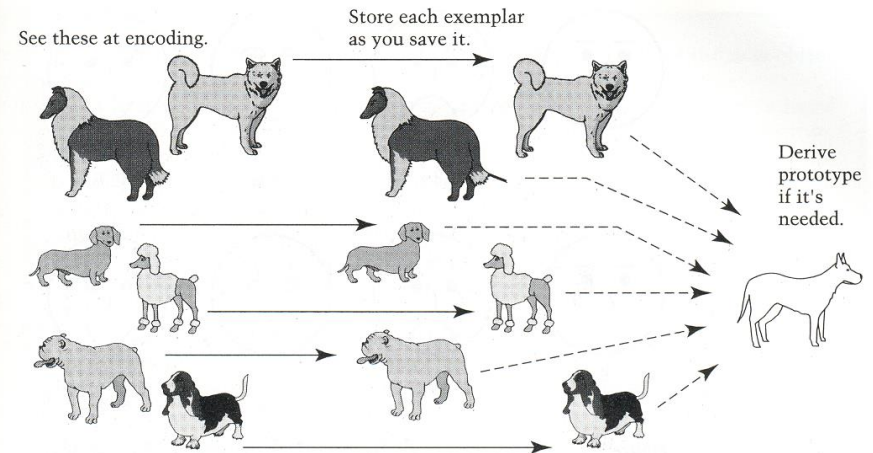


Figure 7.4. Schematic of the exemplar model. As each exemplar is seen, it is encoded into memory. A prototype is abstracted only when it is needed, for example, when a new exemplar must be categorized.

Category judgments are made by comparing a new exemplar to all the old exemplars of a category or to the exemplar that is the most appropriate

Levels of categorization [Rosch 70s]



Definition of Basic Level:

- **Similar shape:** Basic level categories are the highest-level category for which their members have similar shapes.
- **Similar motor interactions:** ... for which people interact with its members using similar motor sequences.
- **Common attributes:** ... there are a significant number of attributes in common between pairs of members.

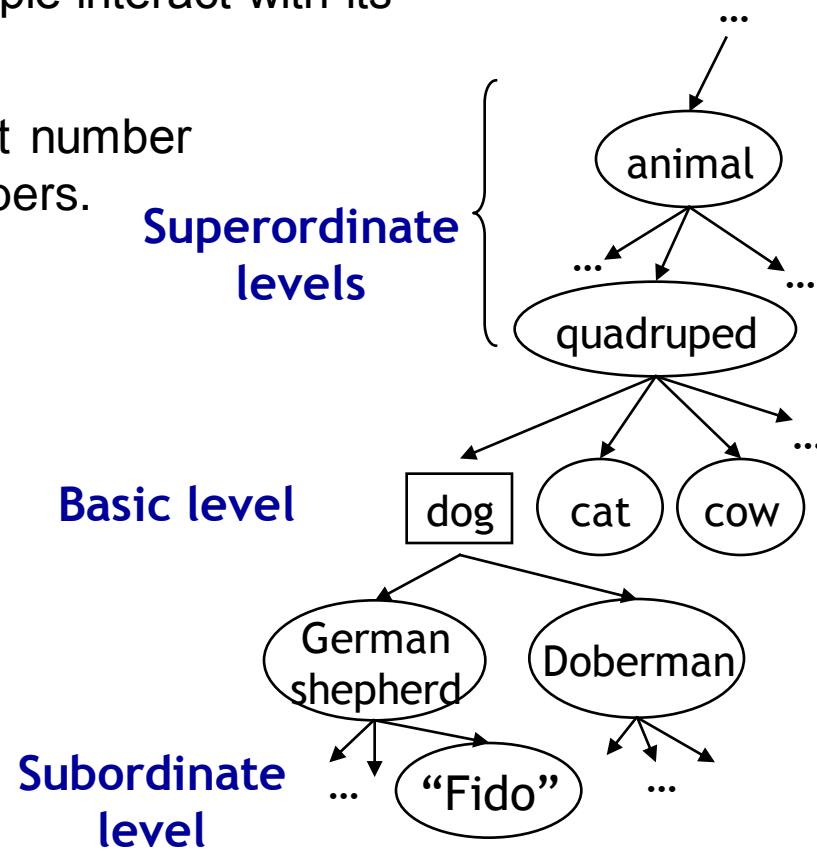
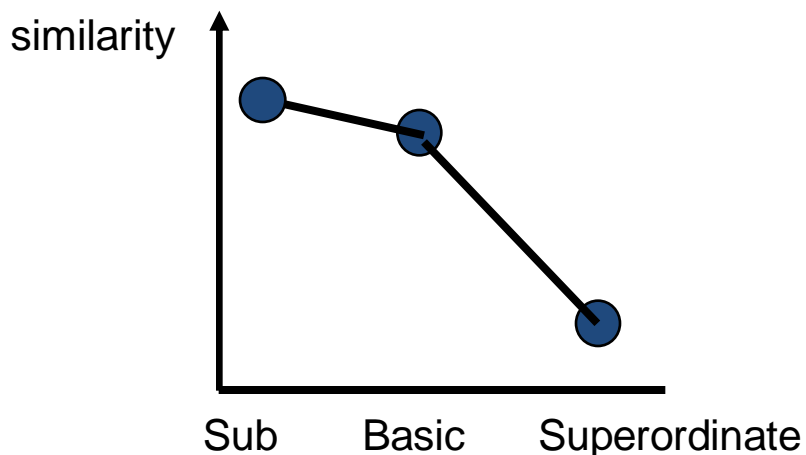


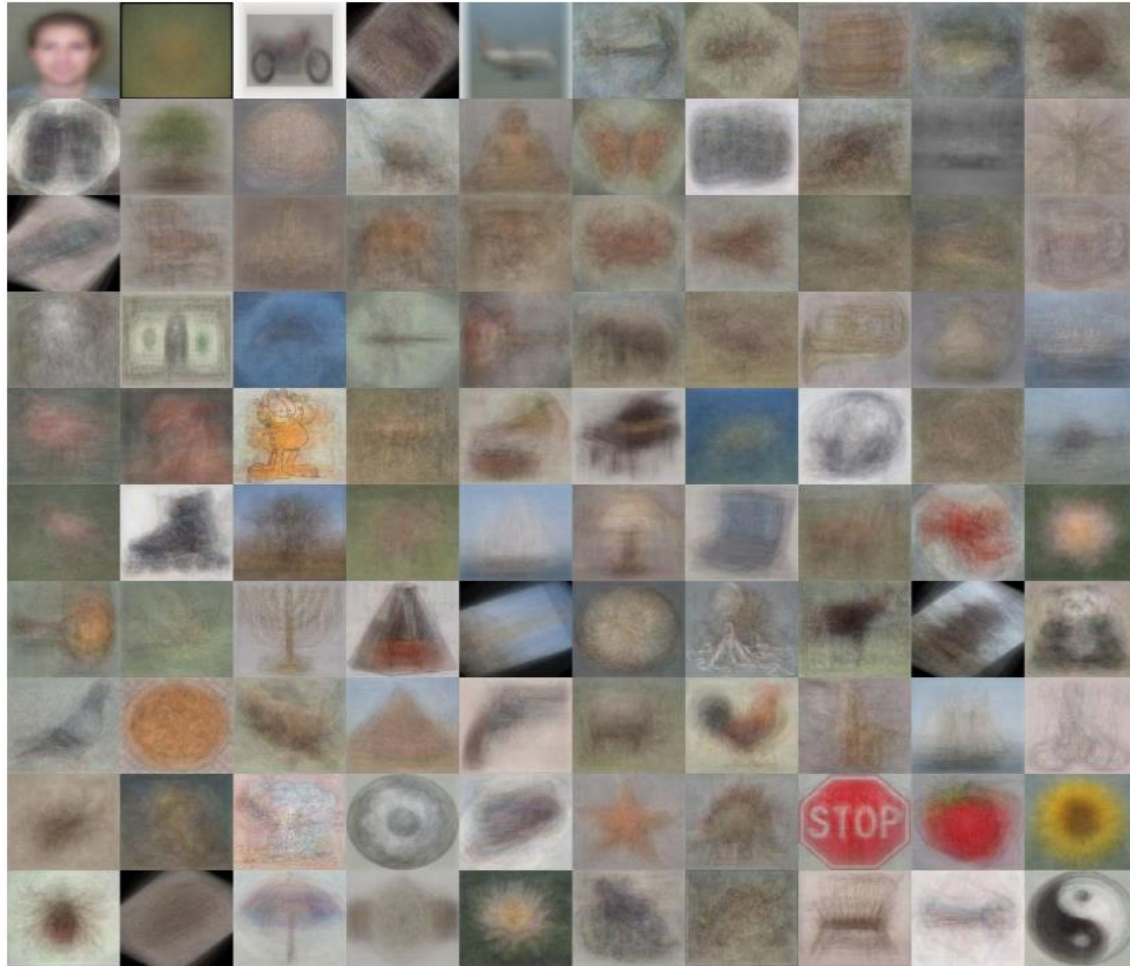
Image categorization

- Cat vs Dog



Image categorization

- Object recognition



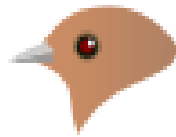
Caltech 101 Average Object Images

Image categorization

- Fine-grained recognition



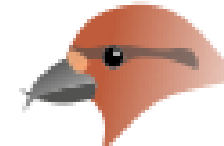
Generalist



Insect catching



Grain eating



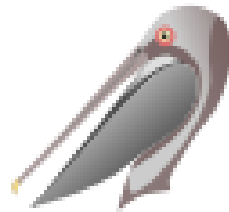
Coniferous-seed eating



Nectar feeding



Chiseling



Dip netting



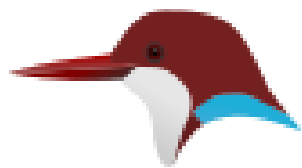
Surface skimming



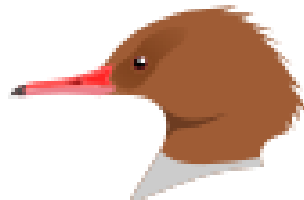
Scything



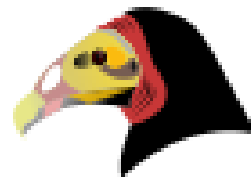
Probing



Aerial fishing



Pursuit fishing



Scavenging



Raptorial



Filter feeding

Image categorization

- Place recognition



Places Database [[Zhou et al. NIPS 2014](#)]

Image categorization

- Visual font recognition



Image categorization

- Image style recognition



HDR



Macro



Baroque



Rococo



Vintage



Noir



Northern Renaissance



Cubism



Minimal



Hazy



Impressionism



Post-Impressionism



Long Exposure



Romantic



Abs. Expressionism



Color Field Painting

Flickr Style: 80K images covering 20 styles.

Wikipaintings: 85K images for 25 art genres.

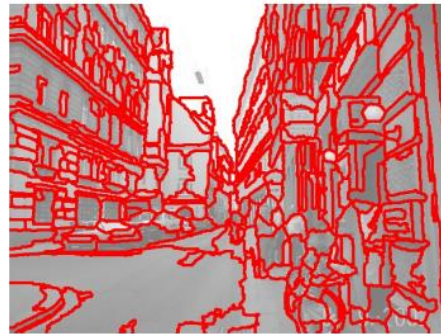
[[Karayev et al. BMVC 2014](#)]

