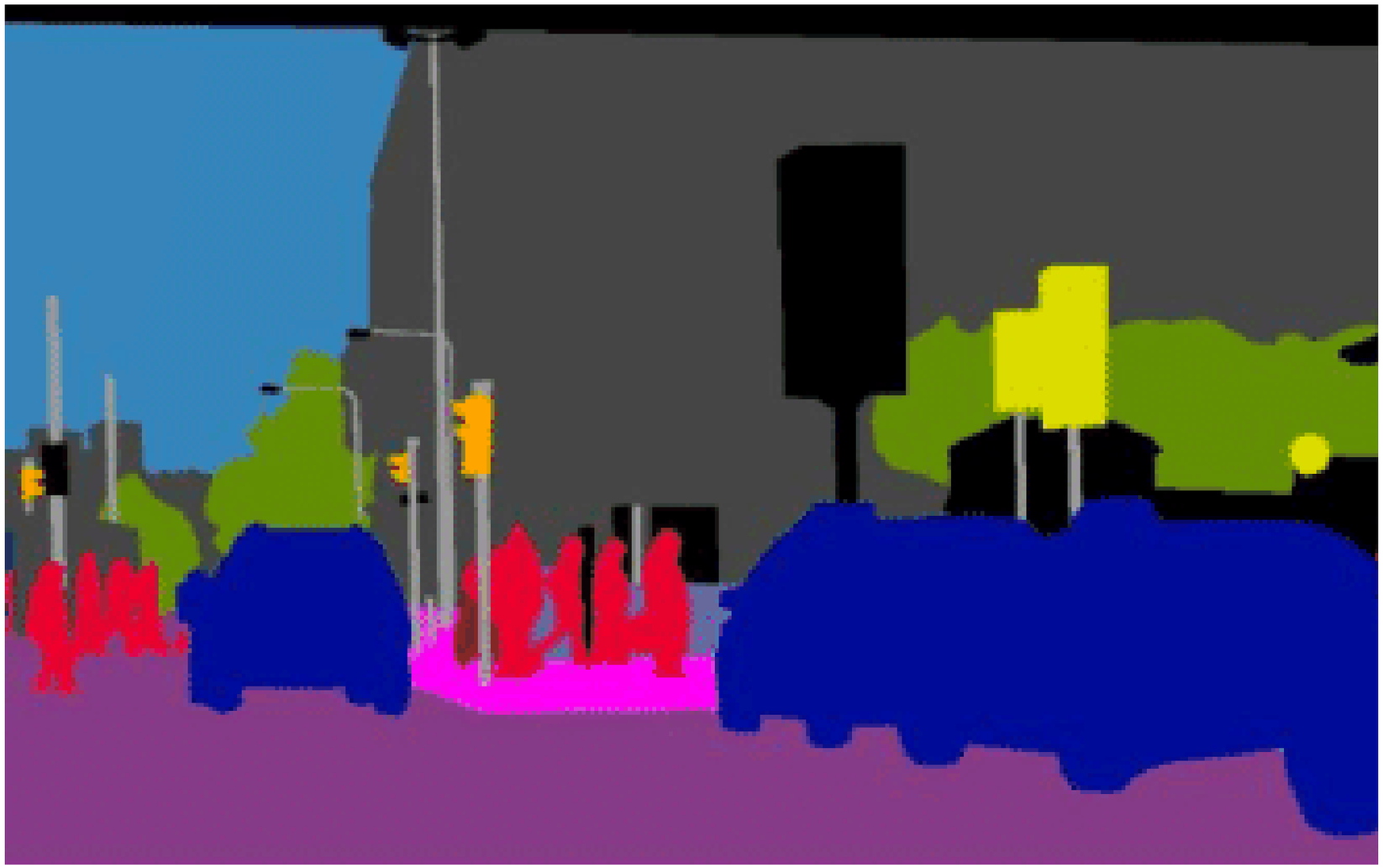
Recognition



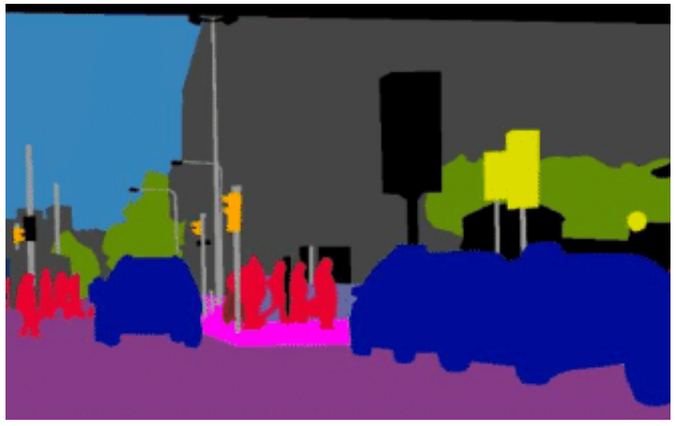
https://cmu-ci-lab.github.io/16705/

16-705 Computer Vision
Spring 2026, Lecture 6

Course announcements

- Homework assignment 1 posted on Canvas, due on April 10.
- Office hours will be posted after class, will be Monday, Wednesday, Friday.

Recognition



- **How to define problem?**
  - Classification / VQA / (panoptic) segmentation
  - Dataset (bias)
  - Metrics / losses
- Case example: finding people
- Statistical classification

Formalizing recognition

Output

What are outputs of interest?
How should space of outputs be represented?

Classic formulations

Image classification

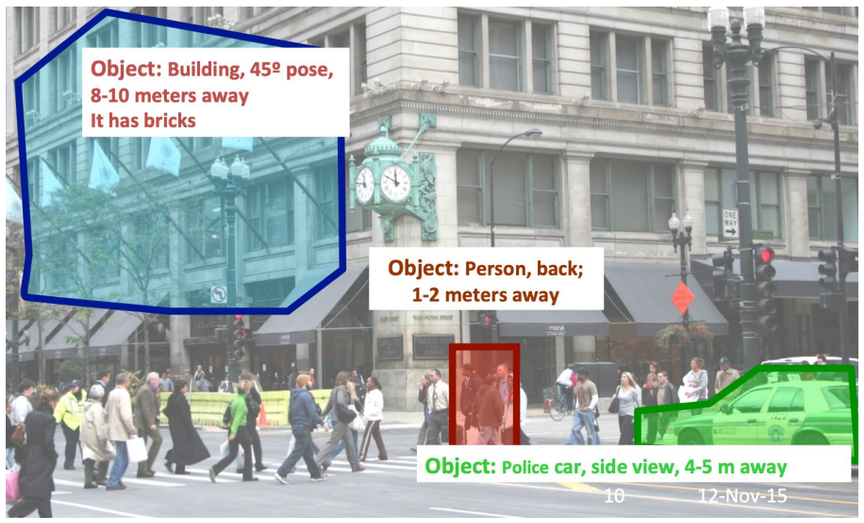
is this a city or beach?

Detection/segmentation

Does this contain a car? Where?

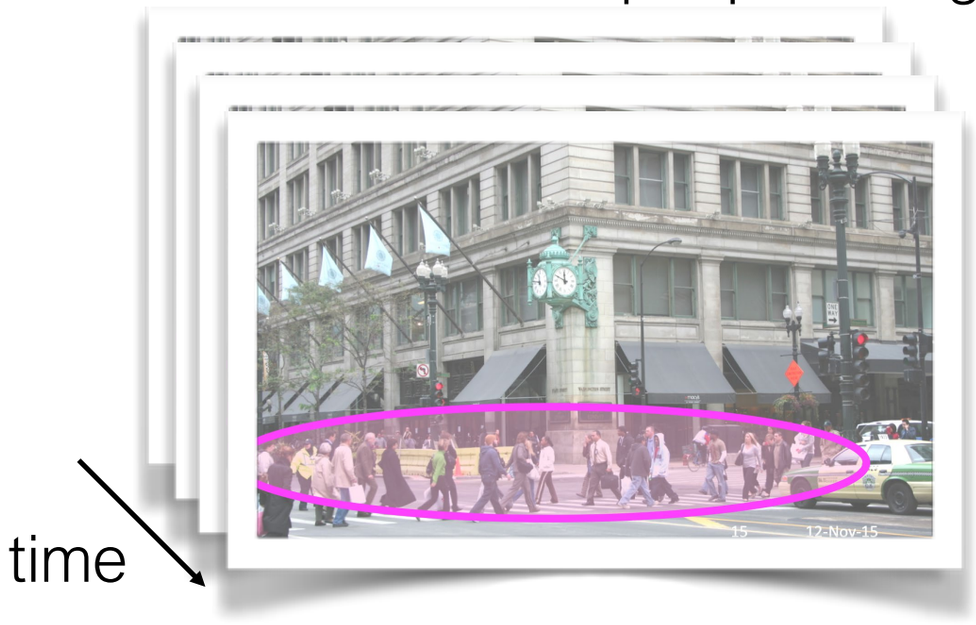
Attributes *(open-ended)*

Is this object furry? Brown?



Activities

What are these people doing?



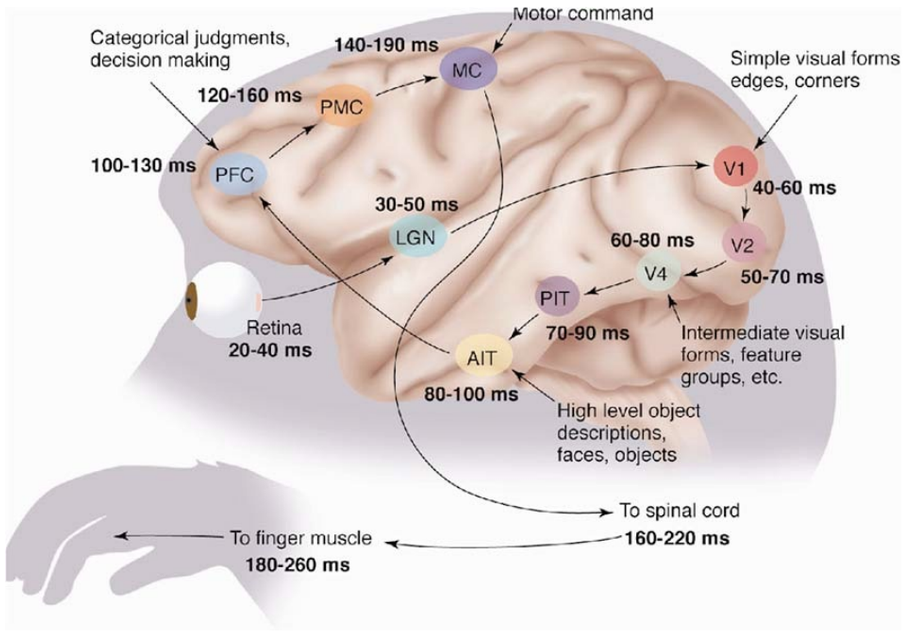
Formalizing recognition

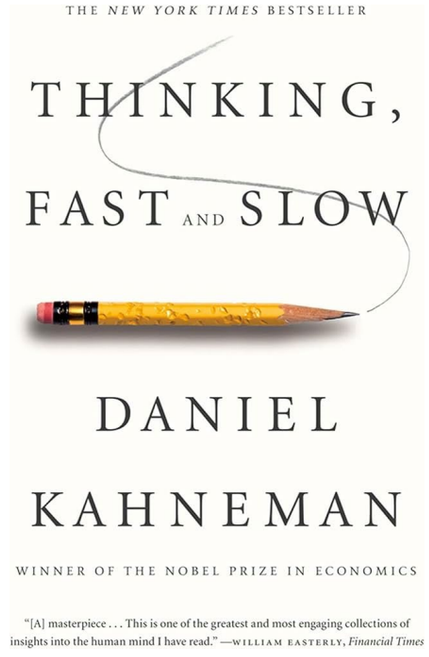
Human vision experiment: what can people describe when looking at images?

