Intro to Machine Learning for Visual Computing

Dorothea Tanning, “Endgame”

CSC320: Introduction to Visual Computing

Slides from Derek Hoiem, Peter Barnum

Michael Guerzhoy

Examples of Categorization in Vision

- Part or object detection
  - E.g., for each window: face or non-face?
- Scene categorization
  - Indoor vs. outdoor, urban, forest, kitchen, etc.
- Action recognition
  - Picking up vs. sitting down vs. standing …
- Emotion recognition
- Region classification
  - Label pixels into different object/surface categories
- Boundary classification
  - Boundary vs. non-boundary
- Etc, etc.

Classification

- (Supervised) Machine Learning
  - Using a training set, make a function that correctly classifies new images
    - Example function that we already saw: return the name of the face that's the closes match in the training set to the input face

Image Categorization

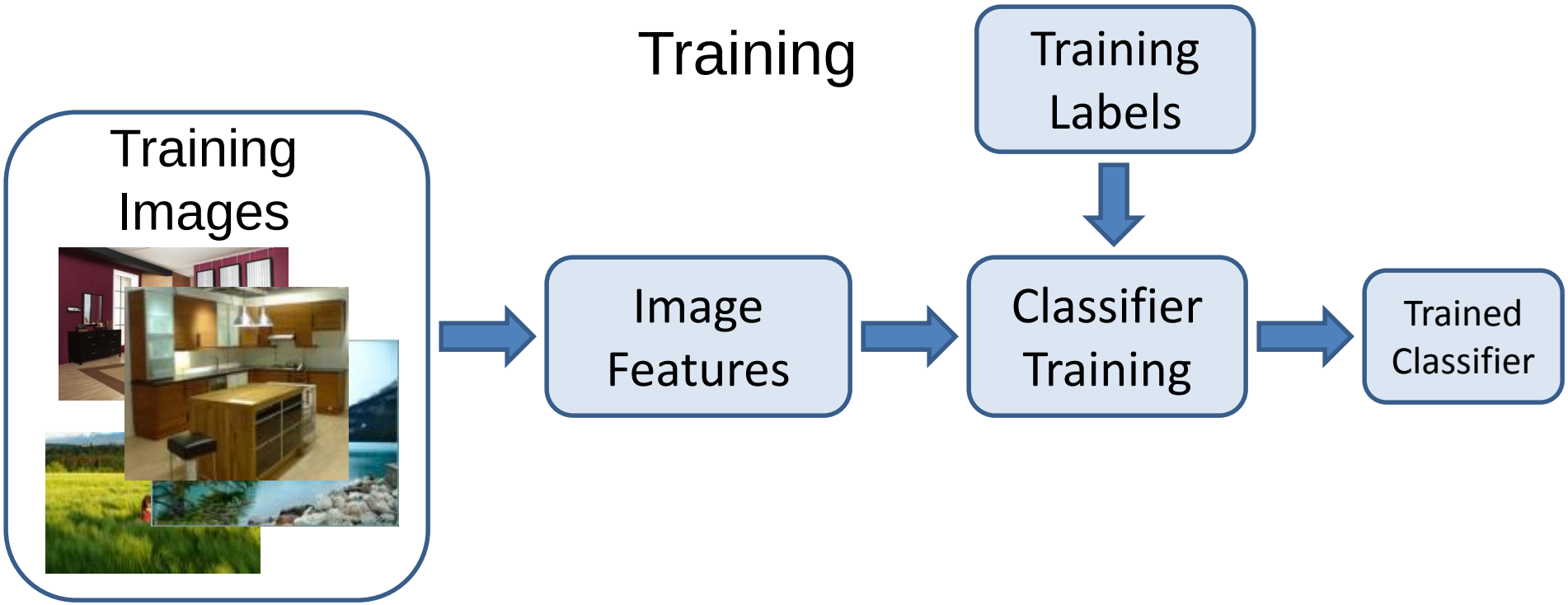
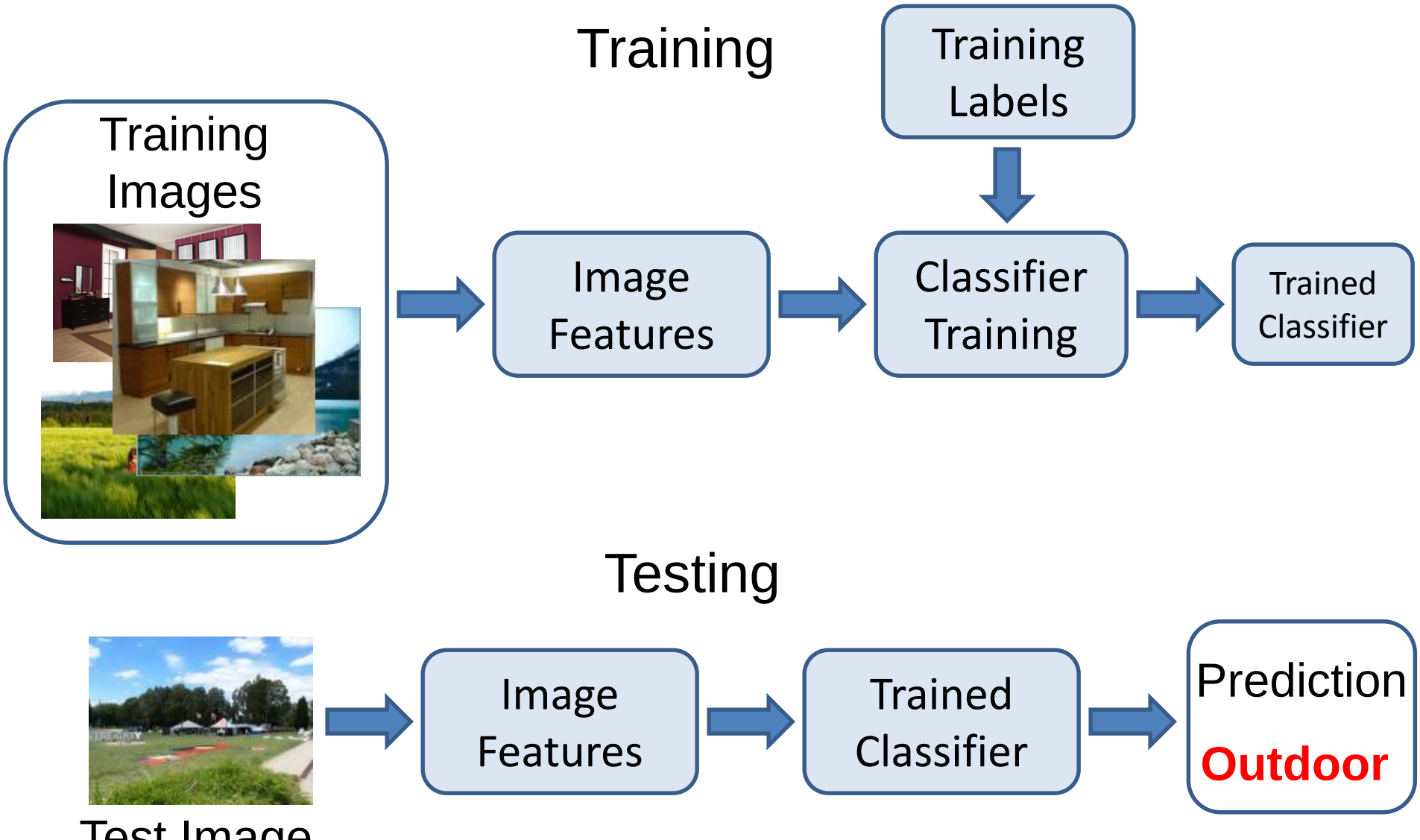