Region categorization

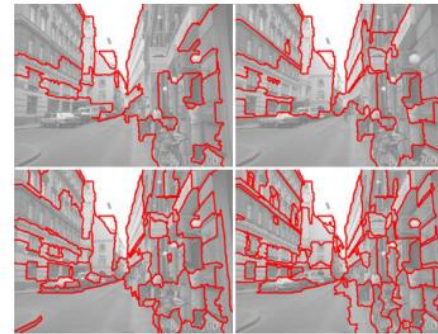
- Layout prediction



Input



Superpixels



Multiple Segmentations



Surface Layout

Assign regions to orientation

Geometric context [[Hoiem et al. IJCV 2007](#)]



a



b



c



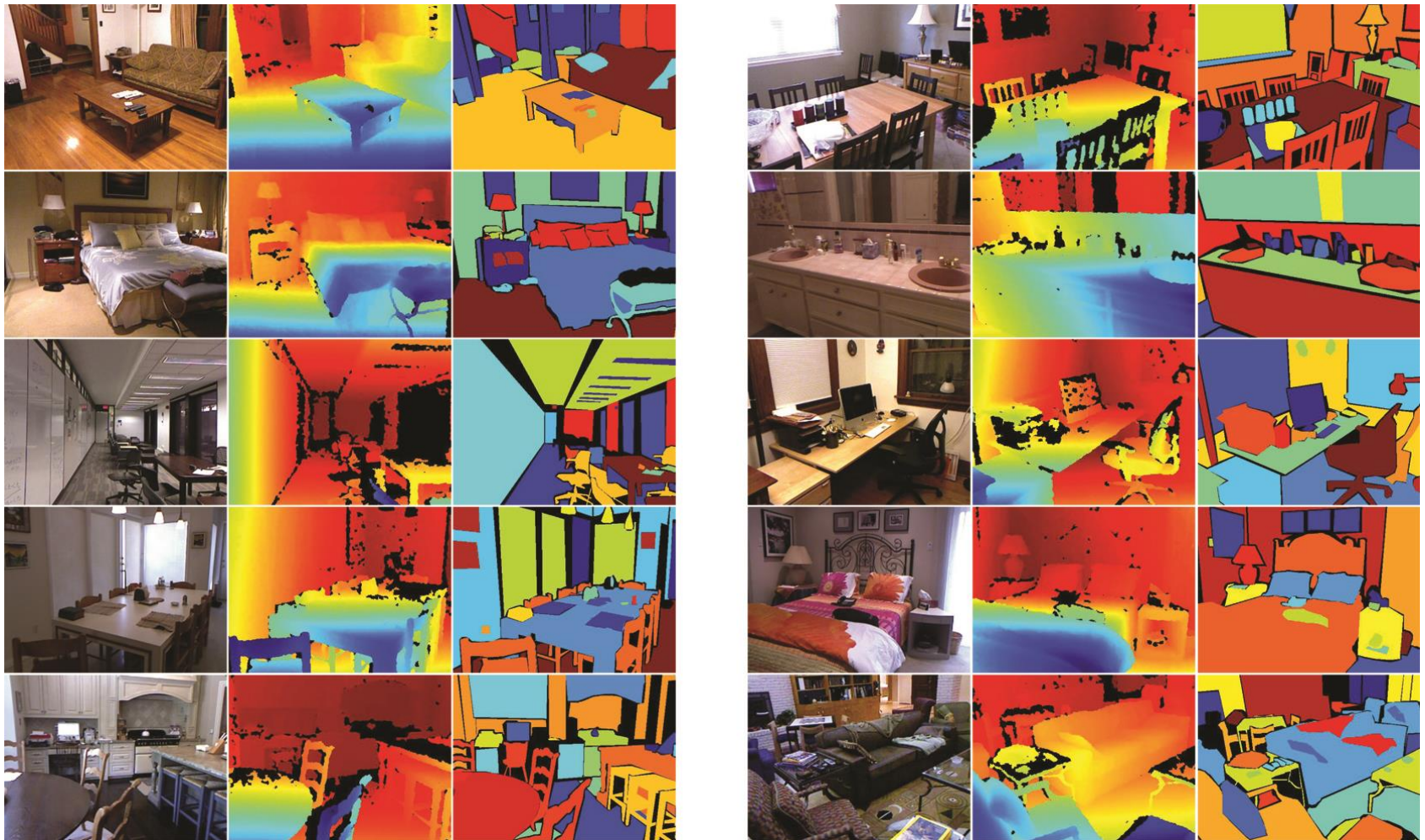
d

Assign regions to depth

Make3D [[Saxena et al. PAMI 2008](#)]

Region categorization

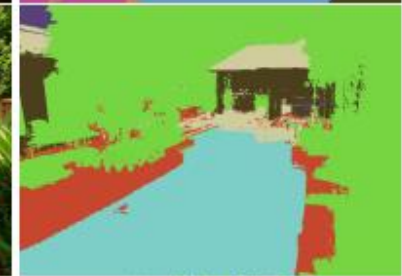
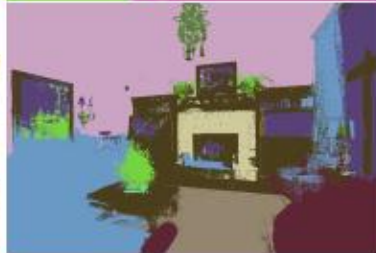
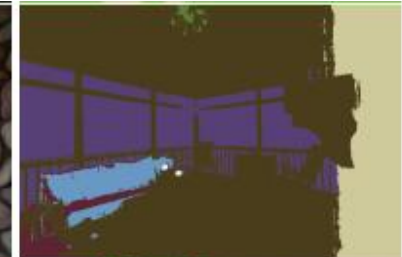
- Semantic segmentation from RGBD images



[Silberman et al. ECCV 2012]

Region categorization

- Material recognition

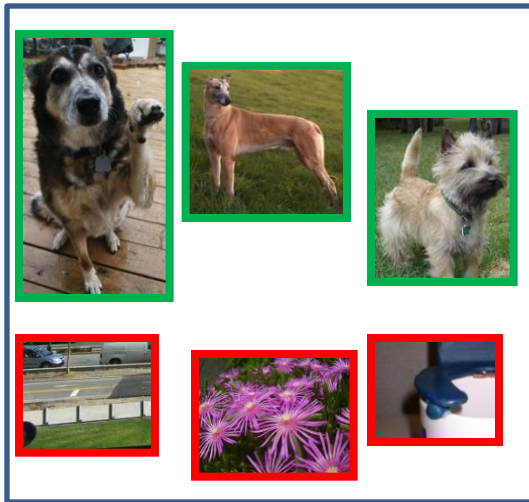


[Bell et al. CVPR 2015]

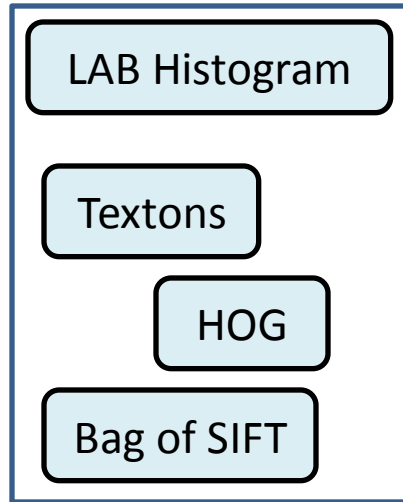
Many vision problems involve categorization

- Image: Classify as indoor/outdoor, which room, what objects are there, etc.
- Object Detection: classify location (bounding box or region) as object or non-object
- Semantic Segmentation: classify pixel into an object, material, part, etc.
- Action Recognition: classify a frame or sequence into an action type
- ...