Tricky question to answer….

Rapid scene catgorization

People can distinguish high-level concepts (animal/transport) in under 150ms (Thorpe)



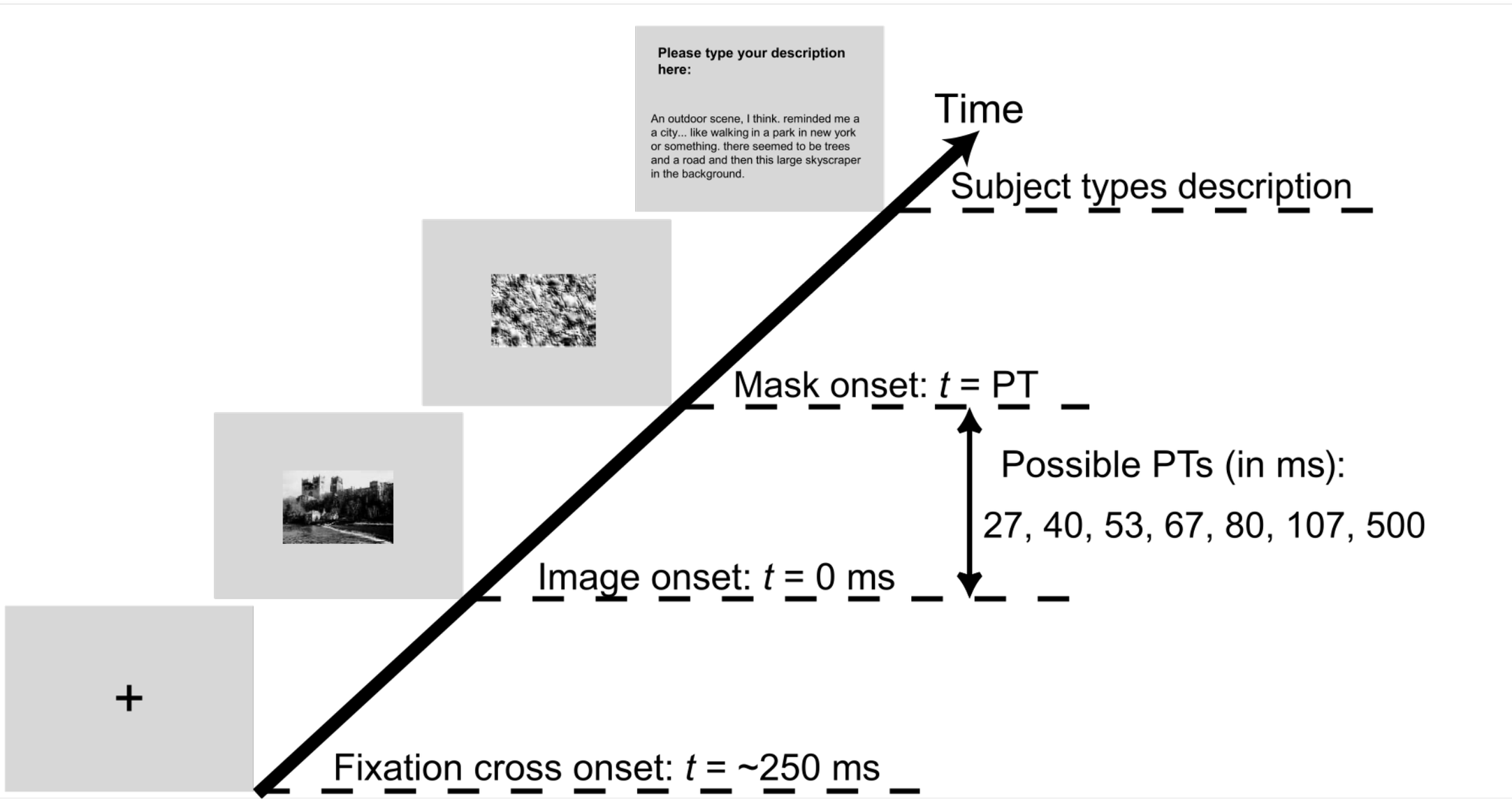


System 1 (fast)
vs system 2 (slow)

Appears to suggest feed-forward computations suffice (or at least dominate)

Journal of Vision (2007) 7(1):10, 1–29 http://journalofvision.org/7/1/10/

What do we perceive in a glance of a real-world scene?



Journal of Vision (2007) 7(1):10, 1–29 http://journalofvision.org/7/1/10/ 1

What do we perceive in a glance of a real-world scene?

PT = 107 ms

This is outdoors. A black, furry dog is running/walking towards the right of the picture. His tail is in the air and his mouth is open. Either he had a ball in his mouth or he was chasing after a ball. (Subject EC)

PT = 500 ms

I saw a black dog carrying a gray frisbee in the center of the photograph. The dog was walking near the ocean, with waves lapping up on the shore. It seemed to be a gray day out. (Subject JB)

Inside a house, like a living room, with chairs and sofas and tables, no ppl. (Subject HS)

A room full of musical instruments. A piano in the foreground, a harp behind that, a guitar hanging on the wall (to the right). It looked like there was also a window behind the harp, and perhaps a bookcase on the left. (Subject RW)

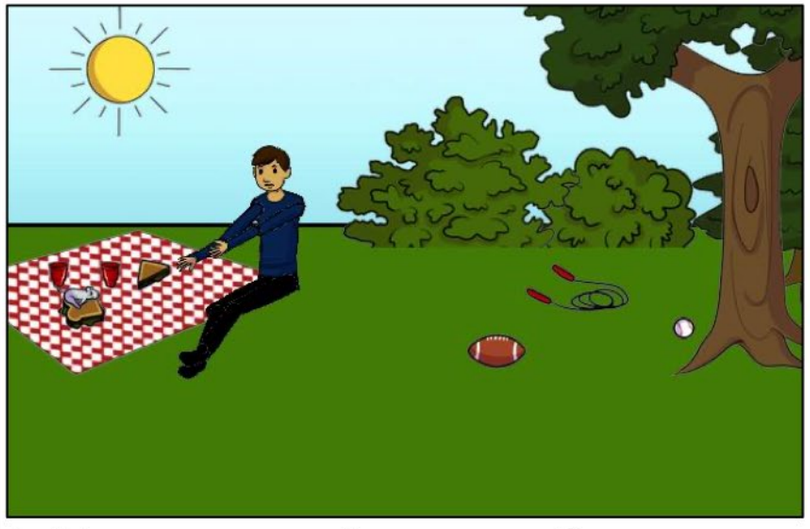
Should language be the right output?

Visual Question Answering (VQA)

Currently popular due to vision-language chatbots

What color are her eyes?
What is the mustache made of?

How many slices of pizza are there?
Is this a vegetarian pizza?



Is this person expecting company?
What is just under the tree?

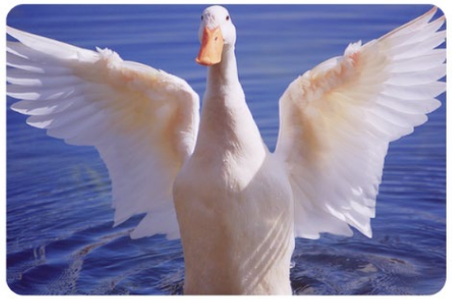
Does it appear to be rainy?
Does this person have 20/20 vision?

Generate questions via Amazon MTurk:

> *"We have built a smart robot. It understands a lot about images. It can recognize and name all the objects, it knows where the objects are, it can recognize the scene (*e.g.*, kitchen, beach), people's expressions and poses, and properties of objects (*e.g.*, color of objects, their texture). Your task is to stump this smart robot!"*.

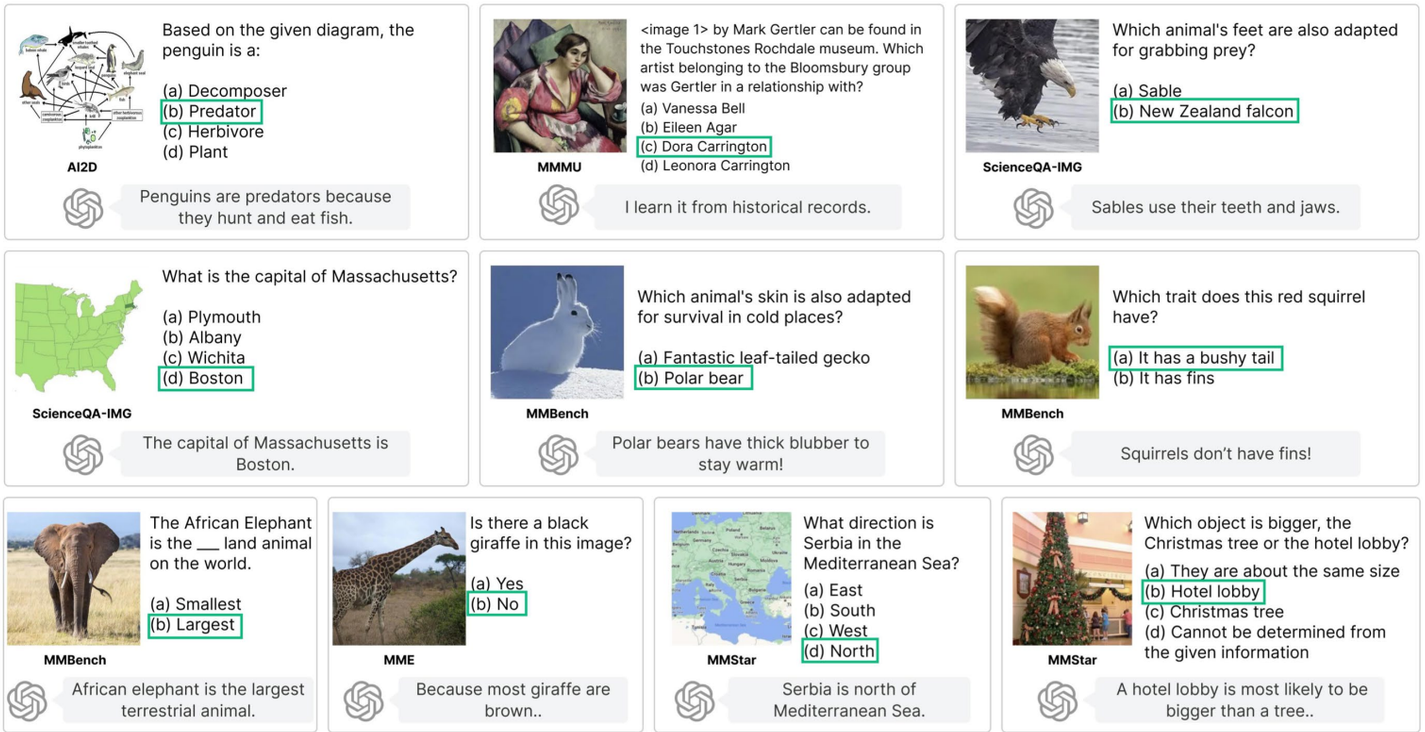
Claim: it’s hard to construct good VQA benchmarks

ARO (2022), Science-QA-IMG(2022), MMBench(2023), MME (2023), MMStar(2024), AI2D (2024), MMMU(2024)



t_1= a white duck spreads its wings while in the water
t_2= a white wings spreads its water while in the duck
t_3= a white duck the its wings while in water spreads
t_4= white a duck spreads its wings in while the water
t_5= while in the spreads its wings water a white duck

ARO (2022)
Designed to stress-test “brown dog on a black street” vs “black dog on brown street”



Blind language models (that don’t look at visual image) can solve many vision-language benchmarks (seemingly forgotten lesson from 2016 *VQA2: Making the V in VQA Matter*)

Turns out that language models are biased for “a,b,c,d” and “yes/no” questions. Guess what’s the bias?