Image Categorization

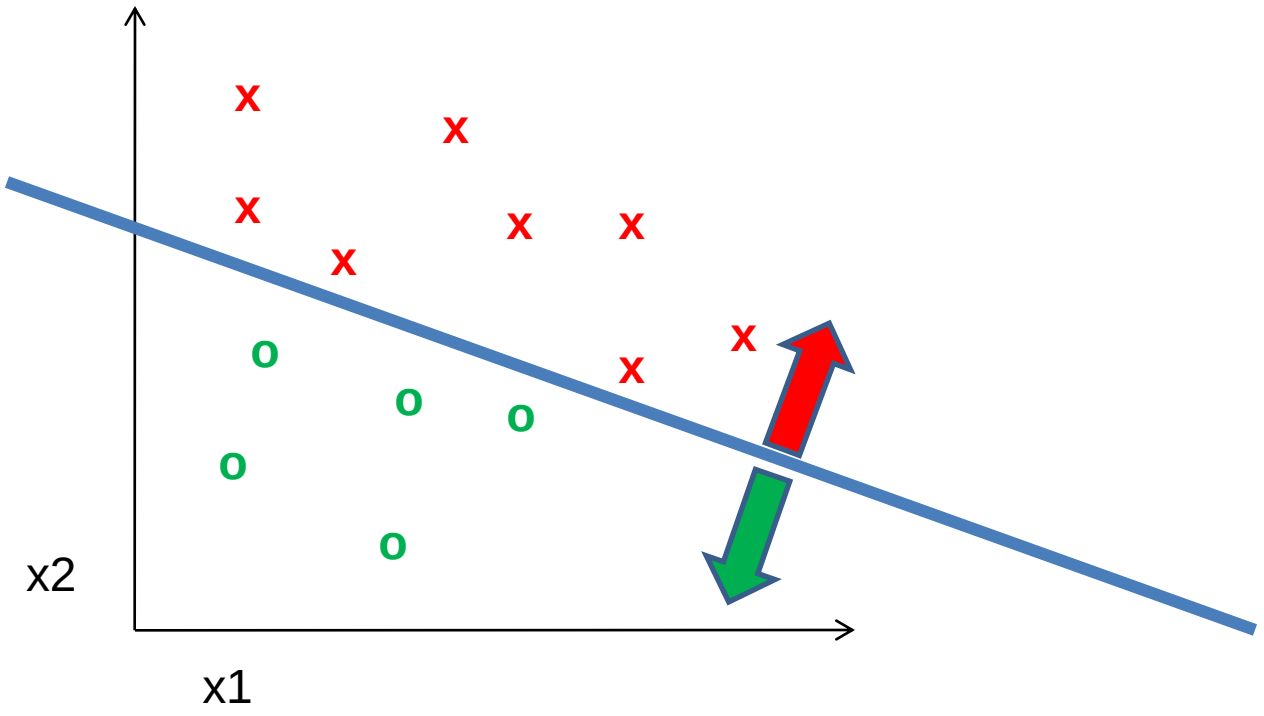


Feature design is paramount

- Most features can be thought of as templates, histograms (counts), or combinations
- Sample features:
  - The intensities of all the pixels in the image (simplest)
  - The average intensity in the image
  - The eigenspace projection coefficients
  - Histograms of Gradients (will talk about them later)

Classifier

A classifier maps from the feature space to a label



Different types of classification

- Exemplar-based: transfer category labels from examples with most similar features
  - What similarity function? What parameters?
- Linear classifier: confidence in positive label is a weighted sum of features
  - What are the weights?
- Non-linear classifier: predictions based on more complex function of features
  - What form does the classifier take? Parameters?
- Generative classifier: assign to the label that best explains the features (makes features most likely)
  - What is the probability function and its parameters?

Note: You can always fully design the classifier by hand, but usually this is too difficult. Typical solution: learn from training examples.