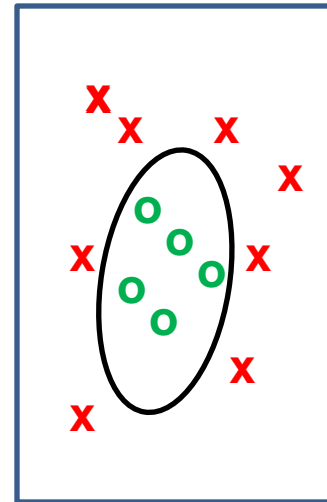
Categorization from supervised learning



Examples



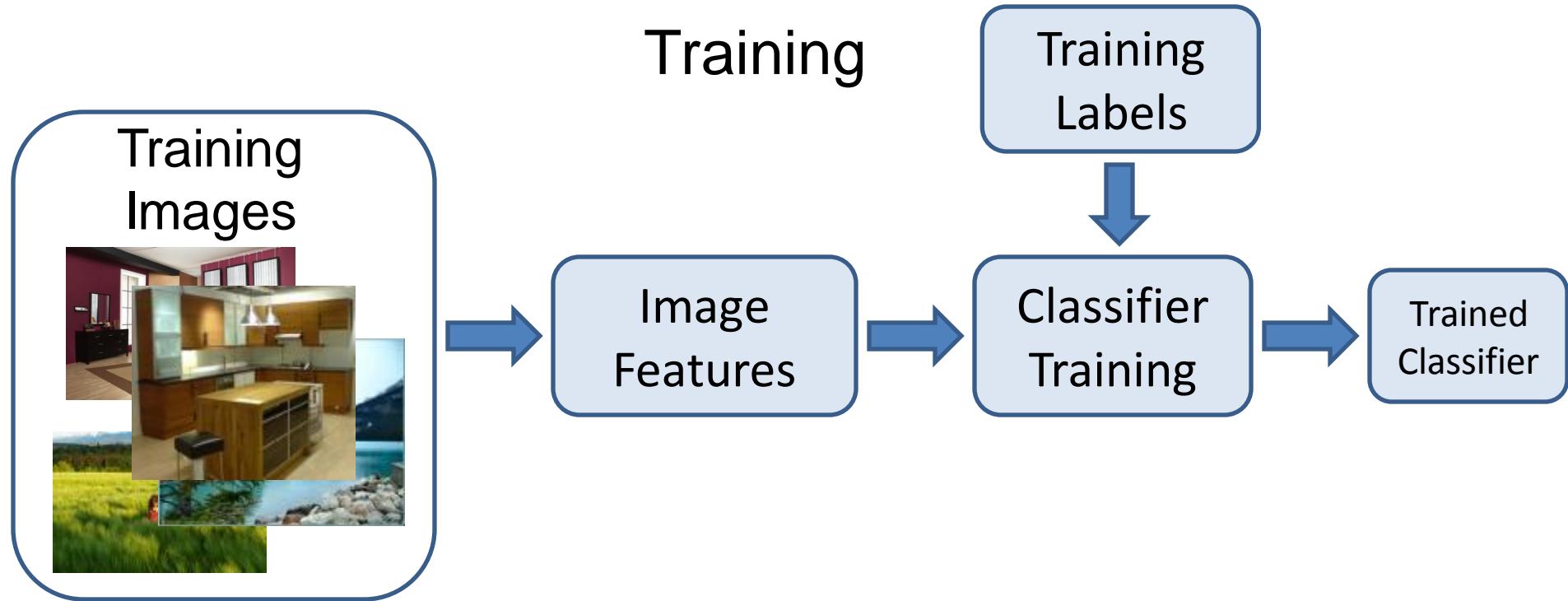
+ Image Features



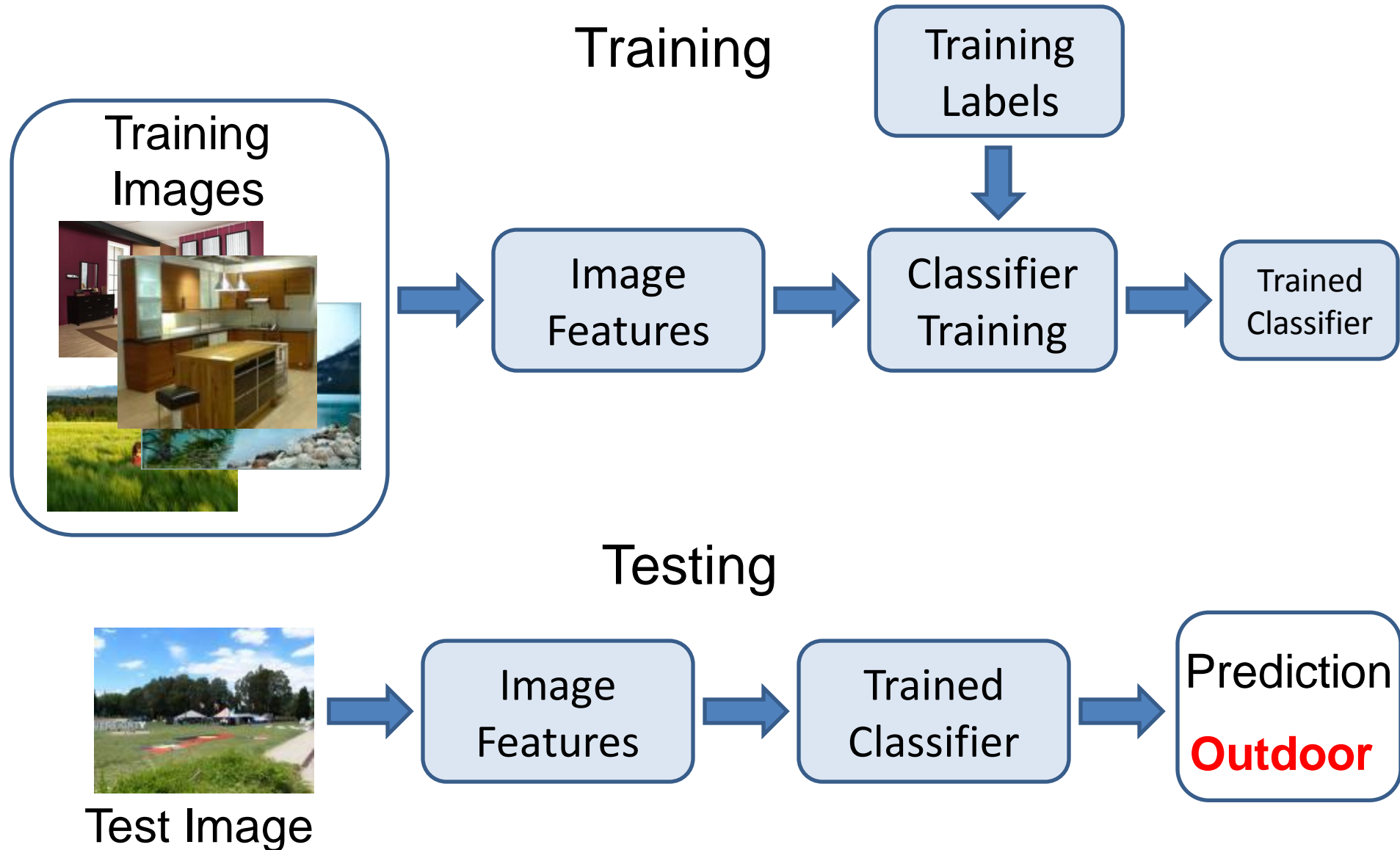
+ Classifier

= Category label

Training phase



Testing phase

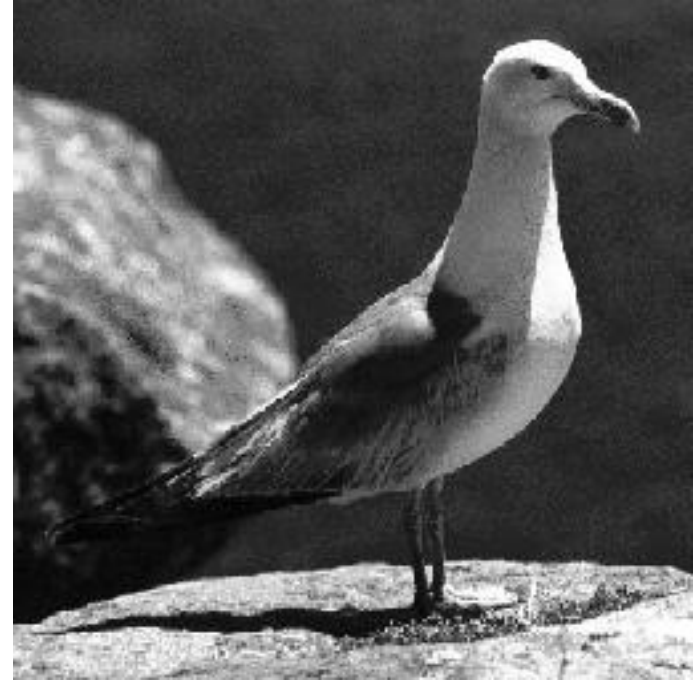
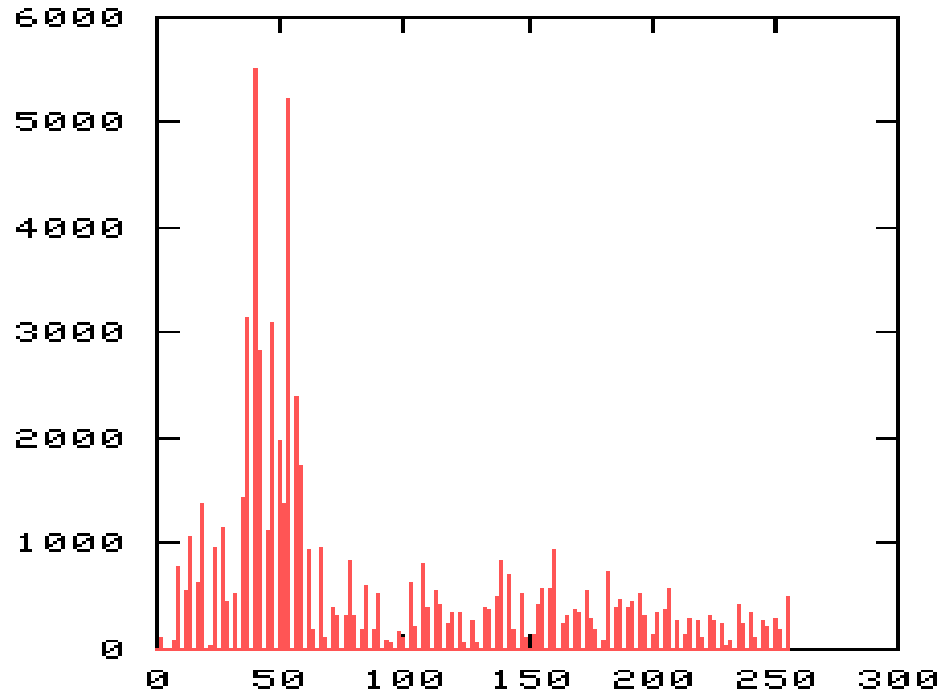


Example: Spatial Pyramid BoW Classifier



- Features: spatially binned histograms of clustered SIFT descriptors
- Classifier: SVM

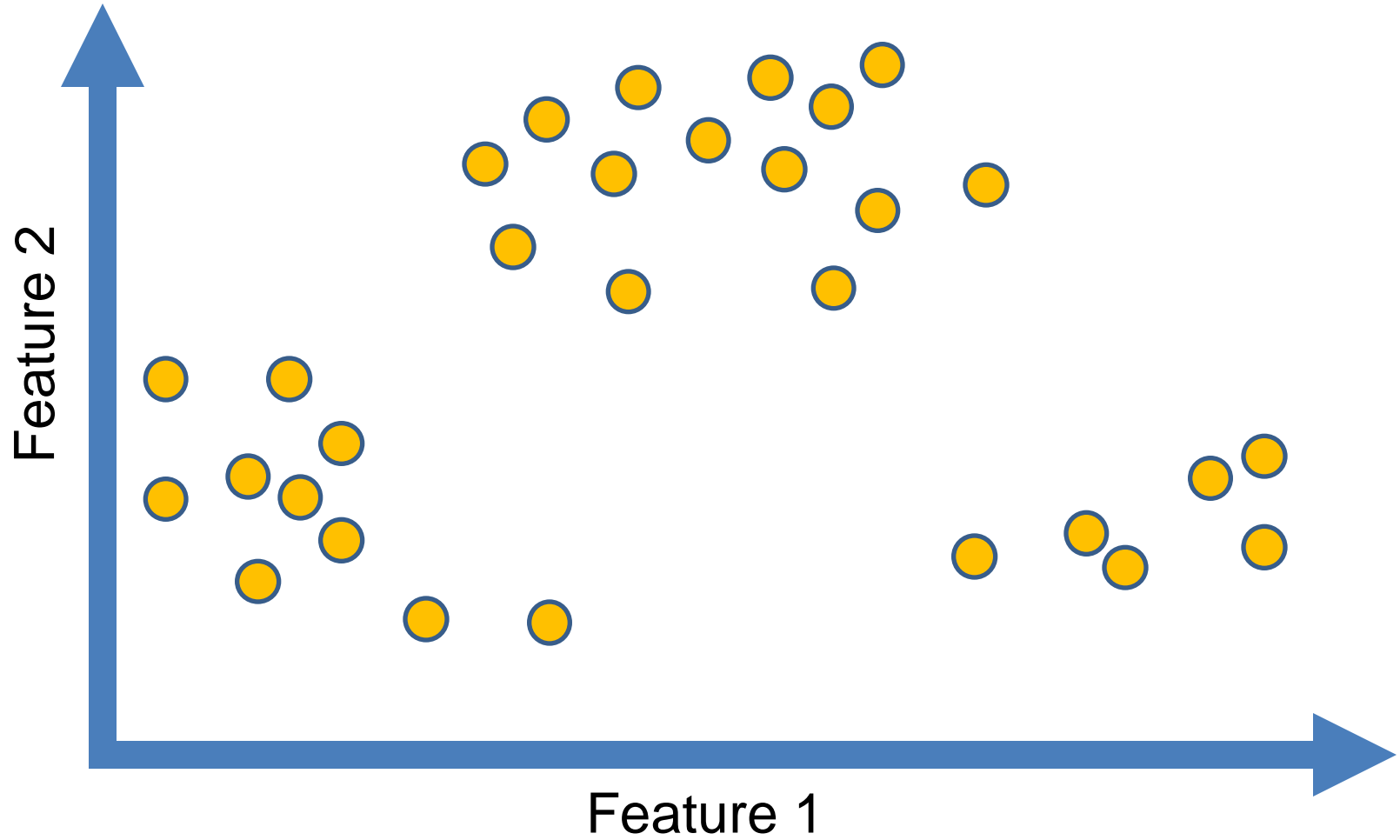
A little background on histograms...



Global histogram

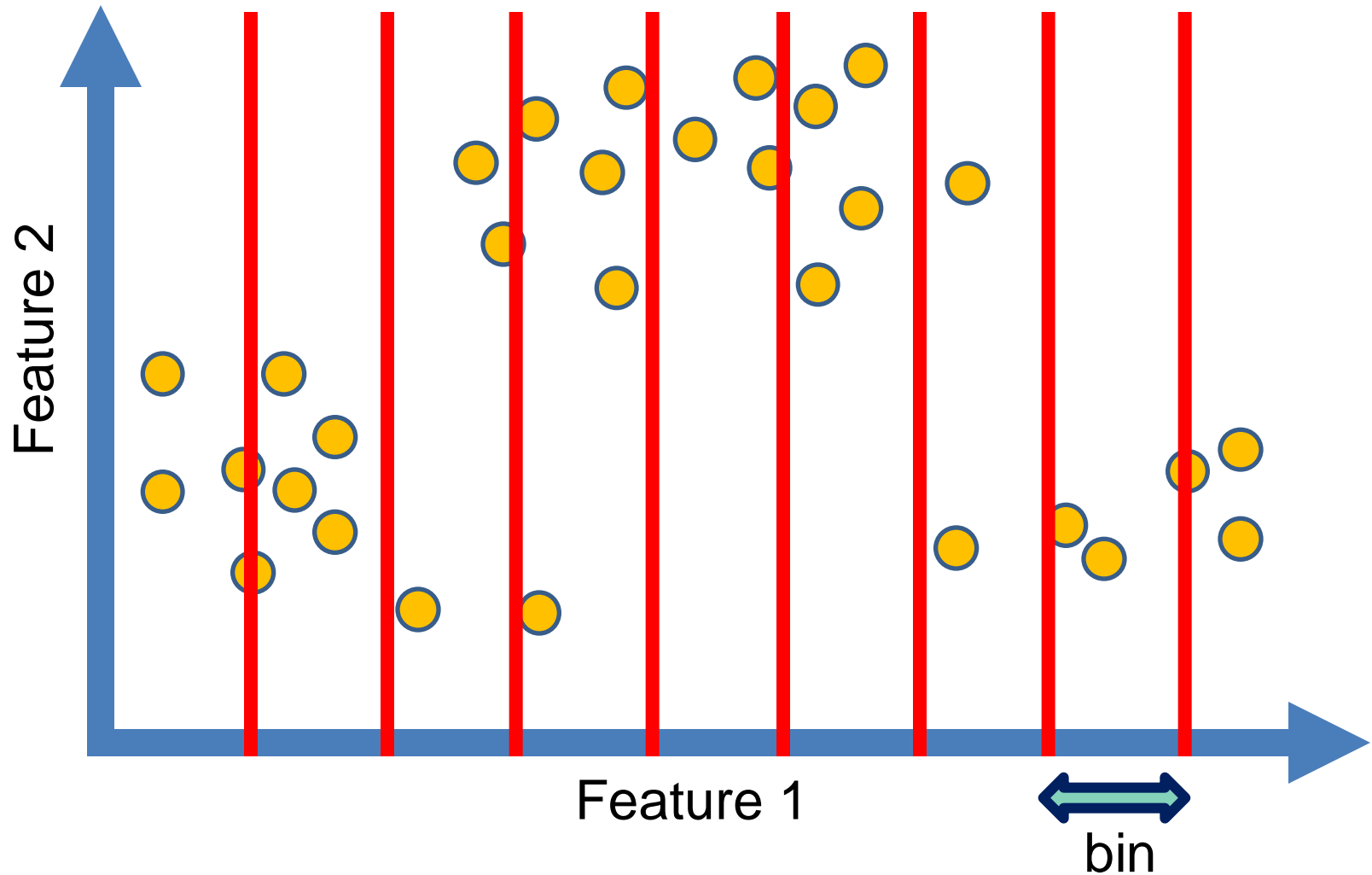
- Represent distribution of features
 - Color, texture, depth, ...