My favorite vision-language benchmark: Winoground (2022)

some plants surrounding a light bulb

a lightbulb surrounding some plants

Goal: identify the correct (image,caption) pair given the 4 candidate combos
(im1,text1) (im1,text2) (im2,text1) (im2,text2)

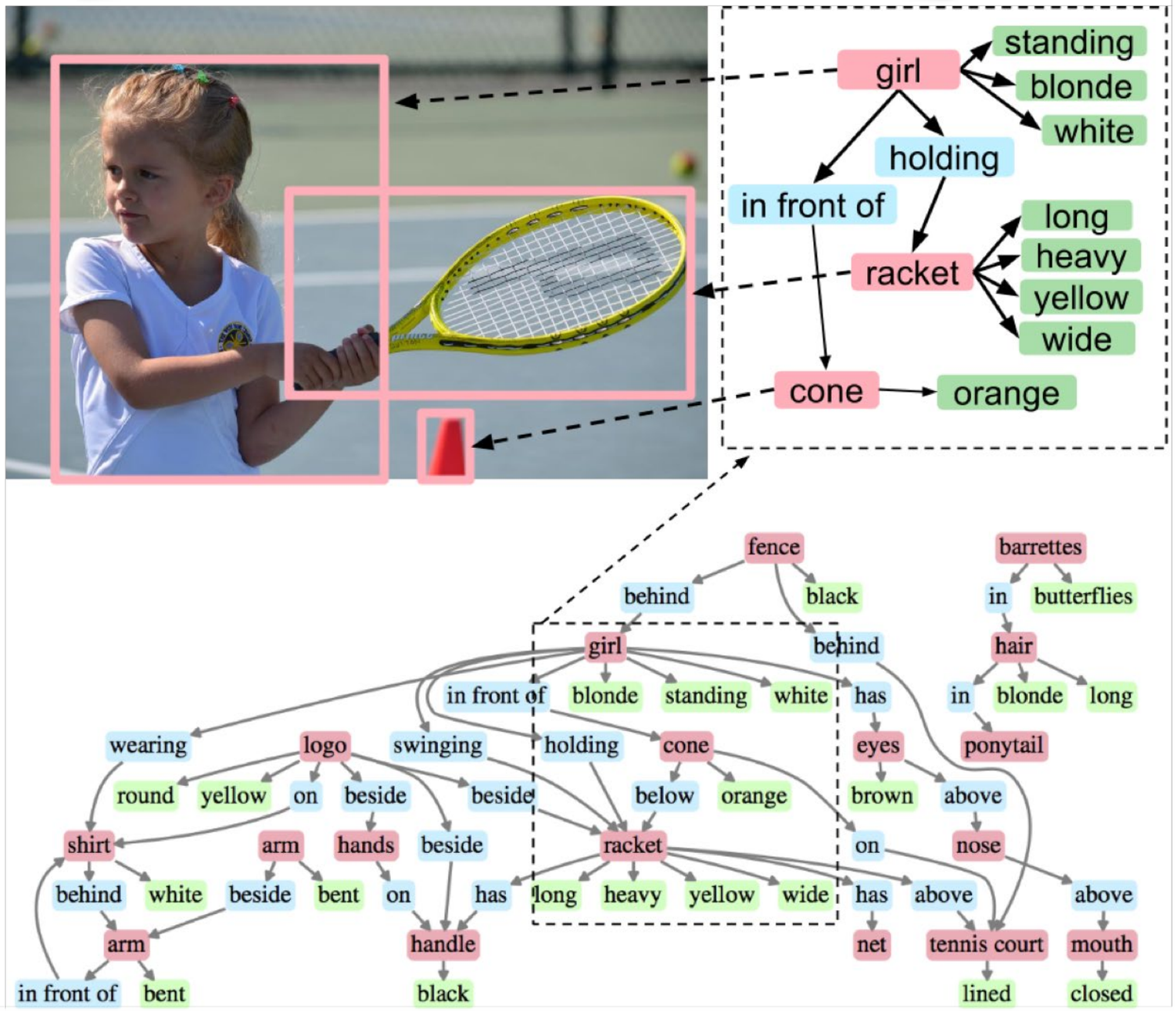
It’s hard to collect meaningful out-of-context examples

My hunch: we need output representations that are more structured (e.g., graph)

Image Retrieval using Scene Graphs

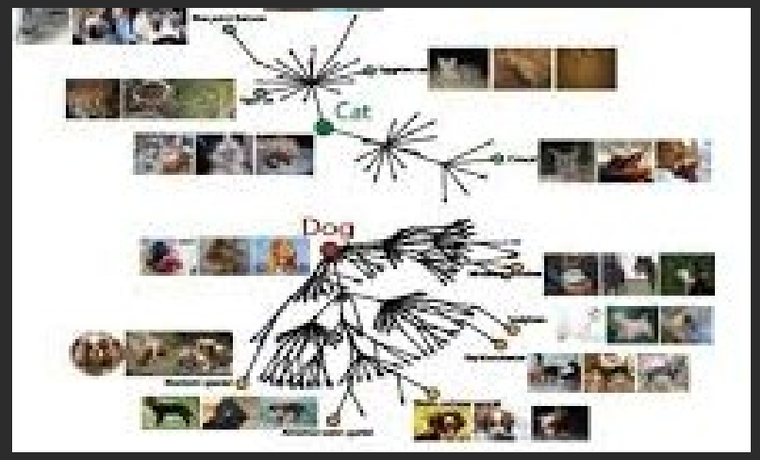
Justin Johnson[1], Ranjay Krishna[1], Michael Stark[2], Li-Jia Li[3,4], David A. Shamma[3], Michael S. Bernstein[1], Li Fei-Fei[1]

Stanford University, [2]Max Planck Institute for Informatics, [3]Yahoo Labs, [4]Snapchat



Other relational structures for organizing recognition outputs
(from the feild of *ontology*, c.f. knowledge representation and philosophy)

Taxonomies *"is a"*
(dogs and cats are types of mammals)

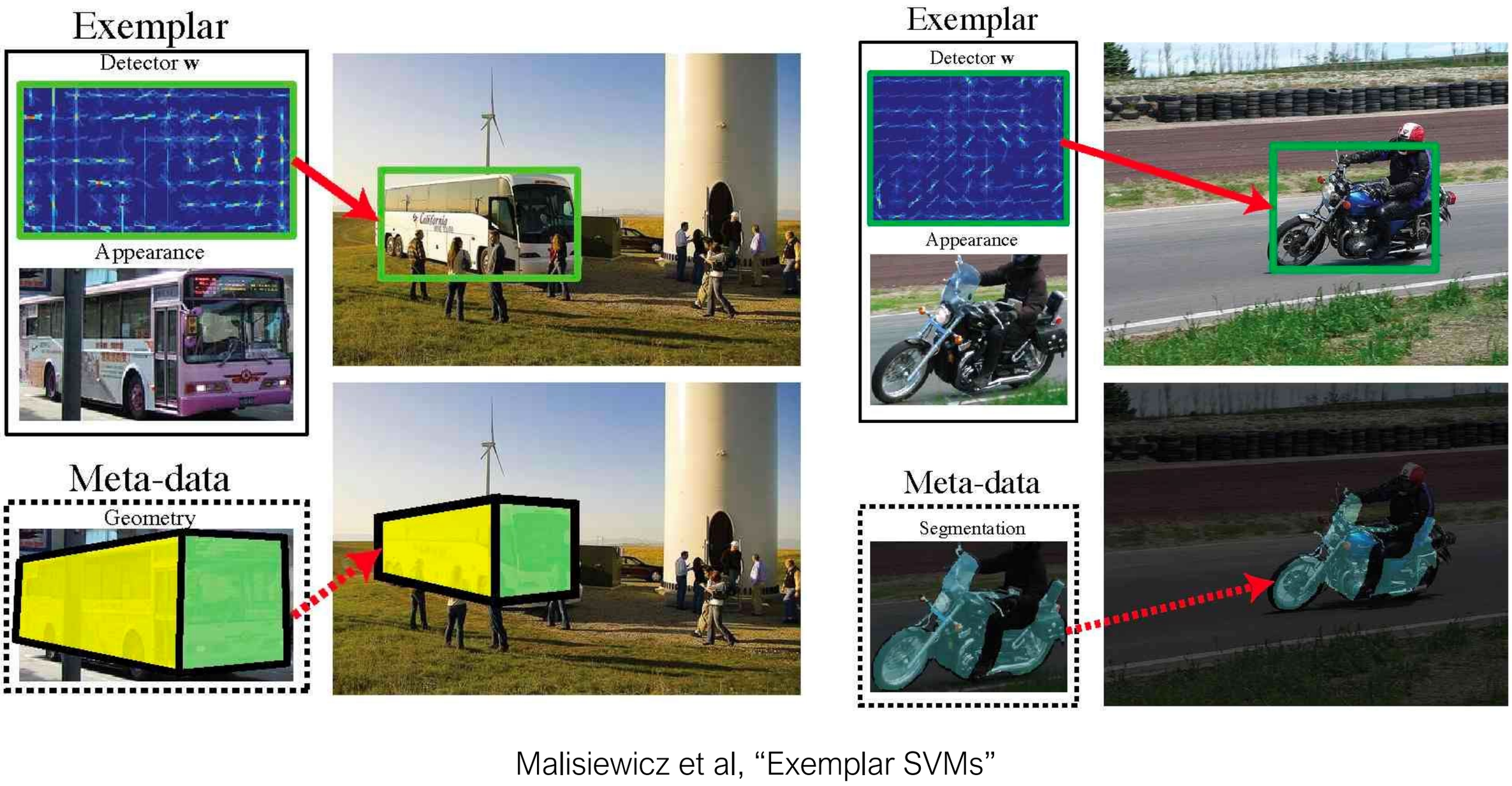


Partonomies / Meronomies *"has a"*
(people are made out of arms and legs)