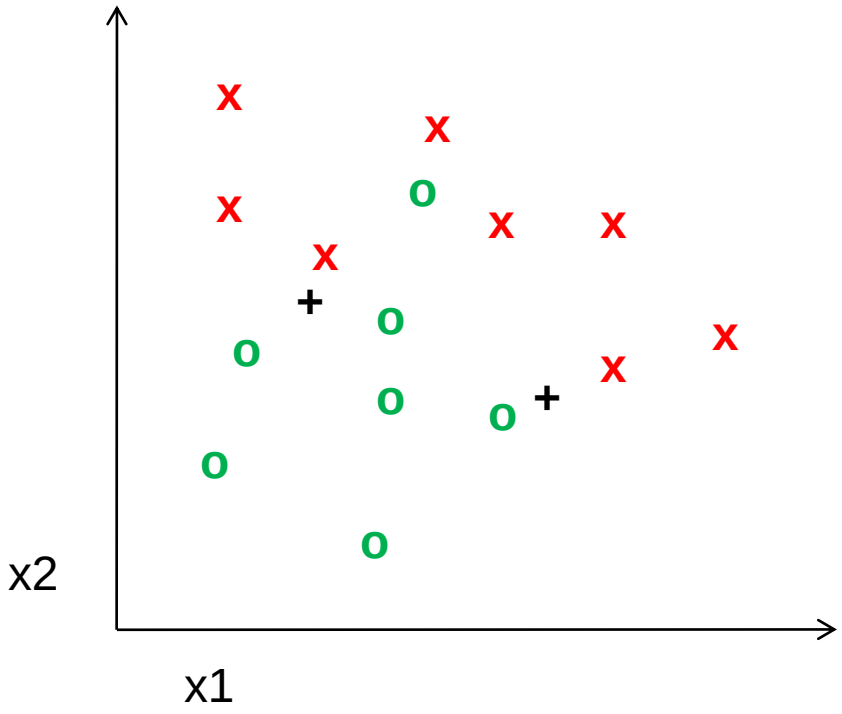
One way to think about it…

- Training labels dictate that two examples are the same or different, in some sense
- Features and distance measures define visual similarity
- Goal of training is to learn feature weights or distance measures so that visual similarity predicts label similarity
- *We want the simplest function that is confidently correct*