2D histogram



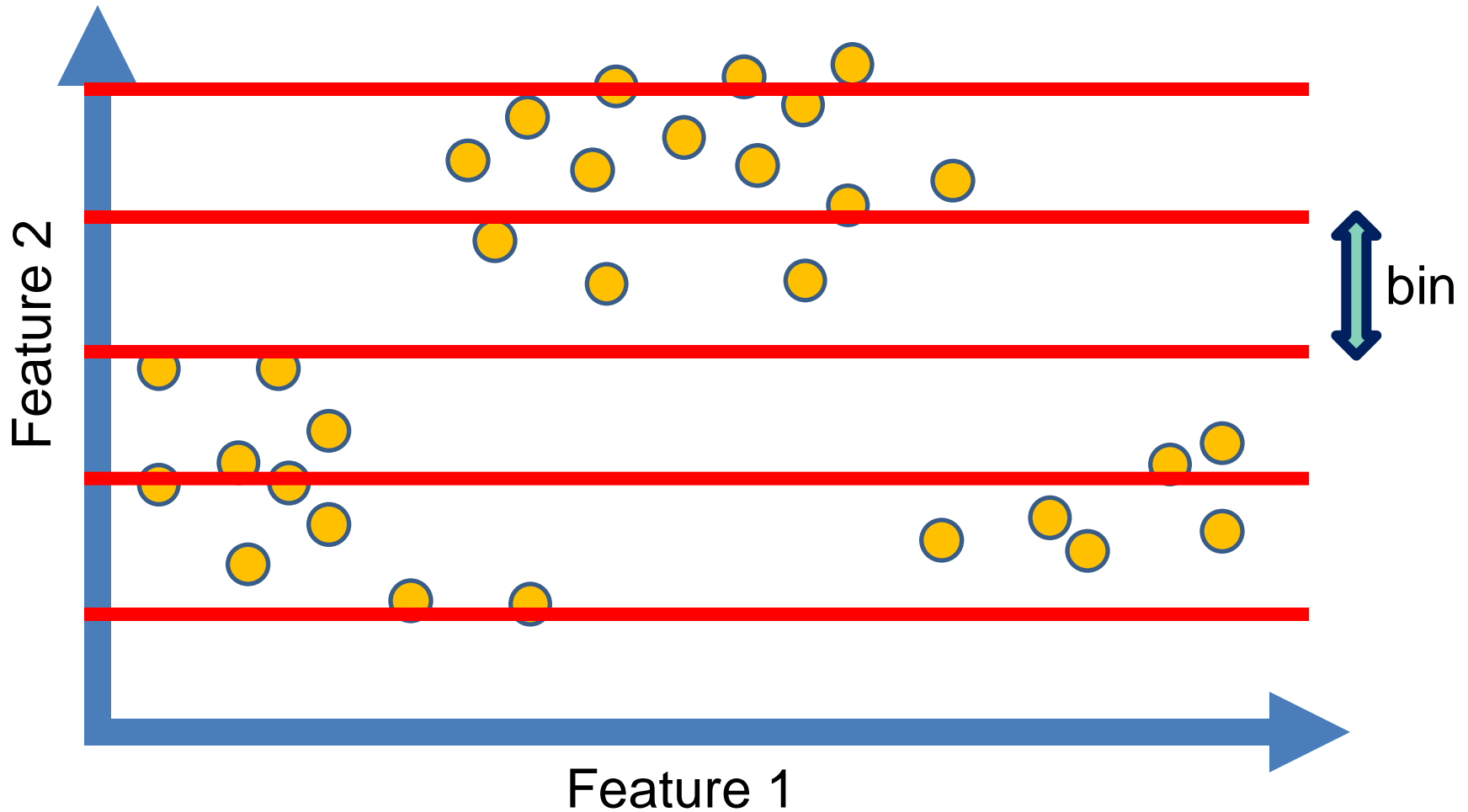
Histogram with marginal binning

- Probability or count of data in each bin
- Marginal histogram on feature 1

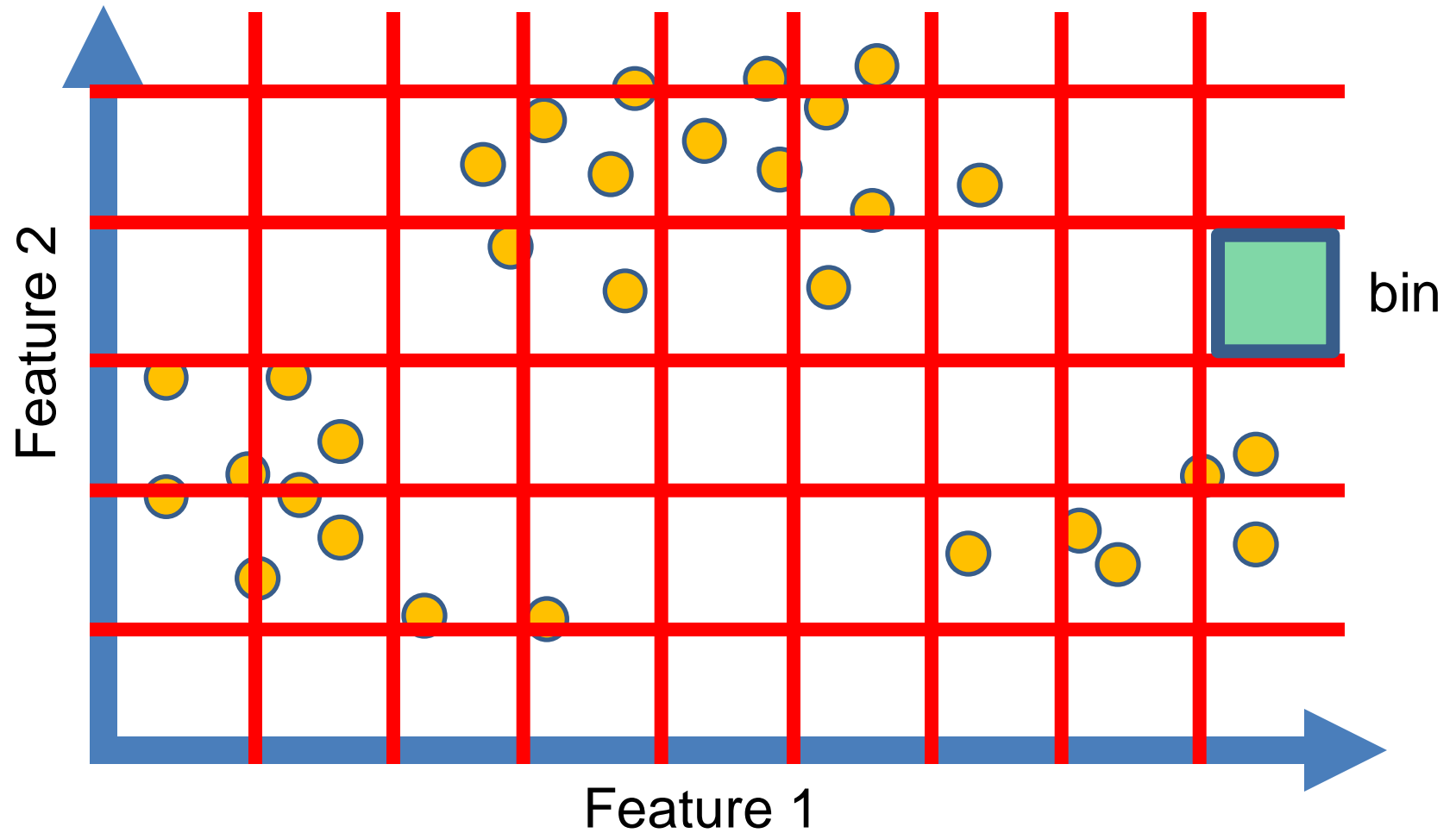


Histogram with marginal binning

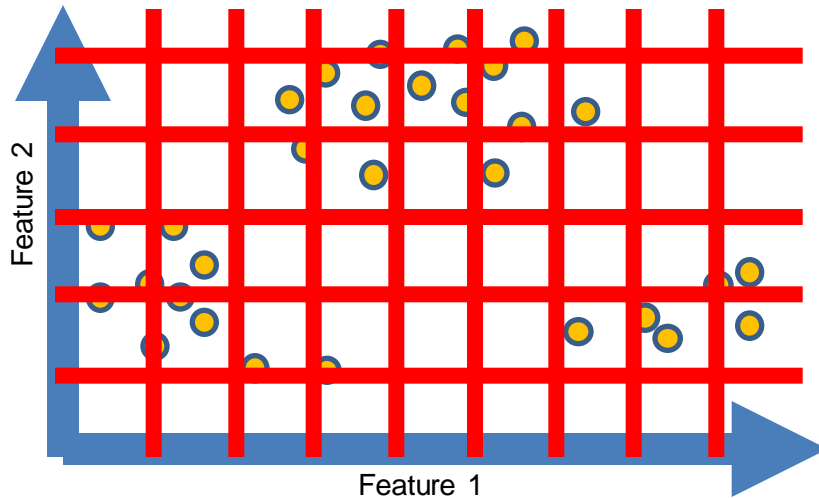
- Marginal histogram on feature 2



Histogram with joint binning

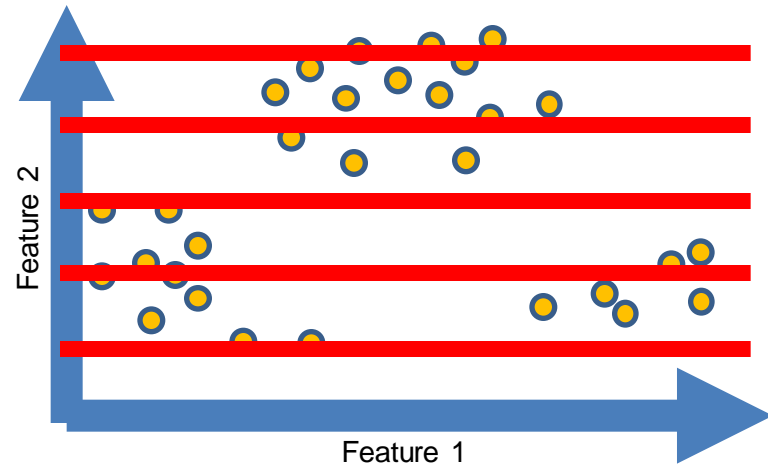
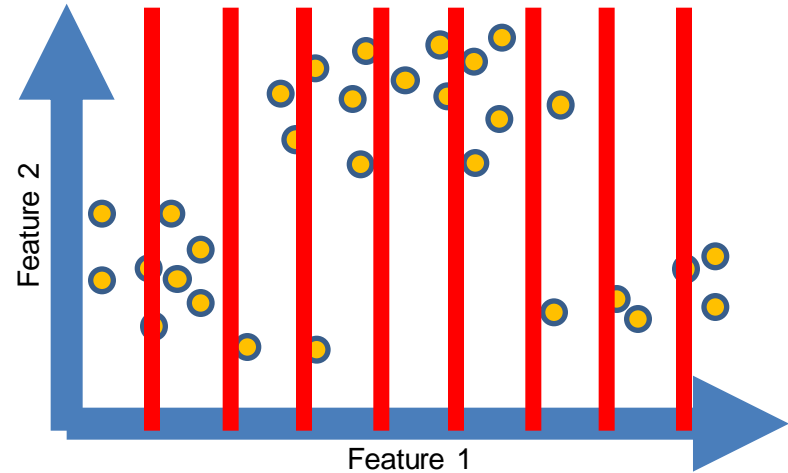


Joint vs marginal binning



Joint histogram

- Requires lots of data
- Loss of resolution to avoid empty bins

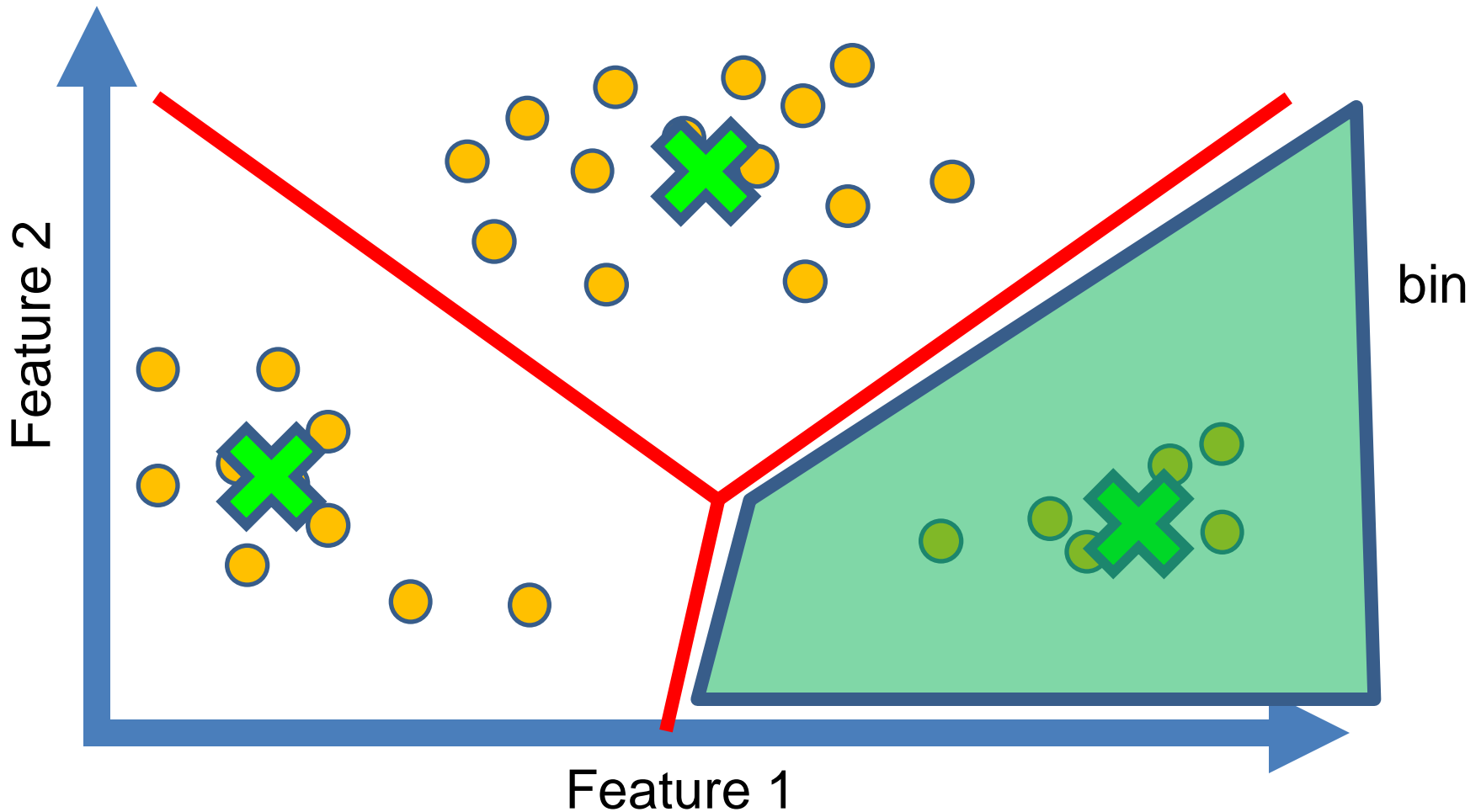


Marginal histogram

- Requires independent features
- More data/bin than joint histogram

Histogram with clustering

- Cluster data (or partition) into K clusters, count how many samples appear in each cluster to get K -dim histogram
- Use the same cluster centers (or partitioning) for all images