Spatiotemporal Relations
(people stand on floors)

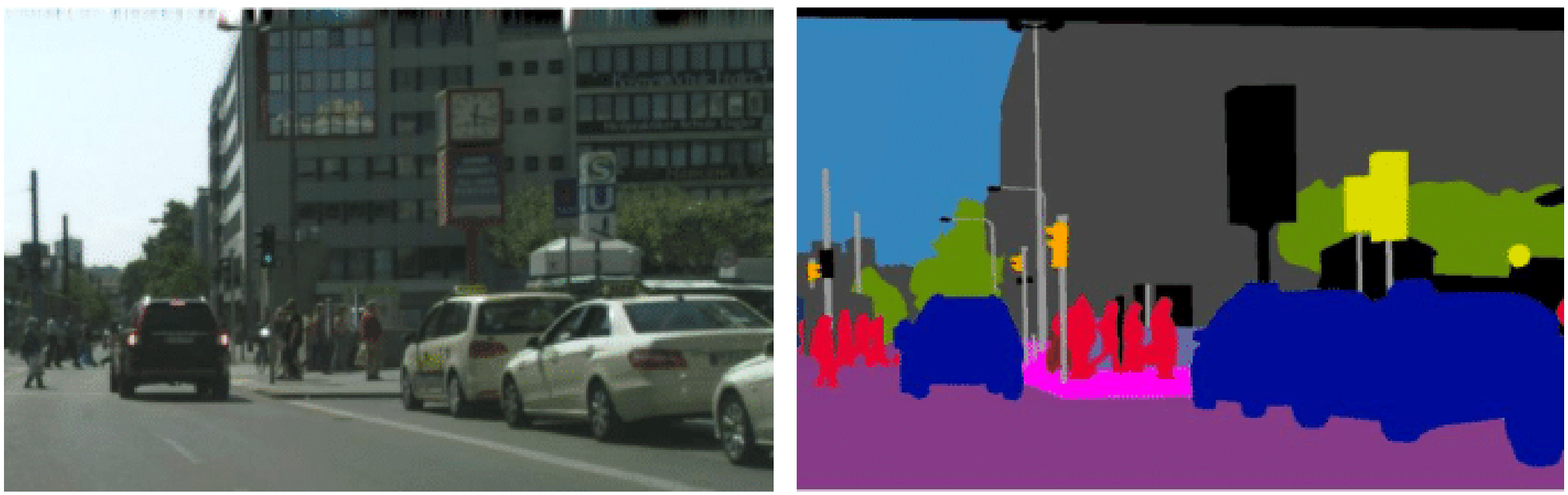
Alternate approach to knowledge represenation: *associative memory*

Big-data philoshopy: ask not “what is this”, but “what is this like”?



Malisiewicz et al, “Exemplar SVMs”

My (current) favorite answer: panoptic image segmentation



Question: We can represent outspace space of semantic labels for pixel i as y[i] in {0,1,…L}.
How about output space of panoptic labels?

One option: predict N^2 binary co-occurence matrix (does pixel i cluster with pixel j?}

Let’s give it a try: let’s ***manually*** annotate all pixels in this image!

Notes on image annotation

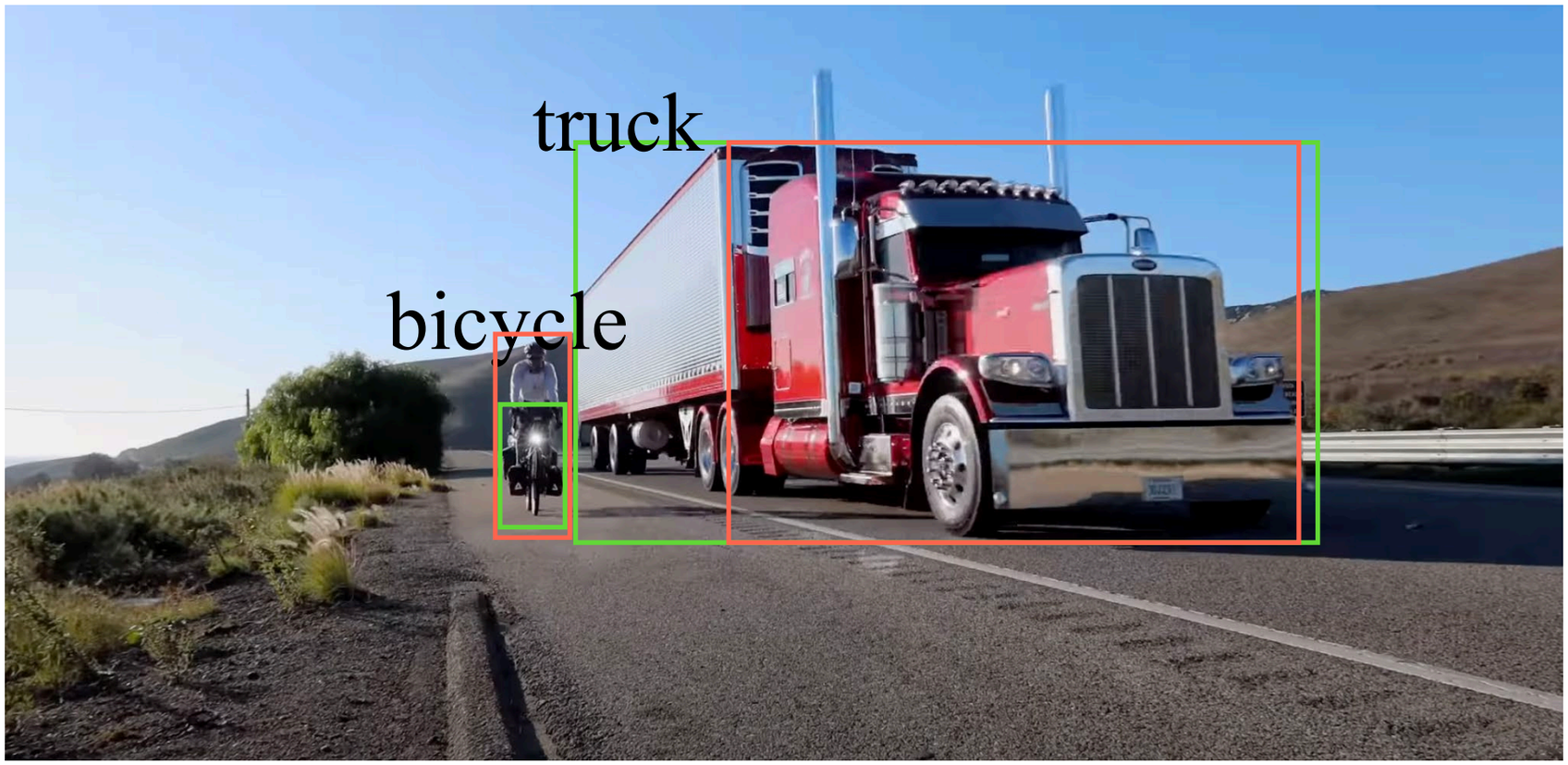
Adela Barriuso, Antonio Torralba
Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology

“ I can see the ceiling, a wall and a ladder, but I do not know how to annotate what is on the right side of the picture. Maybe I just need to admit that I can not solve this picture in an easy and fast way. But if I was forced ”

Semantic blindspots

The difficulty of defining a *closed-world* set of categories…

Let's try running GPT on an image (NuScenes)

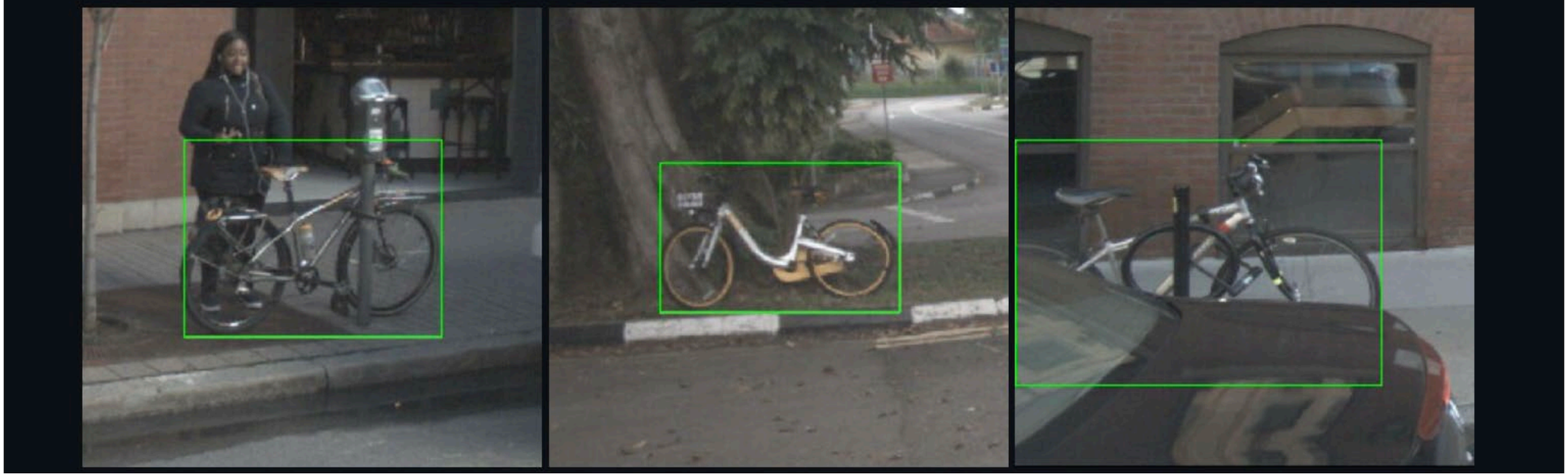


Turns out off-the-shelf predictions don't match the ground-truth (flagship autonomous vehicle dataset). Why the mismatch?

NuScenes labelling instructions

Bicycle

- Human or electric powered 2-wheeled vehicle designed to travel at lower speeds either on road surface, sidewalks or bicycle paths.
  - If there is a rider, include the rider in the box
  - If there is a passenger, include the passenger in the box
  - If there is a pedestrian standing next to the bicycle, do NOT include in the annotation



https://github.com/nutonomy/nuscenes-devkit/blob/master/docs/instructions_nuscenes.md

… which are different than Waymo instructions

Cyclist Labeling Specifications

What is labeled

- All objects that can be recognized as a cyclist and are at least partially visible are labeled.
- If it is not possible to tell by looking at the camera image whether an object is a cyclist, the object is not labeled.
- Bicycles that are parked or do not have a rider are not labeled.
- When a pedestrian is getting onto a bicycle, they are labeled as pedestrian until they are about to get onto the bicycle, and labeled as cyclist after the rider gets into the riding position. Similarly, when a pedestrian is getting off of a bicycle, they are labeled as cyclist while the rider is in the riding position, and labeled as pedestrian once they start getting off the bicycle.
- Labels are created for:
  - pedestrians riding bicycles, unicycles, tricycles, recumbent bicycles, and large, multi-seat cycles
  - children riding bicycles, tricycles or toy with wheels

https://github.com/waymo-research/waymo-open-dataset/blob/master/docs/labeling_specifications.md

… which can lead to major confusion about safety!