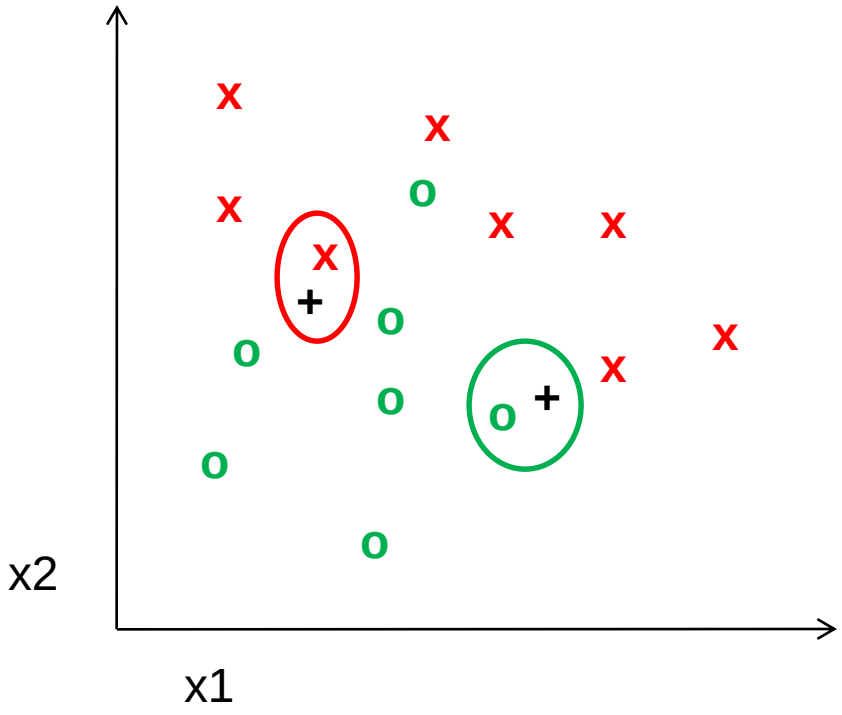
Exemplar-based Models

- Transfer the label(s) of the most similar training examples

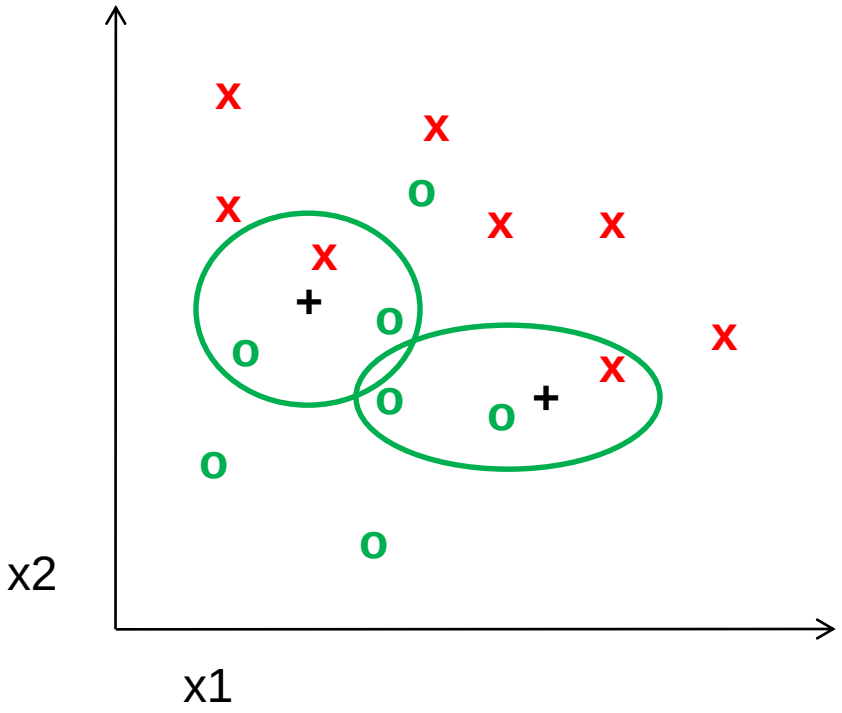
K-nearest neighbor classifier



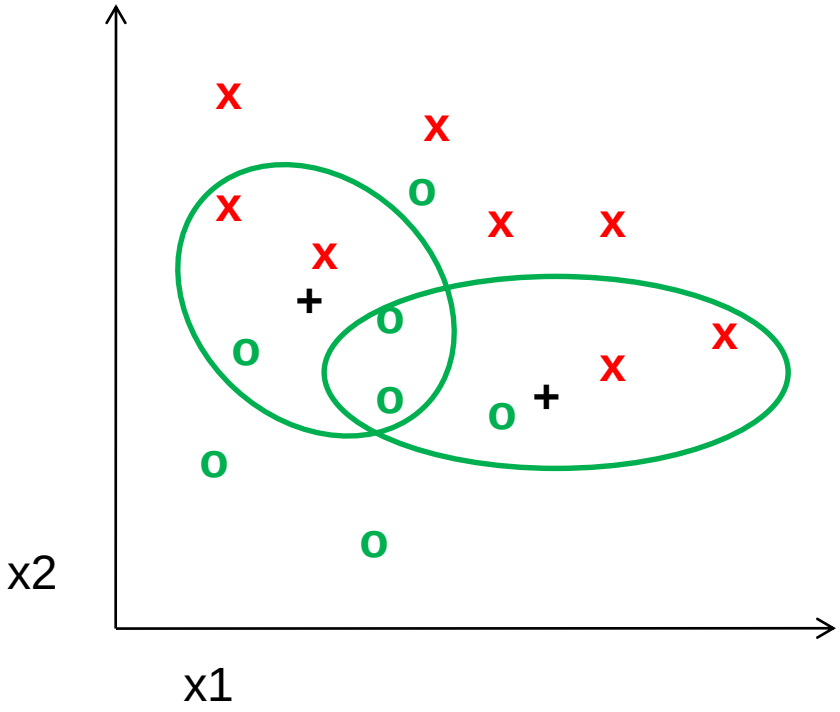
1-nearest neighbor



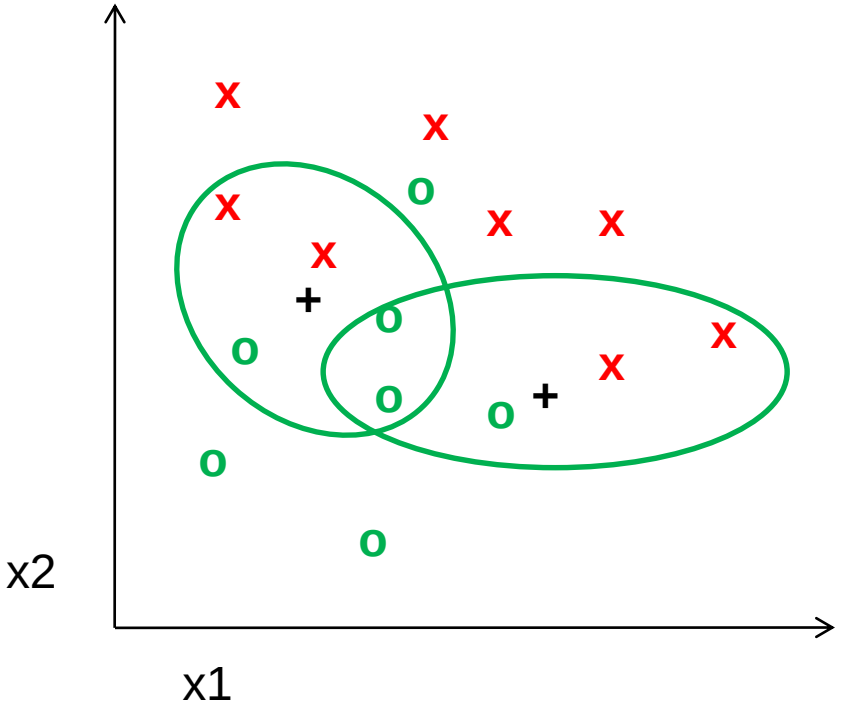
3-nearest neighbor



5-nearest neighbor



K-nearest neighbor



Using K-NN

- Simple, a good one to try first
- Higher K gives smoother functions
- No training time (unless you want to learn a distance function)
- With infinite examples, 1-NN provably has very low error

Discriminative classifiers

Learn a simple function of the input features that confidently predicts the true labels on the training set