Computing histogram distance

- Cosine similarity (dot product of normalized counts)
- Histogram intersection

$$\text{histint}(h_i, h_j) = 1 - \sum_{m=1}^K \min(h_i(m), h_j(m))$$

- Chi-squared Histogram matching distance

$$\chi^2(h_i, h_j) = \frac{1}{2} \sum_{m=1}^K \frac{[h_i(m) - h_j(m)]^2}{h_i(m) + h_j(m)}$$

- Earth mover's distance
(Cross-bin similarity measure)
 - minimal cost paid to transform one distribution into the other

Histograms: implementation issues

- Quantization
 - Grids: fast but applicable only with few dimensions
 - Clustering: slower but can quantize data in higher dimensions



Few Bins

Need less data

Coarser representation

Many Bins

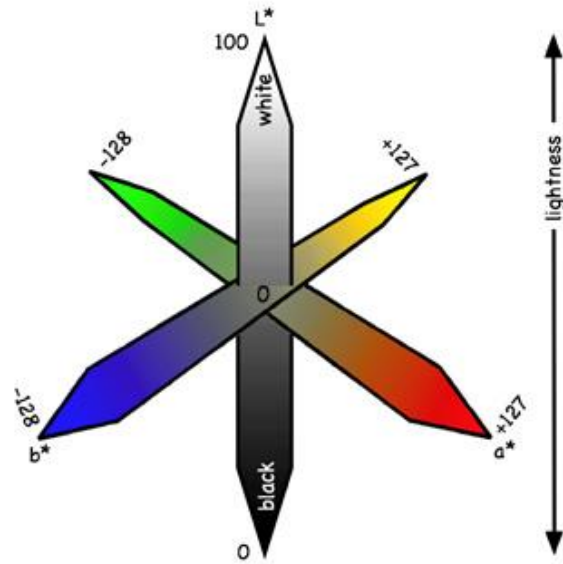
Need more data

Finer representation

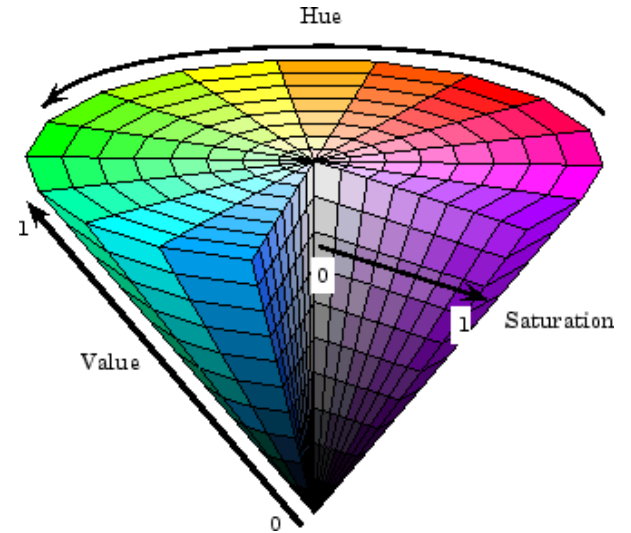
- Matching
 - Histogram intersection or Euclidean/Cosine may be faster
 - Chi-squared often works better
 - Earth mover's distance is good for when nearby bins represent similar values

What kind of things do we compute histograms of?

- Color

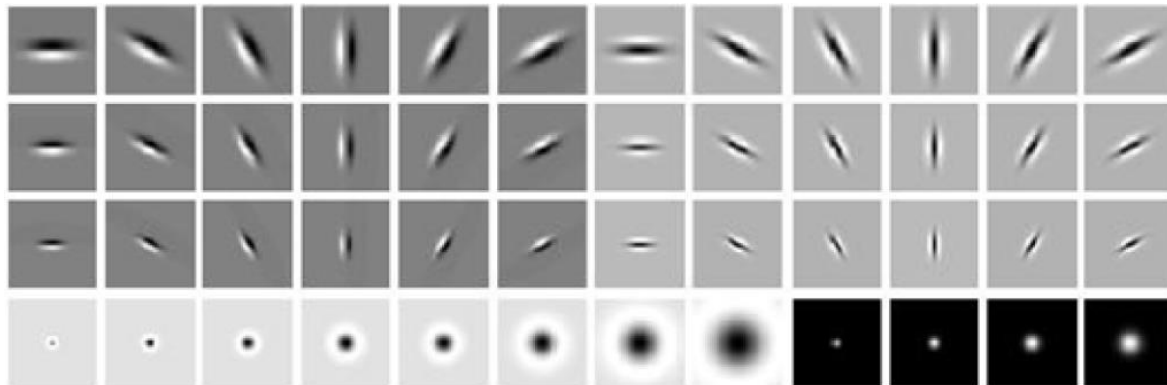


L*a*b* color space



HSV color space

- Texture (filter banks or HOG over regions)



What kind of things do we compute histograms of?

- Histograms of descriptors

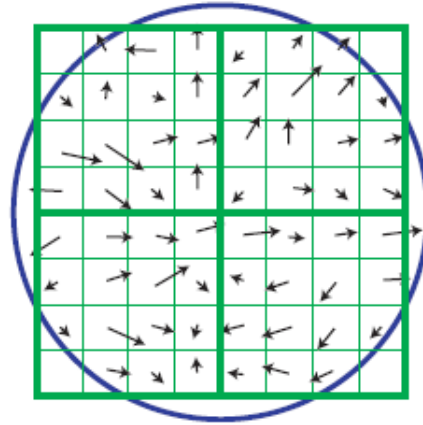
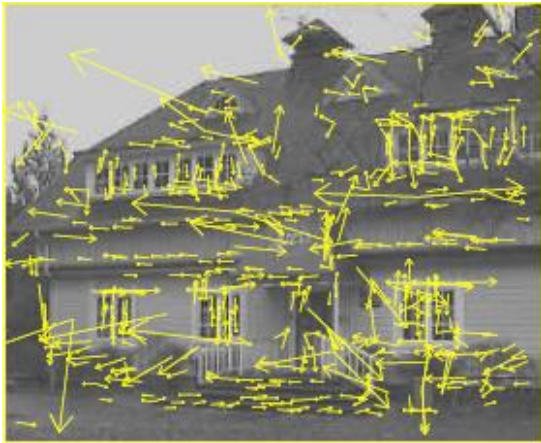
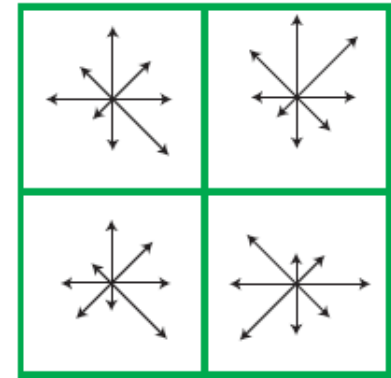


Image gradients



Keypoint descriptor

SIFT – [Lowe IJCV 2004]

- “Bag of visual words”

Image categorization with bag of words

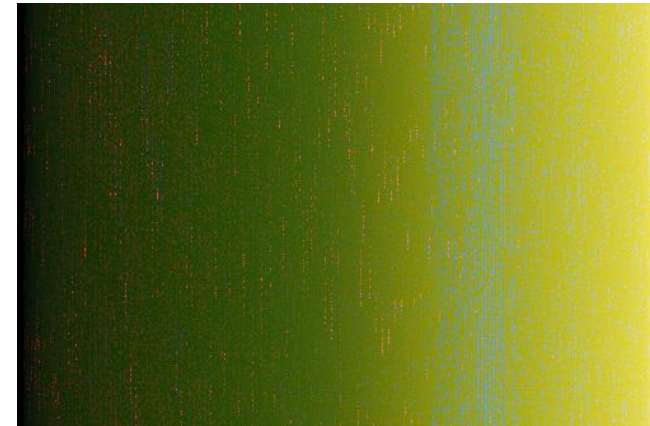
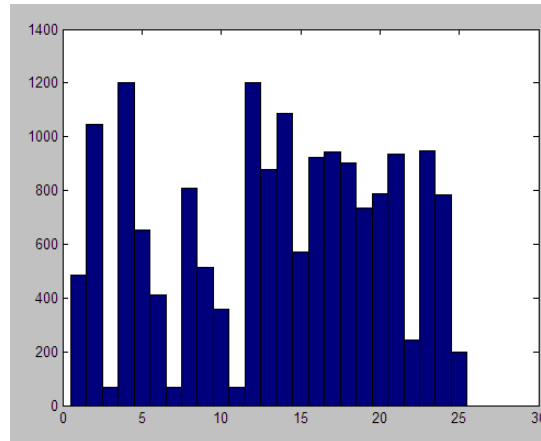
Training

1. Extract keypoints and descriptors for all training images
2. Cluster descriptors
3. Quantize descriptors using cluster centers to get “visual words”
4. Represent each image by normalized counts of “visual words”
5. Train classifier on labeled examples using histogram values as features

Testing

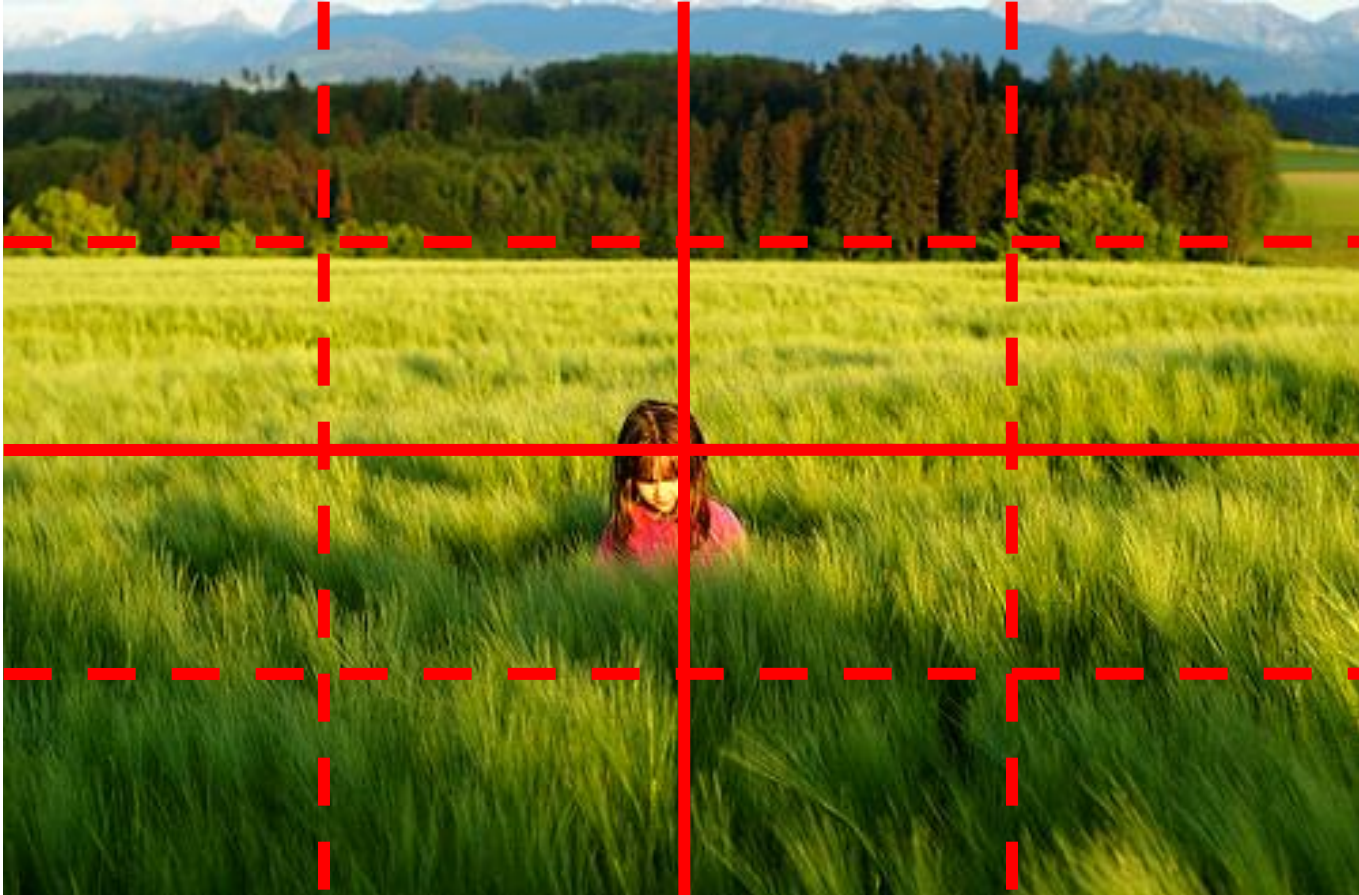
1. Extract keypoints/descriptors and quantize into visual words
2. Compute visual word histogram
3. Compute label or confidence using classifier

But what about spatial layout?