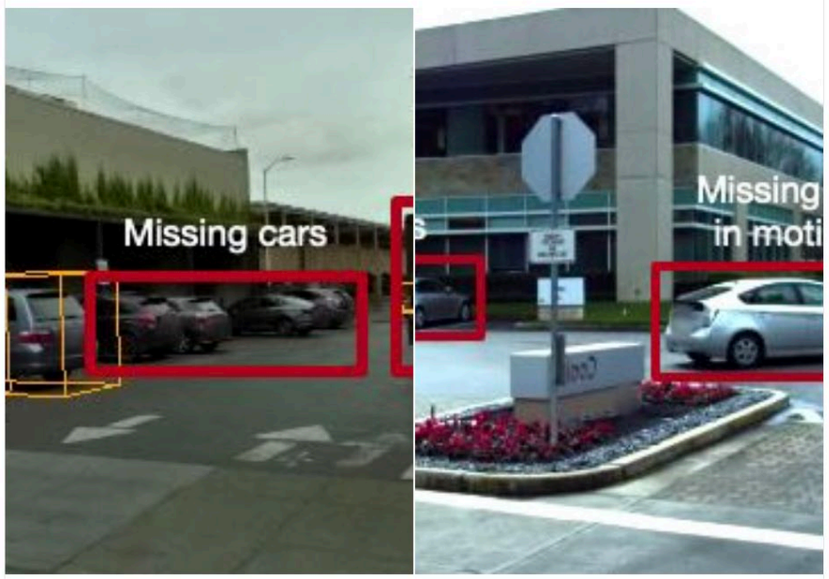
Daniel Kang's Post

Daniel Kang
Assistant professor at UIUC CS
2y · Edited

ML models are increasingly being deployed in mission-critical settings, such as autonomous vehicles, but shockingly the data used to train these models are rarely checked! For example, the Lyft Level 5 dataset has errors in 70% of the validation scenes. Bad data can lead to bad models! Related work shows that bad data can effectively reduce model capacity by 3x (Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks)! See our blog post for more details: https://lnkd.in/gEzViKeP

We developed LOA to find such errors in perception data (accepted to SIGMOD 2022). We deployed LOA over the Lyft Level 5 perception dataset and successfully found errors in every validation scene with an error. We've open-sourced our code for general use (https://lnkd.in/guAuB-38) and have more details in our blog post (https://lnkd.in/gEzViKeP) and full paper (https://lnkd.in/gpJcrxF6).

This is joint work with Nikos, Sudeep Pillai, Peter Bailis, and Matei Zaharia



17 · 2 Comments

Like Comment Share

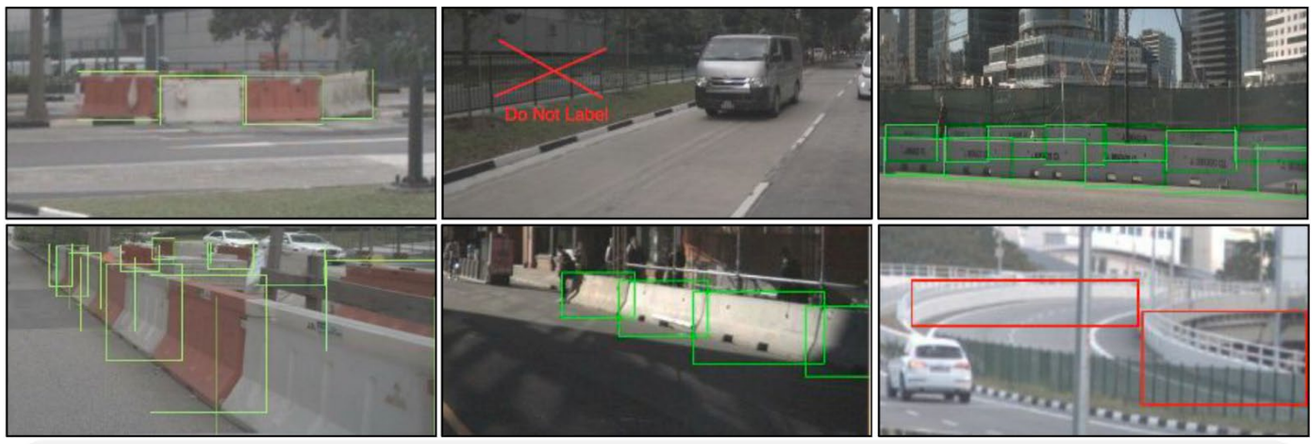
Ashesh Jain 2y
Co-founder & CEO Coram AI

While I cannot comment on the entire validation set labeling quality, it appears the images you have attached in this post are not labeling errors, but the "labeling policy" to ignore dynamic objects which are far from the drivable surface. Typically it is quite expensive (and also wasteful) to label all parked cars in a parking lot. If I recall, the data set is released with a semantic map for the same reason, which can be used to filter out such unlabeled objects. Also, I am not sure if you looked at the "No label zone" boxes, typically such parked cars should be included in the "no label zone" boxes so that they can be accounted for in the metrics calculation & model training. While there could certainly be errors in the validation set, the missing cuboids in the images attached above are likely due to an explicit labeling policy.

Like · Reply | 17 Reactions

https://www.linkedin.com/posts/daniel-kang-1223b343_ml-models-are-increasingly-being-deployed-activity-6891441029417963520-407l

Claim: We should treat instruction prompts as multimodal artifacts



Barrier

- ➔ *Any metal, concrete or water barrier temporarily placed in the scene in order to re-direct vehicle or pedestrian traffic. In particular, includes barriers used at construction zones.*
- ➔ *If there are multiple barriers either connected or just placed next to each other, they should be annotated separately.*
- ➔ *If barriers are installed permanently, then do **NOT** include them.*

Instructions are multimodal (vision + language) few-shot definitions of a concept… often obtained after painstaking conversations between annotators and stakeholders

(Controversial) claim: in the future, such prompts will be worth more than the dataset+labels themselves!

Thinking Like an Annotator: Generation of Dataset Labeling Instructions

Nadine Chang
Carnegie Mellon University
nadinec@cmu.edu

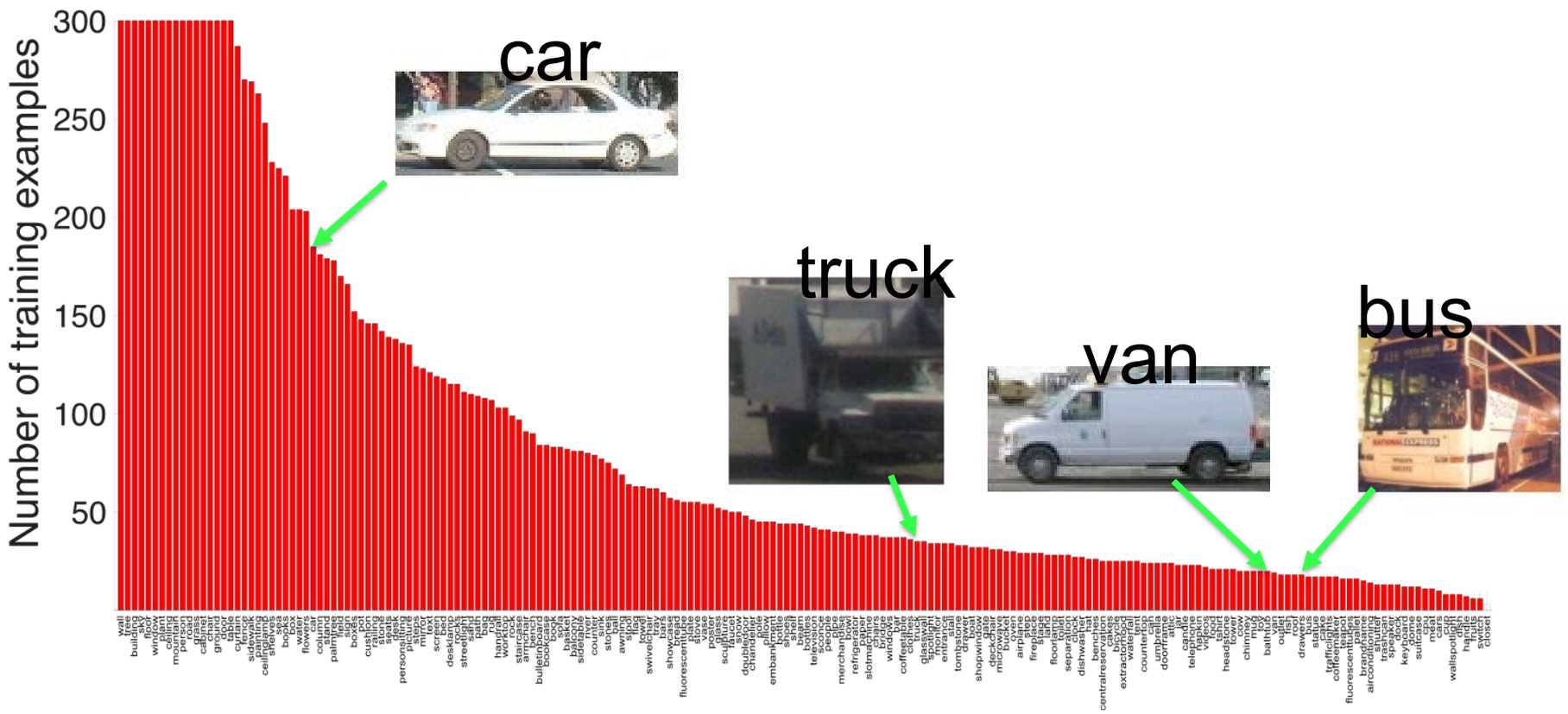
Francesco Ferroni
Nvidia
francescoferroni1@gmail.com

Michael J. Tarr
Carnegie Mellon University
michaeltarr@cmu.edu

Recognition

- Motivation
- How to define problem?
  - Classifiation / VQA / (panoptic) segmentation
    - **Dataset (bias)**
    - Metrics / losses
- Case example: finding people
- Statistical classification