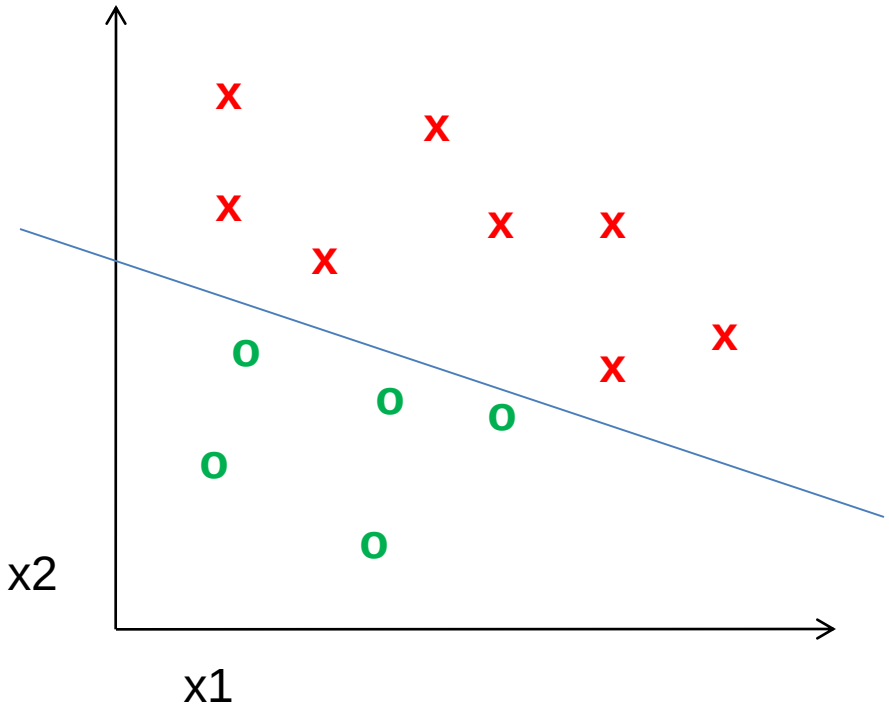
$$y = f(x)$$

Goals

1. Accurate classification of training data
2. Correct classifications are confident
3. Classification function is simple

Classifiers: Linear Regression

- x2 > ax1+b?

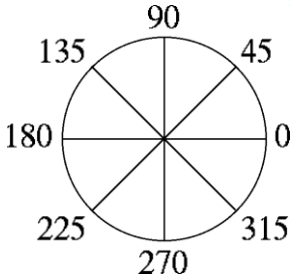


Other classifiers

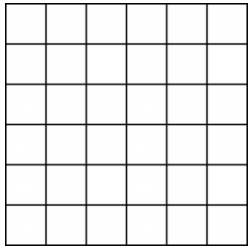
- Support Vector Machines (SVM)
  - Very popular
- Neural Networks
  - Always popular around UofT, currently the best classifiers around for many tasks
- Decision Trees
  - Popular “off-the-shelf” classifiers for new problems you encounter
- To some extent, all of those can be used as “black boxes”

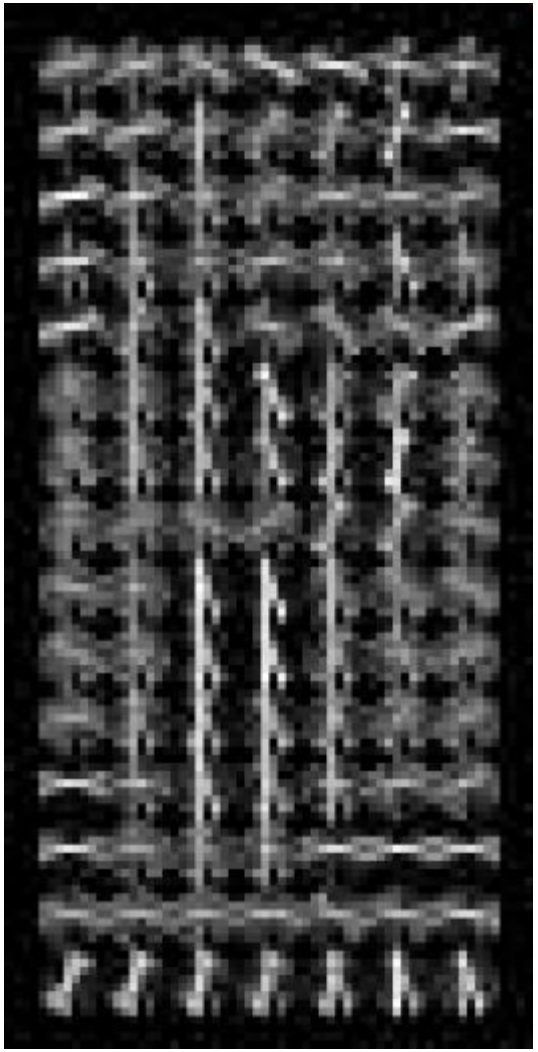
Example Image Feature: Histogram of Gradients (HoG)