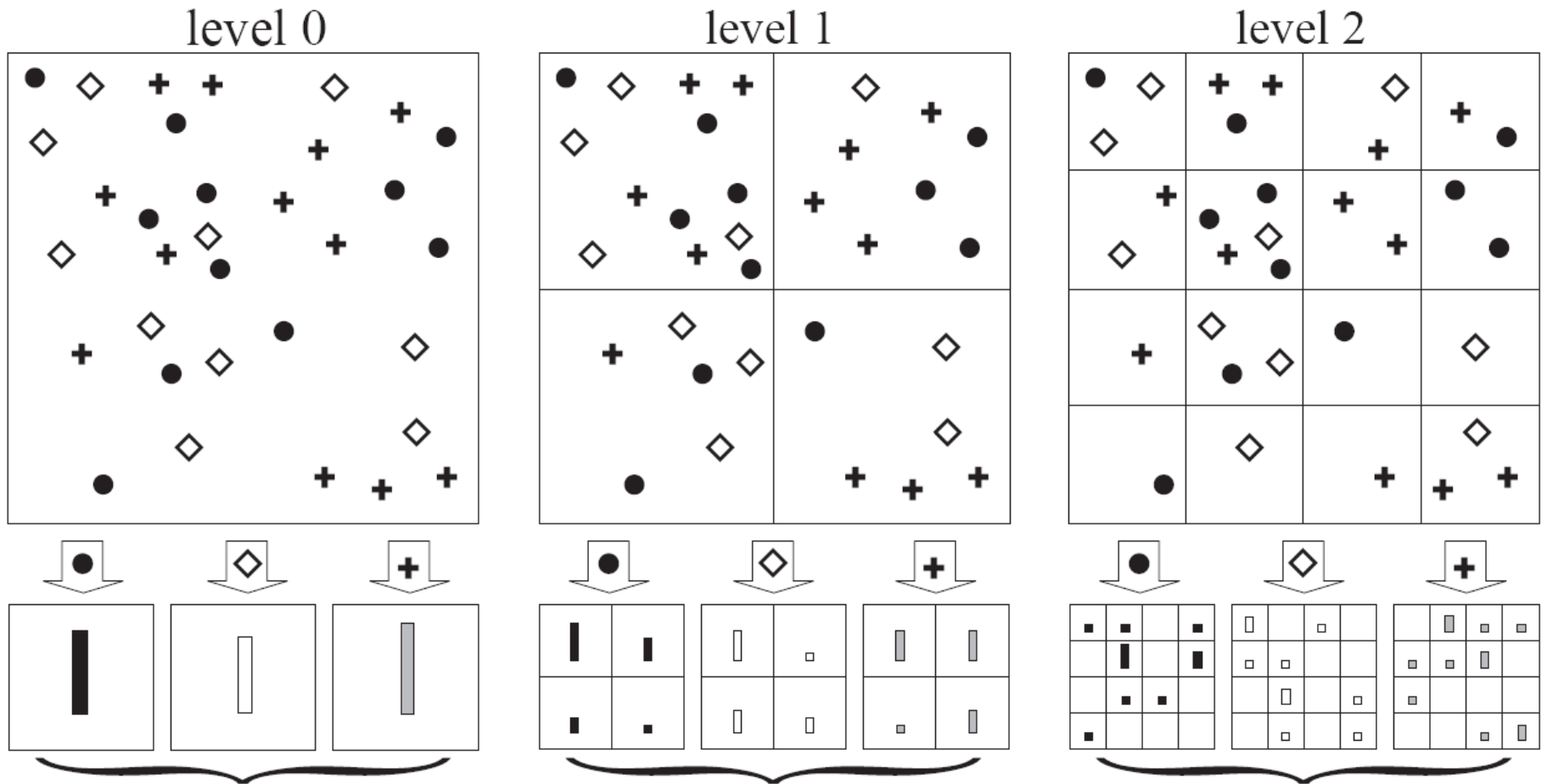
All of these images have the same color histogram

Spatial pyramid



Compute histogram in each spatial bin

Spatial pyramid



Training



Spatial Pyramid BoW

Training Labels

Classifier Training

Trained Classifier

Testing



Test Image

Spatial Pyramid BoW

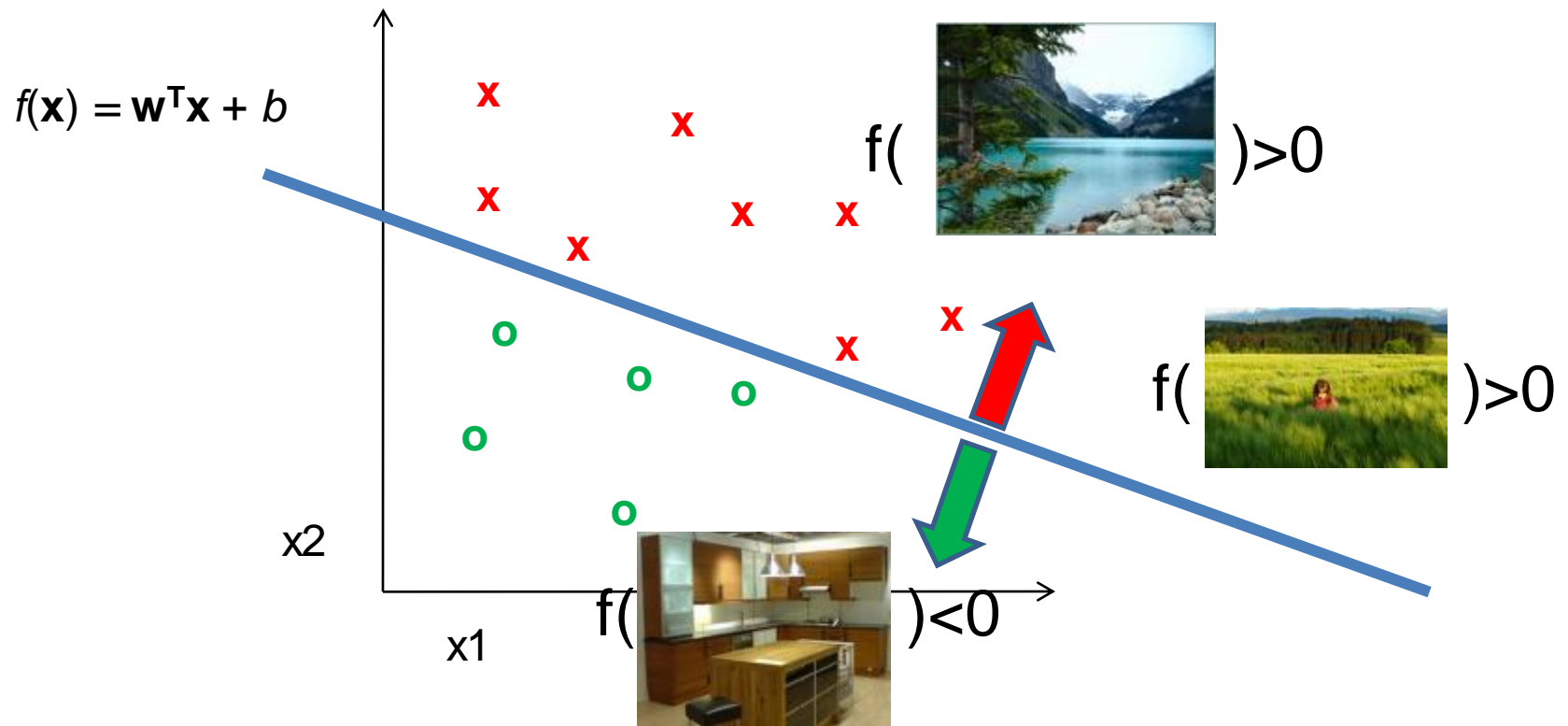
Trained Classifier

Prediction
Outdoor



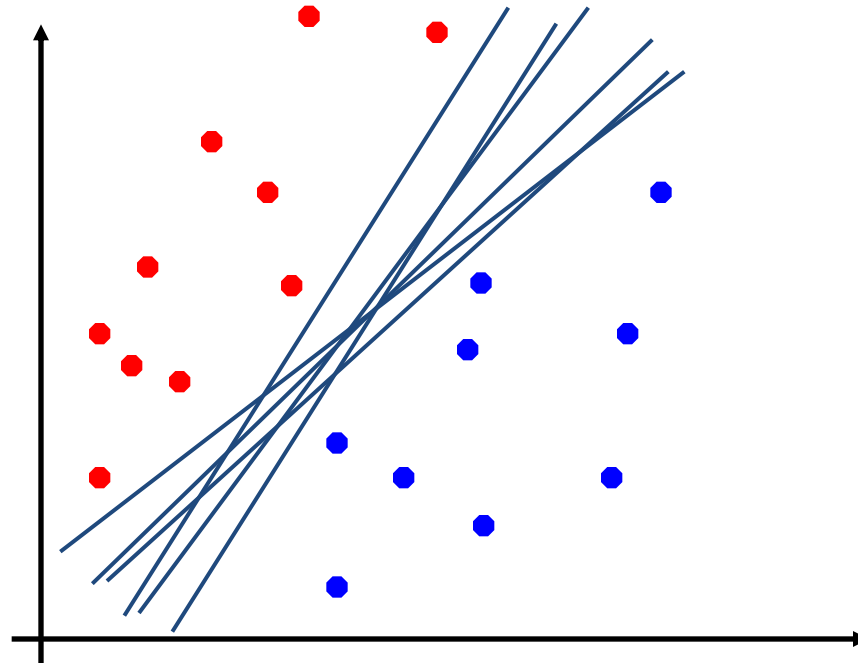
Linear SVM classifier

Find the hyperplane that separate examples of different categories



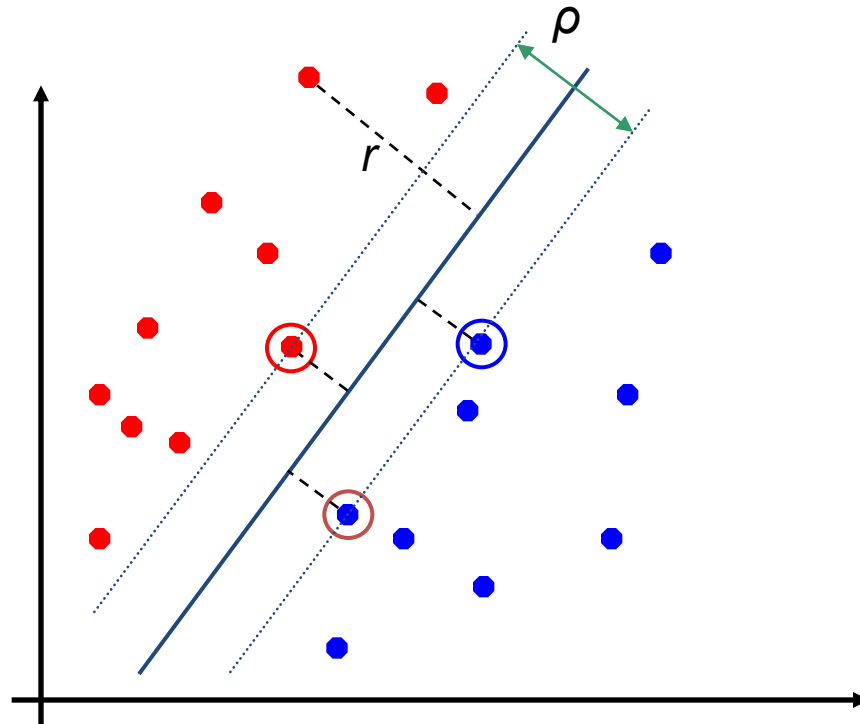
Linear Separators

- Which of the linear separators is best?