Challenges of “long tail” distributions



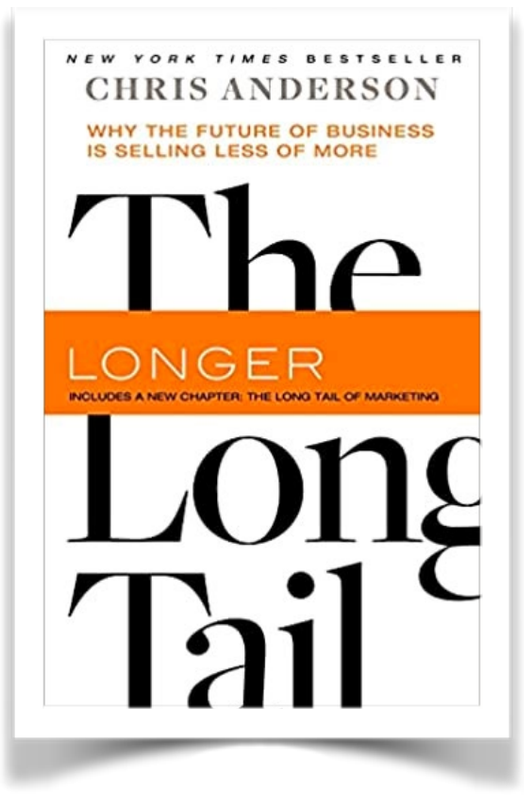
Classes sorted by frequency

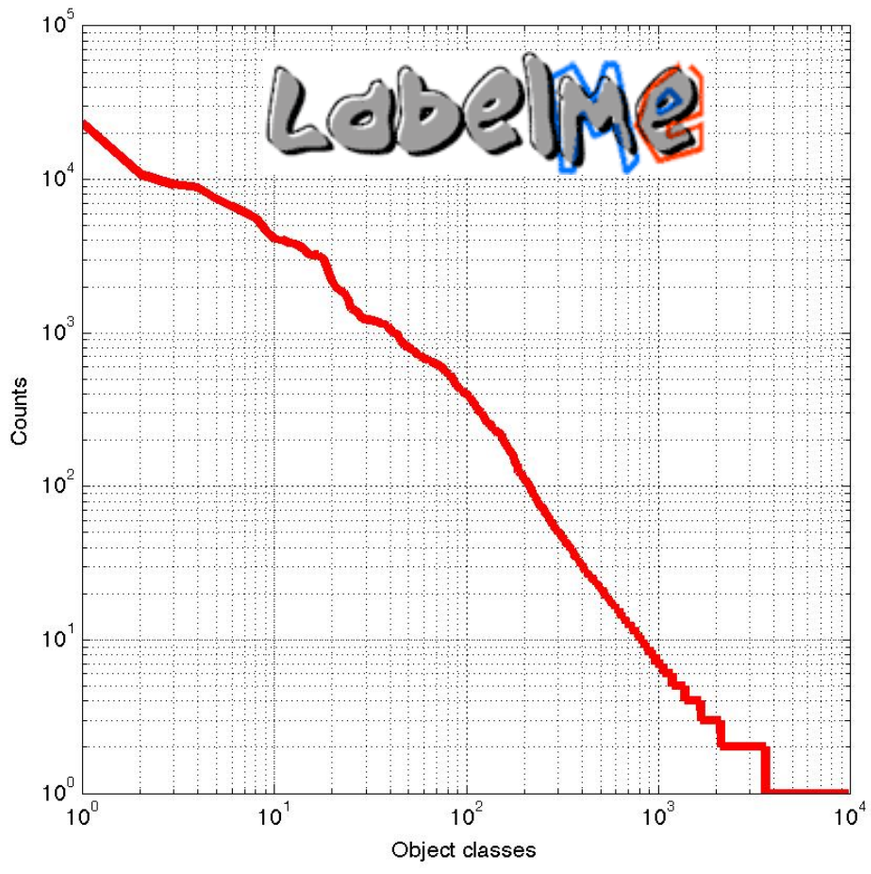
The first 9 objects account for 50% of all training examples
17 classes with more than 300 examples
109 classes with less than 50 examples

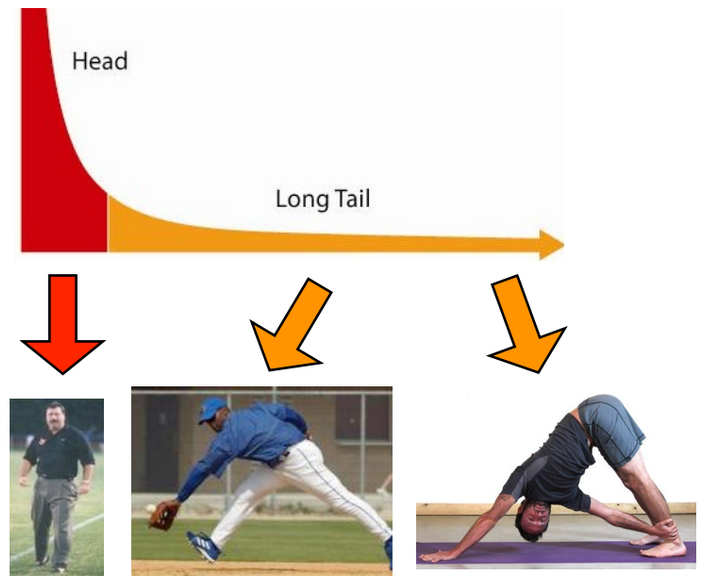
Salakhutdinov, Torralba, and Tenenbaum, CVPR, 2011

Naturally occuring distributions (Zipf's law)

Count of elements at rank k $\propto \frac{1}{k^{\alpha}}$





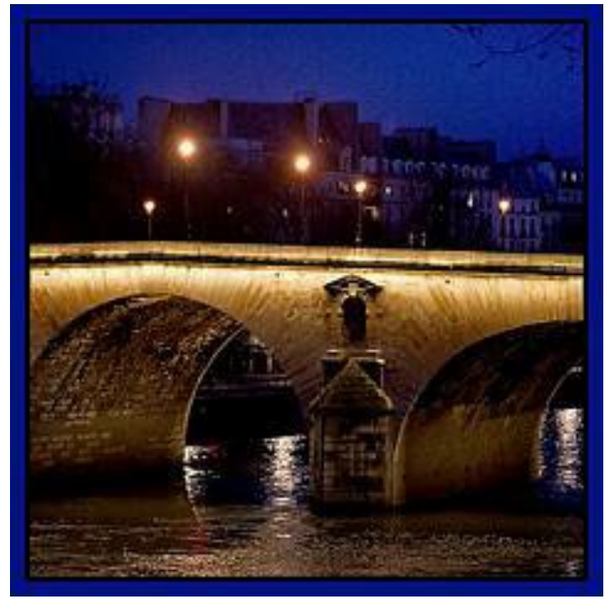


log count = -α log k

Dataset Bias

- Internet is a tremendous repository of visual data (Flickr, YouTube, Picassa, etc)
- But it's not random samples of visual world
- Many sources of bias:
  - Sampling bias
  - Photographer bias
  - Social bias

Flickr Paris

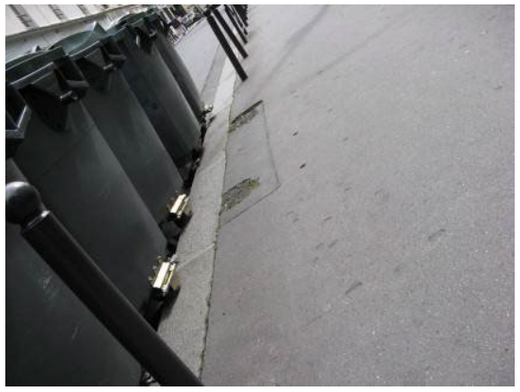


Real Paris

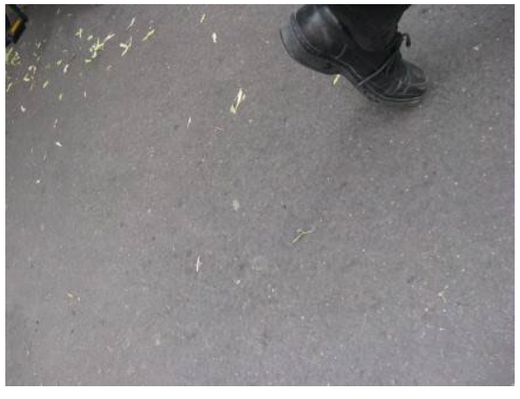
(random images taken by an “agent” walking in Paris)







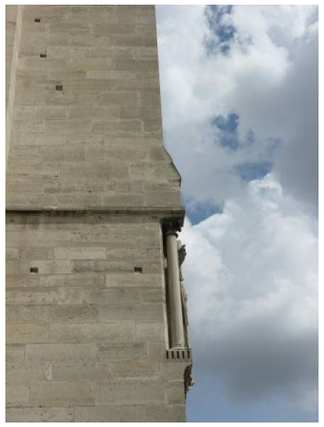


“Instagram vs no-filter”

Photographer Bias

- People want their pictures to be recognizable and/or interesting

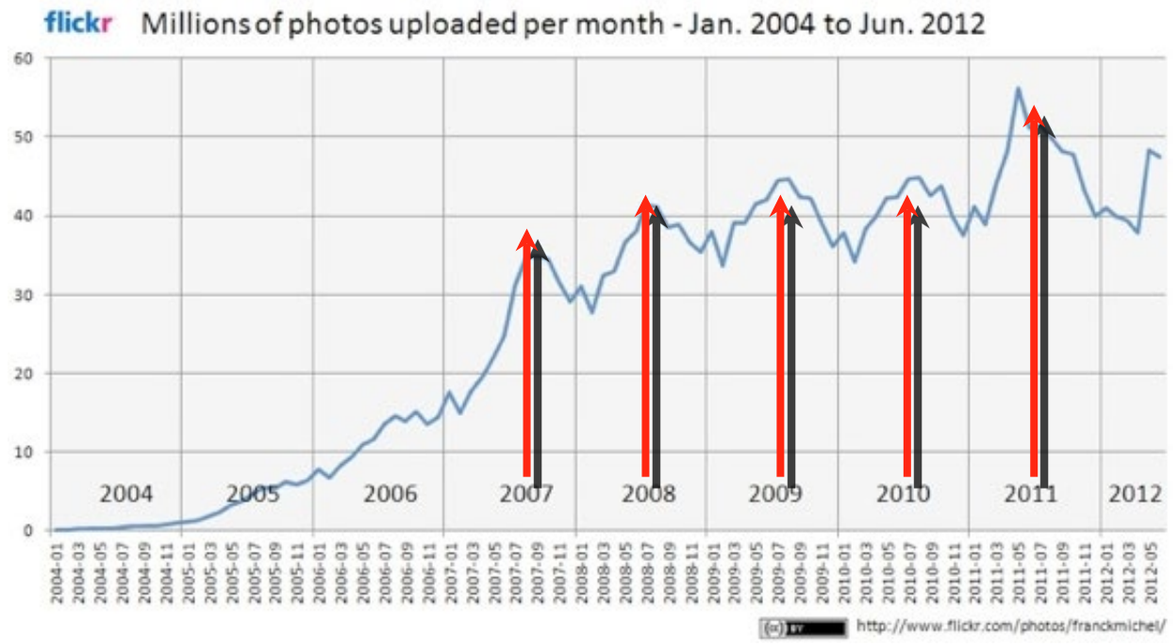


vs.