Orientation: 9 bins
(for unsigned angles)



Histograms in
8x8 pixel cells

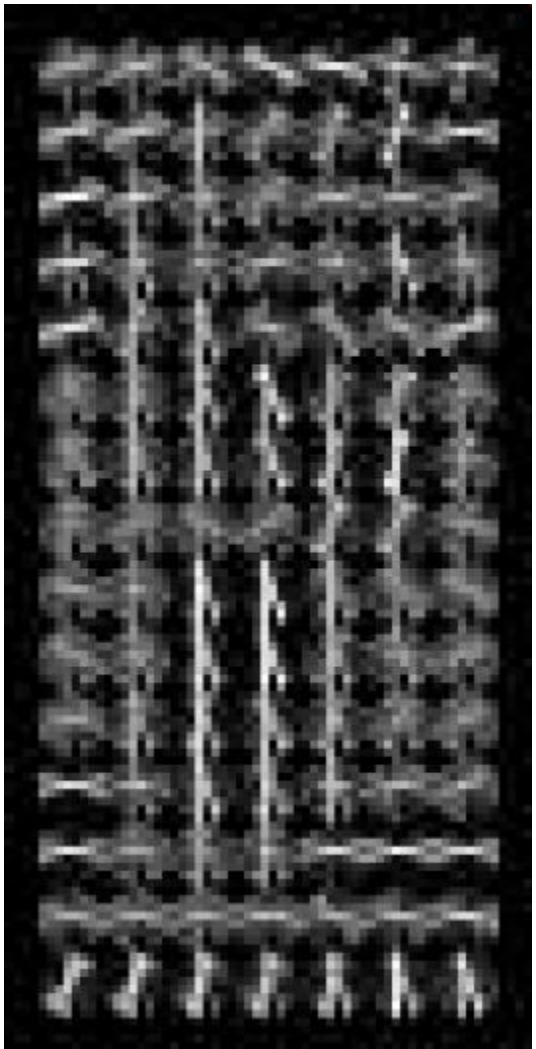




HoG

- Compute the HoG in each neighbourhood in the image, stack all the HoG together. Now classify those vectors

X=



features = 15 x 7 x 9 x 4 = 3780

orientations

cells

normalizations by neighboring cells

- (Find the w and b with Machine Learning magic)

$$0.16 = w^T x - b$$

$$sign(0.16) = 1$$

$$=>$$ pedestrian

Histograms of Oriented Gradients for Human Detection

Navneet Dalal and Bill Triggs
INRIA Rhône-Alps, 655 avenue de l'Europe, Montbonnot 38334, France
{Navneet.Dalal,Bill.Triggs}@inrialpes.fr, http://lear.inrialpes.fr

Abstract

We study the question of feature sets for robust visual object recognition, adopting linear SVM based human detection as a test case. After reviewing existing edge and gradient based descriptors, we show experimentally that grids of Histograms of Oriented Gradient (HOG) descriptors significantly outperform existing feature sets for human detection. We study the influence of each stage of the computation on performance, concluding that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping descriptor blocks are all important for good results. The new approach gives near-perfect separation on the original MIT

We briefly discuss previous work on human detection in §2, give an overview of our method §3, describe our data sets in §4 and give a detailed description and experimental evaluation of each stage of the process in §5–6. The main conclusions are summarized in §7.

2 Previous Work

There is an extensive literature on object detection, but here we mention just a few relevant papers on human detection [18, 17, 22, 16, 20]. See [6] for a survey. Papageorgiou *et al* [18] describe a pedestrian detector based on a polynomial SVM using rectified Haar wavelets as input descriptors, with a parts (subwindow) based variant in [17]. Depoortere *et al*

Computer Vision and Pattern Recognition, (CVPR) 2005