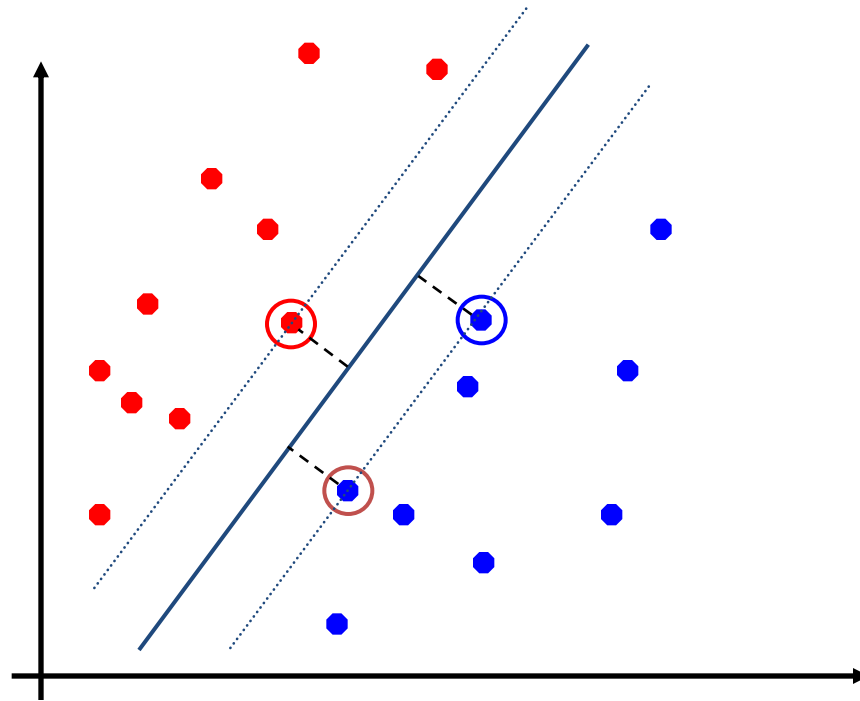
Classification Margin

- Distance from example \mathbf{x}_i to the separator is $r = \frac{\mathbf{w}^T \mathbf{x}_i + b}{\|\mathbf{w}\|}$
- Examples closest to the hyperplane are **support vectors**.
- **Margin** ρ of the separator is the distance between support vectors.



Maximum Margin Classification

- Implies that only support vectors matter; other training examples are ignorable.



Linear SVM Formulation

Find weights that try to get all examples right with a confidence margin while limiting the L2 norm of the weight vector

$$\text{minimize } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \quad \leftarrow \text{Loss and regularization}$$

such that

$$y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

Parameter C trades off between cost of hard examples and penalty on having large feature weights

Linear SVM Dual Formulation

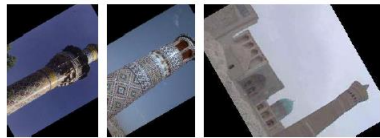
$$\text{maximize } \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

such that

$$0 \leq \alpha_i \leq C$$

$$\sum_{i=1}^n \alpha_i y_i = 0$$

Spatial Pyramids Results



minaret (97.6%)



windsor chair (94.6%)



joshua tree (87.9%)



okapi (87.8%)



cougar body (27.6%)



beaver (27.5%)



crocodile (25.0%)



ant (25.0%)

	Weak features		Strong features (200)	
L	Single-level	Pyramid	Single-level	Pyramid
0	15.5 \pm 0.9		41.2 \pm 1.2	
1	31.4 \pm 1.2	32.8 \pm 1.3	55.9 \pm 0.9	57.0 \pm 0.8
2	47.2 \pm 1.1	49.3 \pm 1.4	63.6 \pm 0.9	64.6 \pm 0.8
3	52.2 \pm 0.8	54.0 \pm 1.1	60.3 \pm 0.9	64.6 \pm 0.7

Table 2. Classification results for the Caltech-101 database.

Recap: Spatial Pyramid BoW Classifier

- Features

1. Extract dense SIFT (spatially pooled and normalized histograms of gradients)
2. Assign each SIFT vector to a cluster number
3. Compute histograms of spatially pooled clustered SIFT vectors
 - Variations like Fisher vectors and 2nd order pooling shown to improve performance

- Classifier

- Linear SVM (or slightly better performance with chi-squared or histint SVMs)

Now let's go back to the core concepts of image categorization in general

Categorization involves **features** and a classifier

Training
Images



Training

Training
Labels

Image
Features

Classifier
Training

Trained
Classifier

Testing

Image
Features

Trained
Classifier

Prediction
Outdoor

Test Image



Q: What are good features for...

- recognizing a beach?



Q: What are good features for...

- recognizing cloth fabric?



Q: What are good features for...

- recognizing a mug?



What are the right features?

Depend on what you want to know!

- Object: shape
 - Local shape info, shading, shadows, texture
- Scene : geometric layout
 - linear perspective, gradients, line segments
- Material properties: albedo, feel, hardness
 - Color, texture
- Action: motion
 - Optical flow, tracked points

General principles of features

- Coverage
 - Ensure that all relevant info is captured
- Concision
 - Minimize number of features without sacrificing coverage
- Directness
 - Ideal features are independently useful for prediction

It's hard to design good features for a specific problem

Many machine learning algorithms solve for both feature and classifier parameters, such as

- Decision trees
- Random forests
- Multilayer neural networks (including CNNs)

Classifiers

Goal: From labeled training samples, learn parameters of a scoring/decision function that is likely to predict the correct label on test samples

Typical assumptions:

- Training and test samples drawn from same distribution (i.i.d.)
- Training labels are correct

Many classifiers to choose from

- SVM
- Neural networks
- Naïve Bayes
- Bayesian network
- Logistic regression
- Randomized Forests
- Boosted Decision Trees
- K-nearest neighbor
- RBMs
- Deep networks
- Etc.

Which is the best one?

Classifiers: three main options

- **Nearest neighbor:** take a vote from K closest neighbors
 - Can learn features or distance measure
- **Linear:** score is linear combination of features
 - SVM, perceptron, naïve bayes, logistic regression
 - Others learn features and then apply linear classifier (e.g. deep network, random forest)
- **Structured prediction:** score an interdependent set of labels
 - E.g. label body part positions
 - Structured SVM, CNN, graphical model algorithms

Generalization Theory

It's not enough to do well on the training set:
also should make good predictions for new
examples

No Free Lunch Theorem

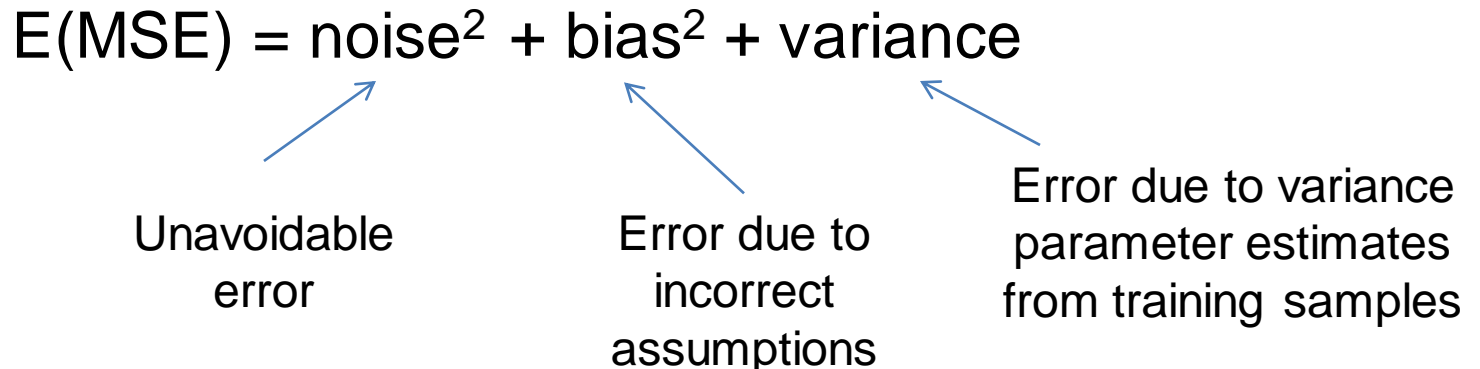
© Original Artist
Reproduction rights obtainable from
www.CartoonStock.com



Bias-Variance Trade-off

$$E(\text{MSE}) = \text{noise}^2 + \text{bias}^2 + \text{variance}$$

Unavoidable
error



Error due to
incorrect
assumptions

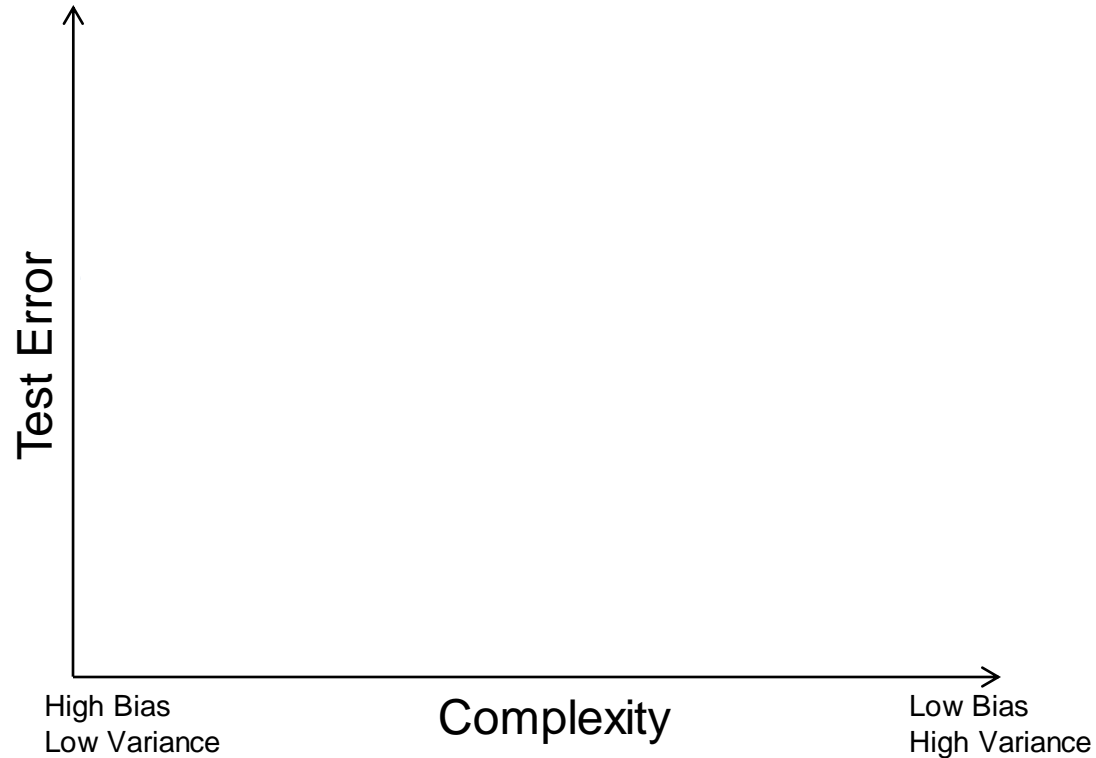
Error due to variance
parameter estimates
from training samples

See the following for explanation of bias-variance (also Bishop's "Neural Networks" book):

- <http://www.inf.ed.ac.uk/teaching/courses/mlsc/Notes/Lecture4/BiasVariance.pdf>