Sampling Bias

(space)

(time)

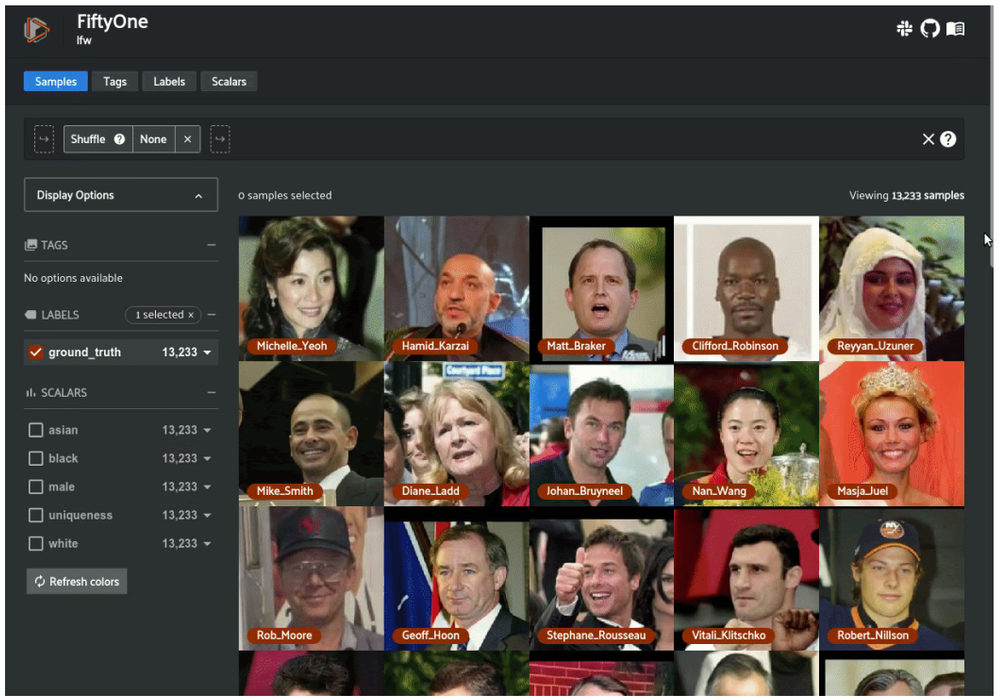


Why are there peaks in June-Aug?

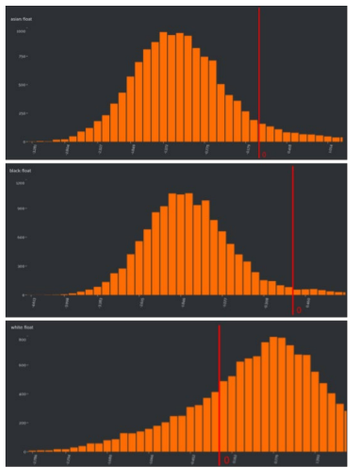
People tend to vacation and take more pictures

Labeled Face in the Wild, 2007 (13,233 images)

Social Bias



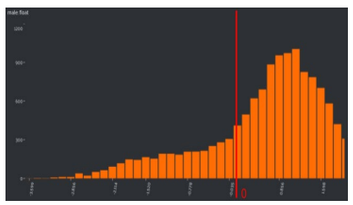
© Eric Hofesmann



Asian

African American

White



Male

George W. Bush: 500+ images
On average 2 images per person

Best practice: *document* data collection

Datasheets for Datasets

TIMNIT GEBRU, Black in AI
JAMIE MORGENSTERN, University of Washington
BRIANA VECCHIONE, Cornell University
JENNIFER WORTMAN VAUGHAN, Microsoft Research
HANNA WALLACH, Microsoft Research
HAL DAUMÉ III, Microsoft Research; University of Maryland
KATE CRAWFORD, Microsoft Research

Sample Questions

Motivation

- For what purpose was the dataset created?
- Who created the dataset?
- Who funded the creation of the dataset?
- …

Collection Process

- Does the dataset identify any subpopulations?
- Is it possible to identify individuals?
- Does the dataset contain data that might be considered sensitive in any way?
- How was the data associated with each instance acquired?
- …

Ethics and Societal Impact

- Were any ethical review processes conducted?
- Has an analysis of the potential impact of the dataset and its use on data subjects
- …

Unbiased Look at Dataset Bias

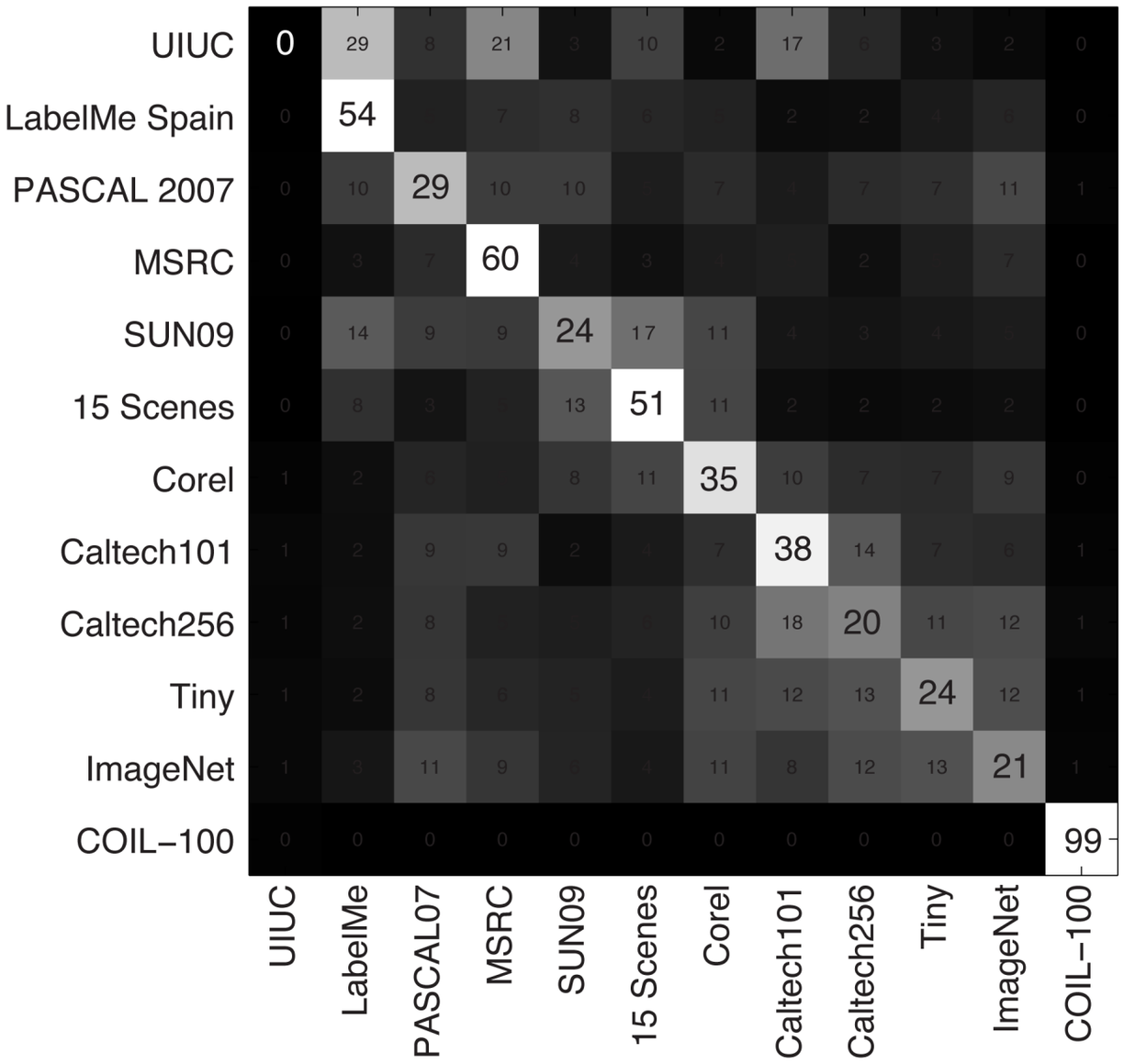
Antonio Torralba
Massachusetts Institute of Technology
torralba@csail.mit.edu

Alexei A. Efros
Carnegie Mellon University
efros@cs.cmu.edu

- Caltech 101
- Caltech 256
- MSRC
- UIUC cars
- Tiny Images
- Corel
- PASCAL 2007
- LabelMe
- COIL-100
- ImageNet
- 15 Scenes
- SUN’09

“Name That Dataset!” game

Build a classifier to play *"Name that dataset!"*



- 12 1-vs-all classifiers
- Standard full-image features

- 39% performance (chance is 8%)

Best practice: evaluate cross-dataset performance!