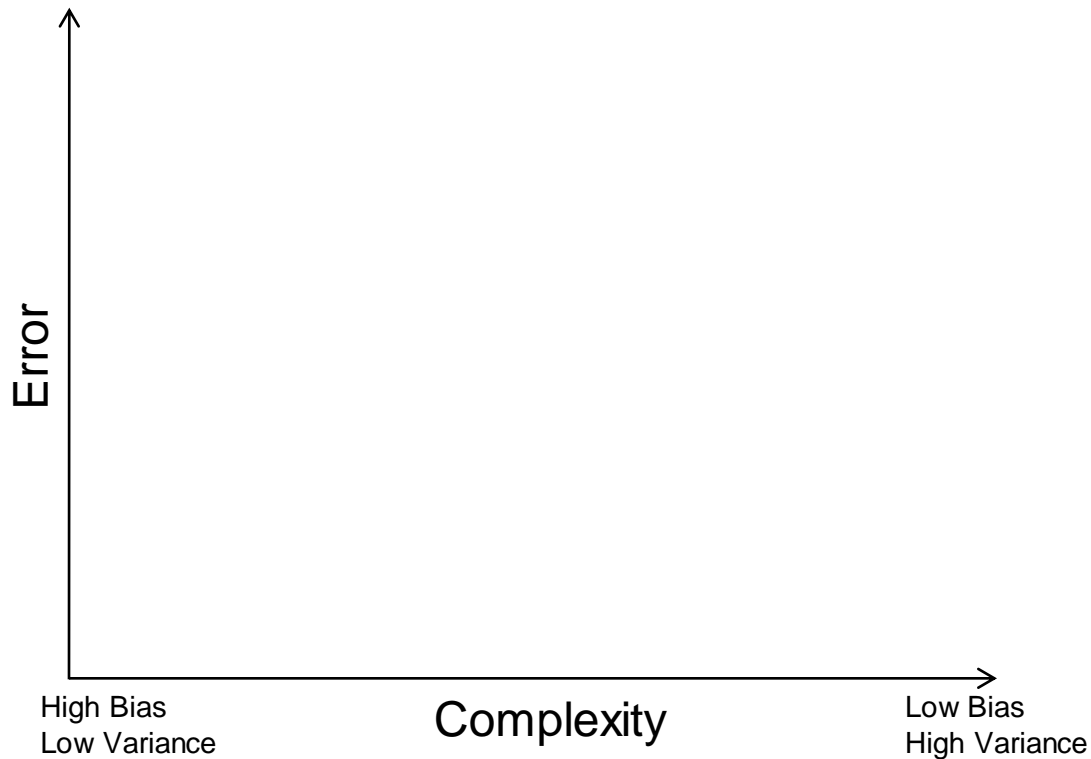
Bias and Variance

$$\text{Error} = \text{noise}^2 + \text{bias}^2 + \text{variance}$$



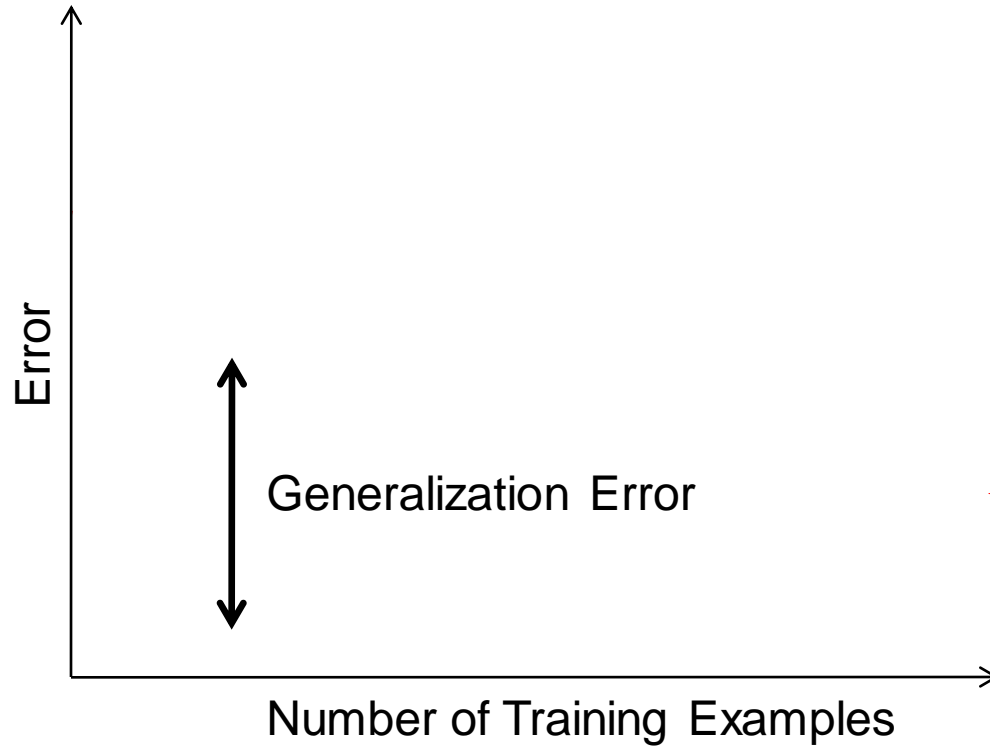
Choosing the trade-off

- Need validation set
- Validation set is separate from the test set



Effect of Training Size

Fixed classifier



How to measure complexity?

- VC dimension

What is the VC dimension of a linear classifier for N-dimensional features? For a nearest neighbor classifier?

Upper bound on generalization error

$$\text{Test error} \leq \text{Training error} + \sqrt{\frac{h(\log(2N/h) + 1) - \log(\eta/4)}{N}}$$

N: size of training set

h: VC dimension

η : 1-probability that bound holds

- Number of parameters

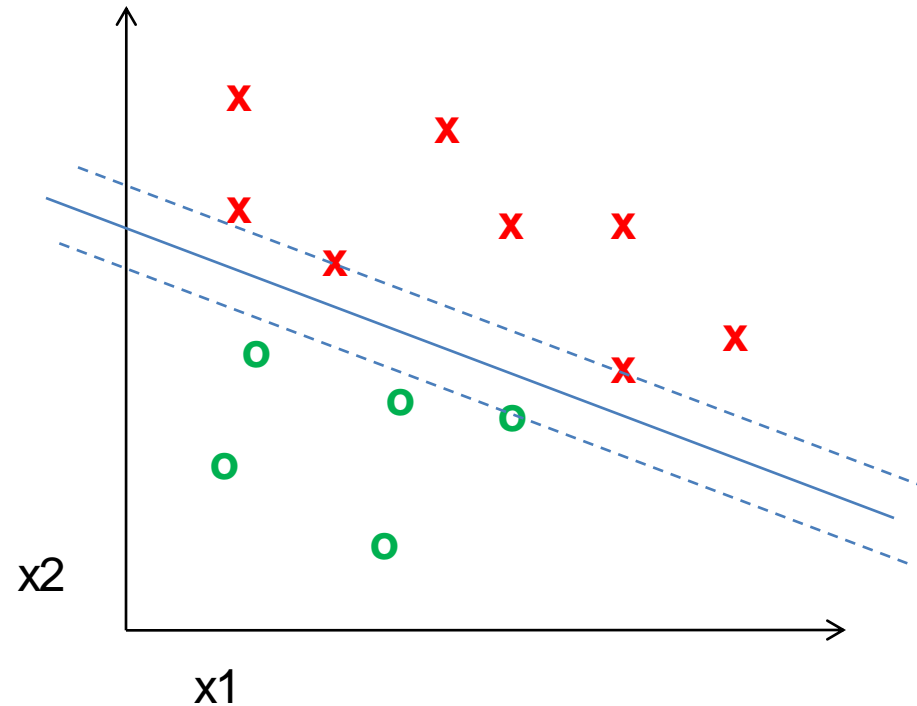
How to reduce variance?

- Choose a simpler classifier
- Regularize the parameters
- Use fewer features
- Get more training data

Which of these could actually lead to greater test error?

Maximizing Margin Reduces Risk of Error

- Key idea behind SVM



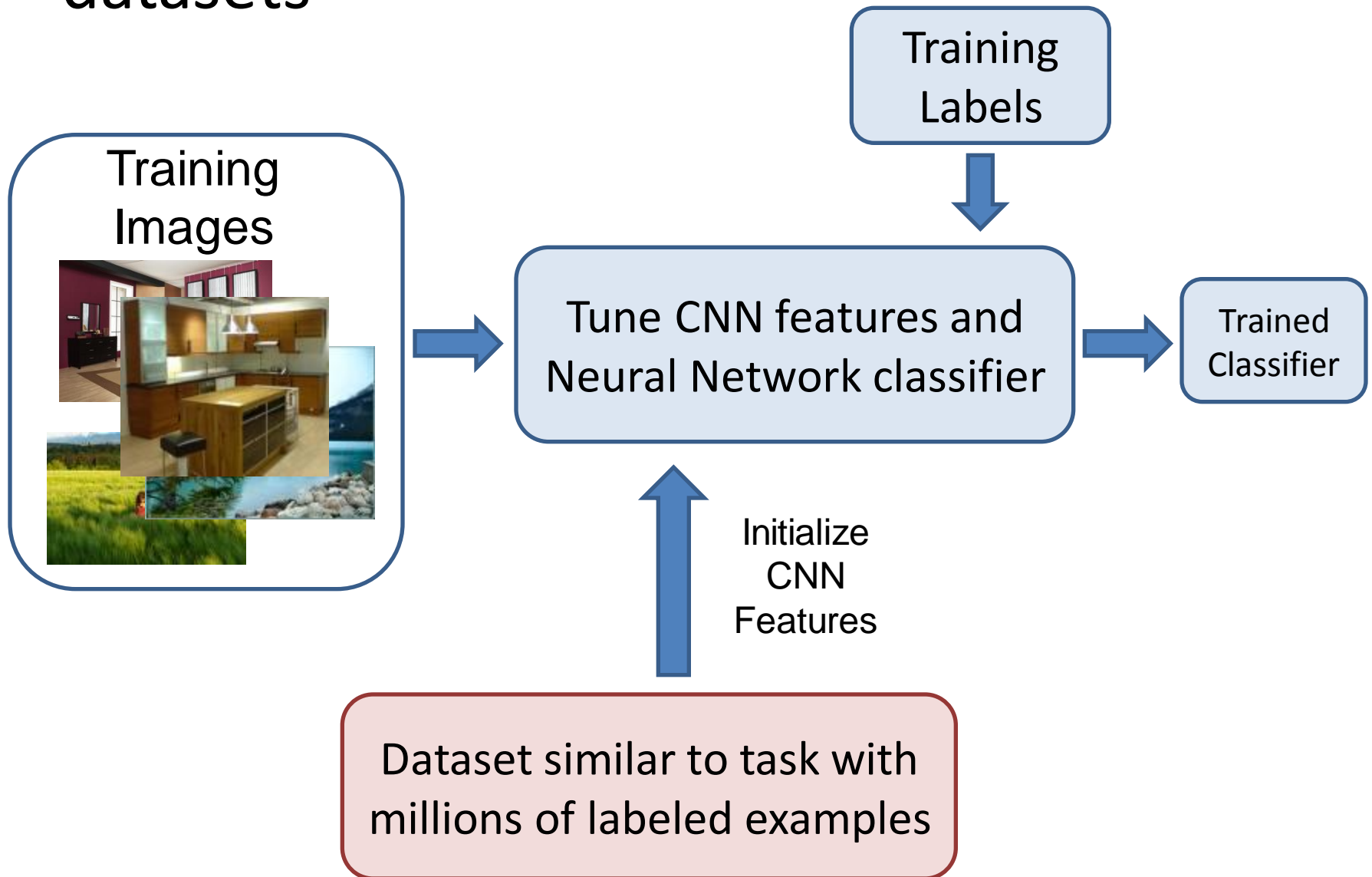
Characteristics of vision learning problems

- Lots of continuous features
 - E.g., HOG template may have 1000 features
 - Spatial pyramid may have ~15,000 features
- Imbalanced classes
 - often limited positive examples, practically infinite negative examples
- Difficult prediction tasks

When a massive training set is available

- Relatively new phenomenon
 - MNIST (handwritten letters) in 1990s, LabelMe in 2000s, ImageNet (object images) in 2009, ...
- Want classifiers with low bias (high variance ok) and reasonably efficient training
- Very complex classifiers with simple features are often effective
 - Random forests
 - Deep convolutional networks

New training setup with moderate sized datasets



Practical tips

- Preparing features for linear classifiers
 - Often helps to make zero-mean, unit-dev (whitening)
 - For non-ordinal features (e.g. cluster numbers), convert to a set of binary features
- Selecting classifier meta-parameters (e.g., regularization weight)
 - Cross-validation: split data into subsets; train on all but one subset, test on remaining; repeat holding out each subset
 - Leave-one-out, 5-fold, etc.
- Most popular classifiers in vision
 - *SVM*: linear for when fast training/classification is needed; performs well with lots of weak features
 - *Logistic Regression*: outputs a probability; easy to train and apply
 - *Nearest neighbor*: simple and hard to beat if there is tons of data (e.g., character recognition)
 - *Boosted stumps or decision trees*: applies to flexible features, incorporates feature selection, powerful classifiers
 - *Random forests*: outputs probability; good for simple features, tons of data
 - *Deep networks / CNNs*: flexible output; learns features; adapt existing network (which is trained with tons of data) or train new with tons of data

Next class

Deep convolutional neural networks (CNNs)

- Shock and awe
- Architecture (Alexnet, VGG, Inception, Resnet)
- Optimizer (stochastic gradient descent, Adam)
- Tips and tricks (batch-norm)
- Transfer
- Why it works