SUN

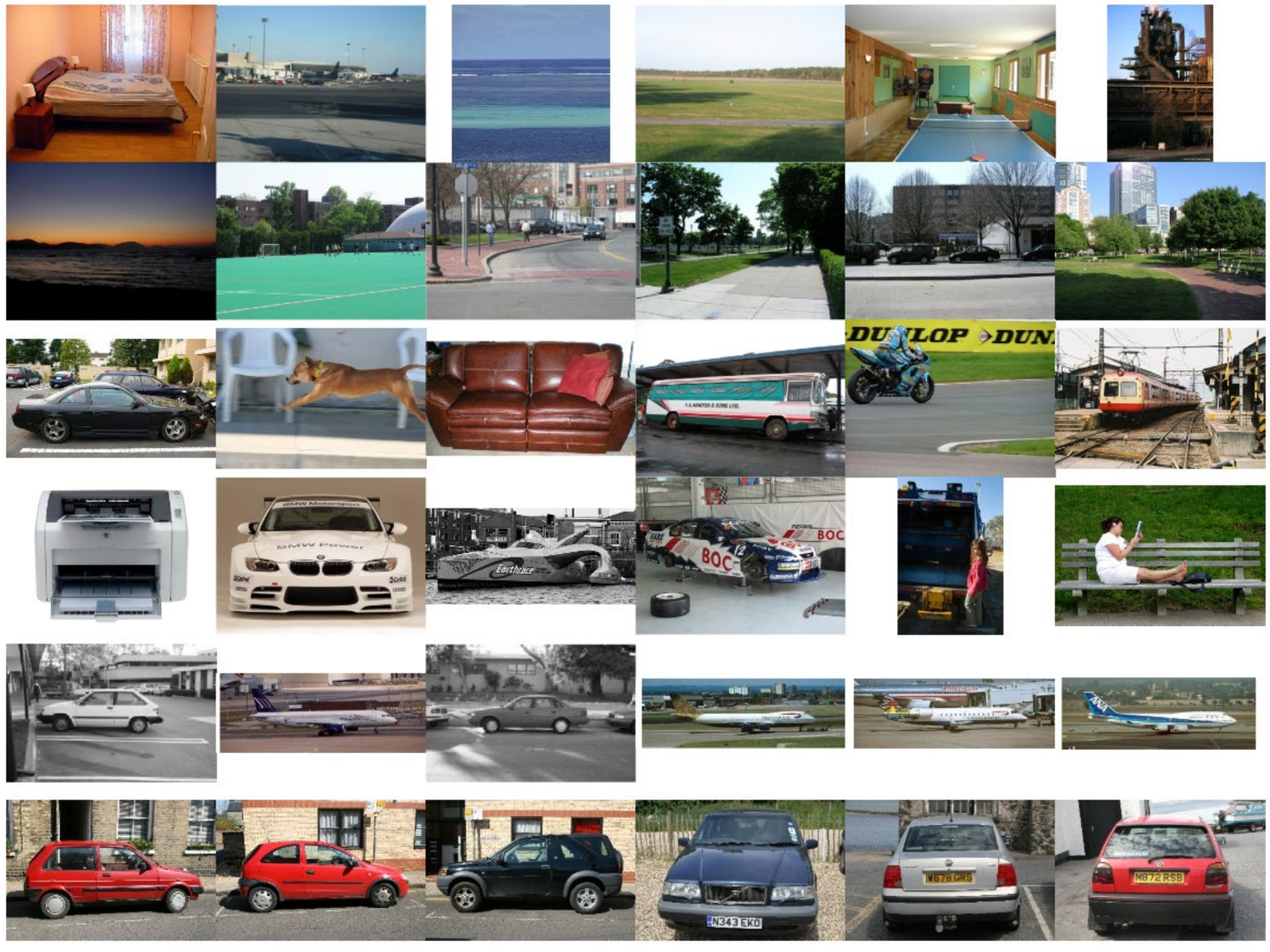
LabelMe

PASCAL

ImageNet

Caltech101

MSRC



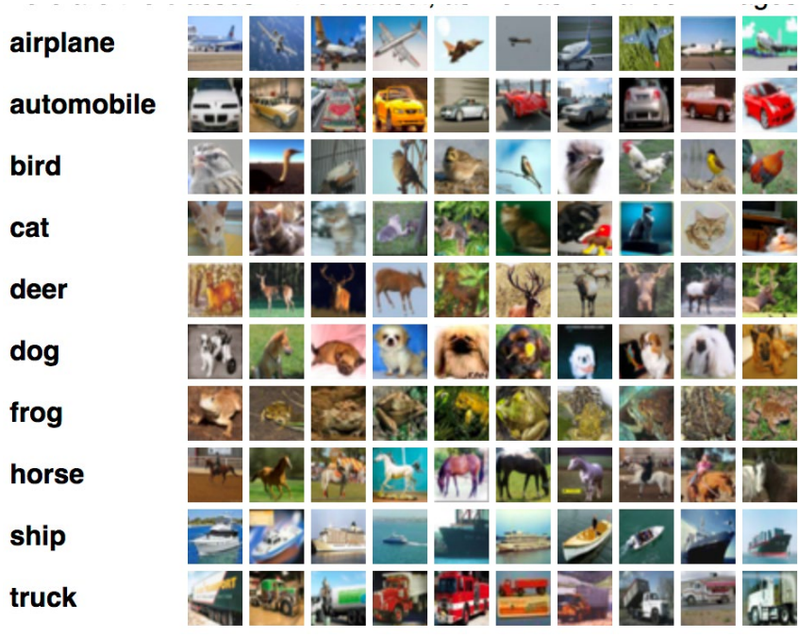
Classifier trained on MSRC cars

Recognition

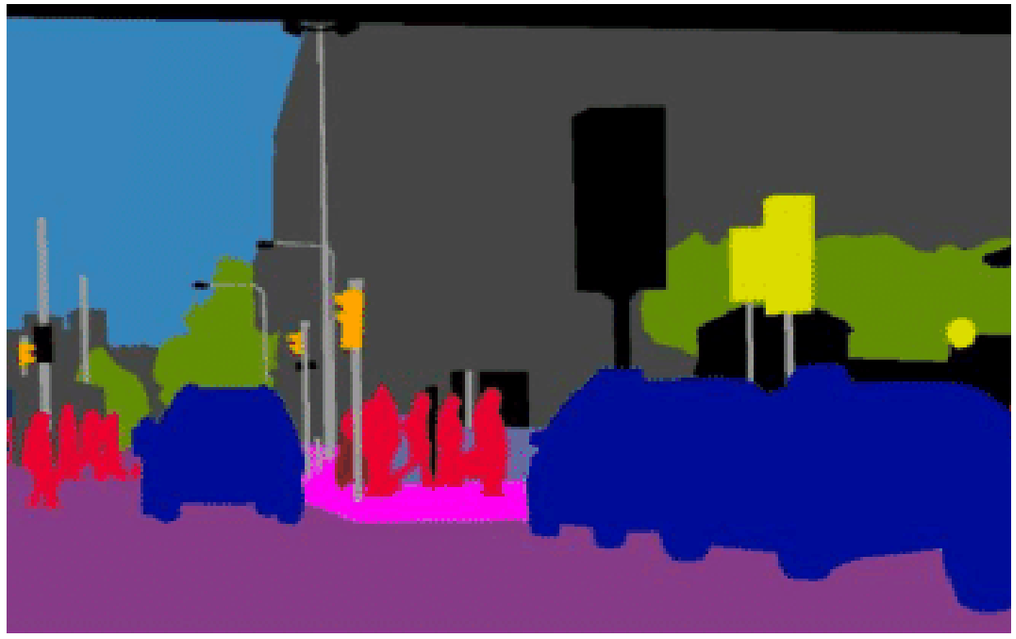
- Motivation
- How to define problem?
  - Classifiation / VQA / (panoptic) segmentation
  - Dataset (bias); cross-dataset generalization
  - **Metrics / losses**
- Case example: finding people
- Statistical classification

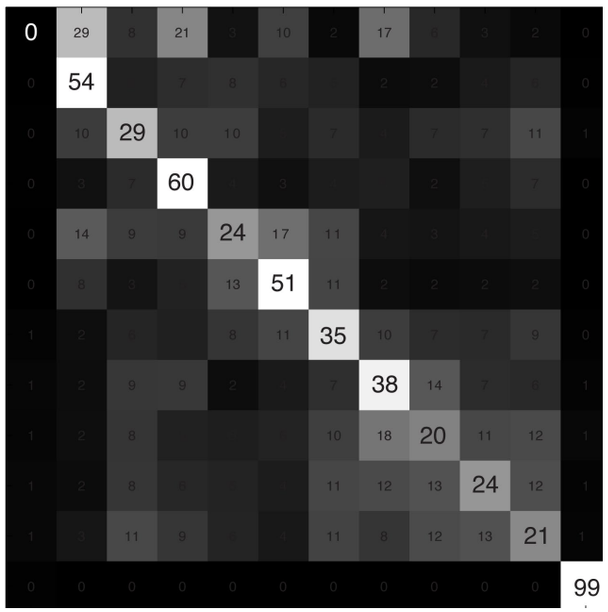
Final piece of puzzle: metrics and evaluation

Image classification



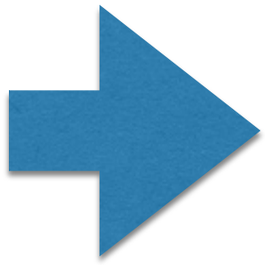
Semantic segmentation

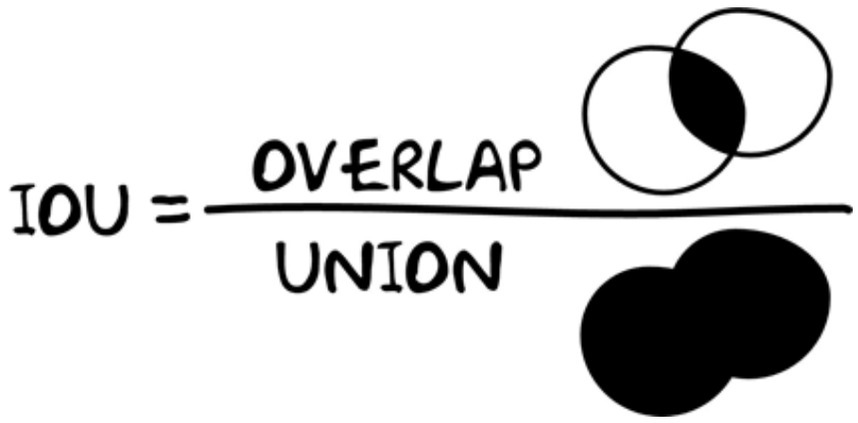




Top1,Top5 accuracy

Average accuracy per example or per class makes a large difference on long-tailed datasets. Why?



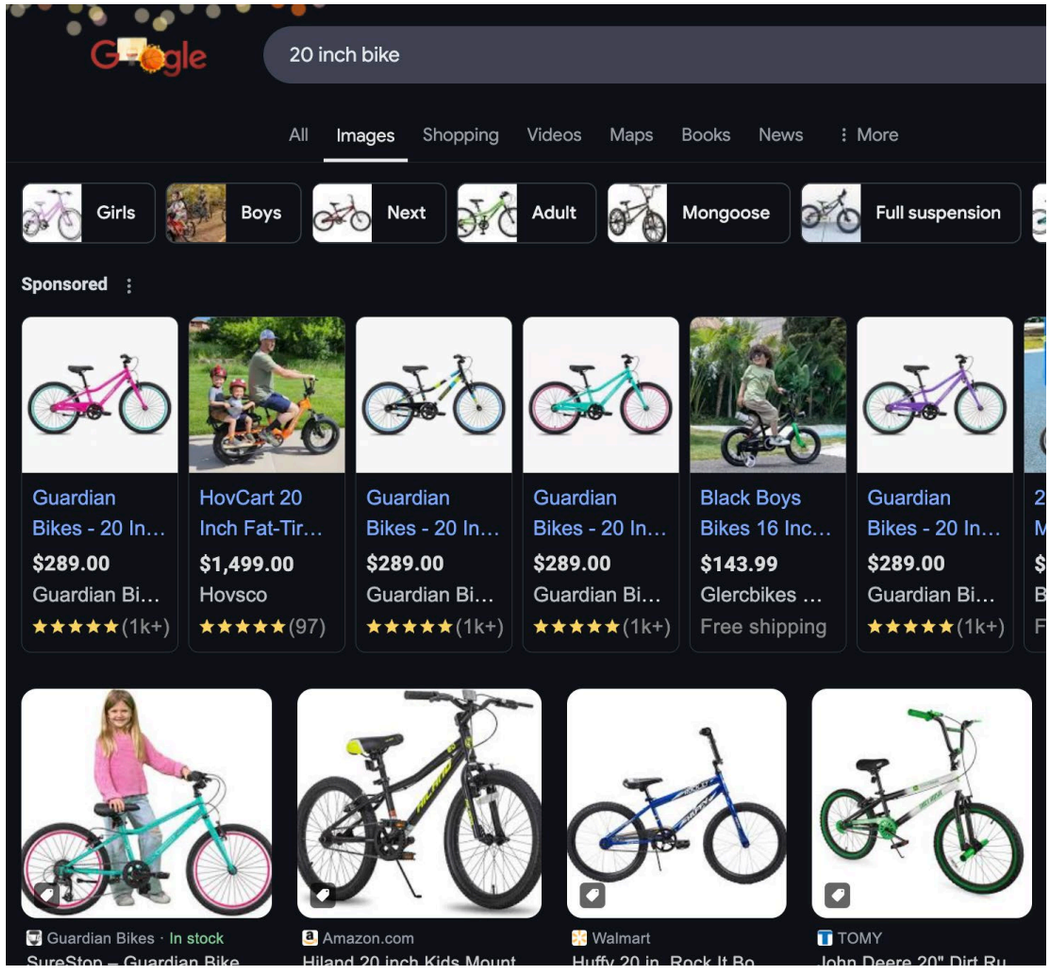


How to weight large + small objects equally?

Compute Intersection over Union (IoU) per class or per object

Challenge: how to evaluate imbalanced classification tasks?

(naively computing errors won't work; classifying everything as negative will reduce errors)

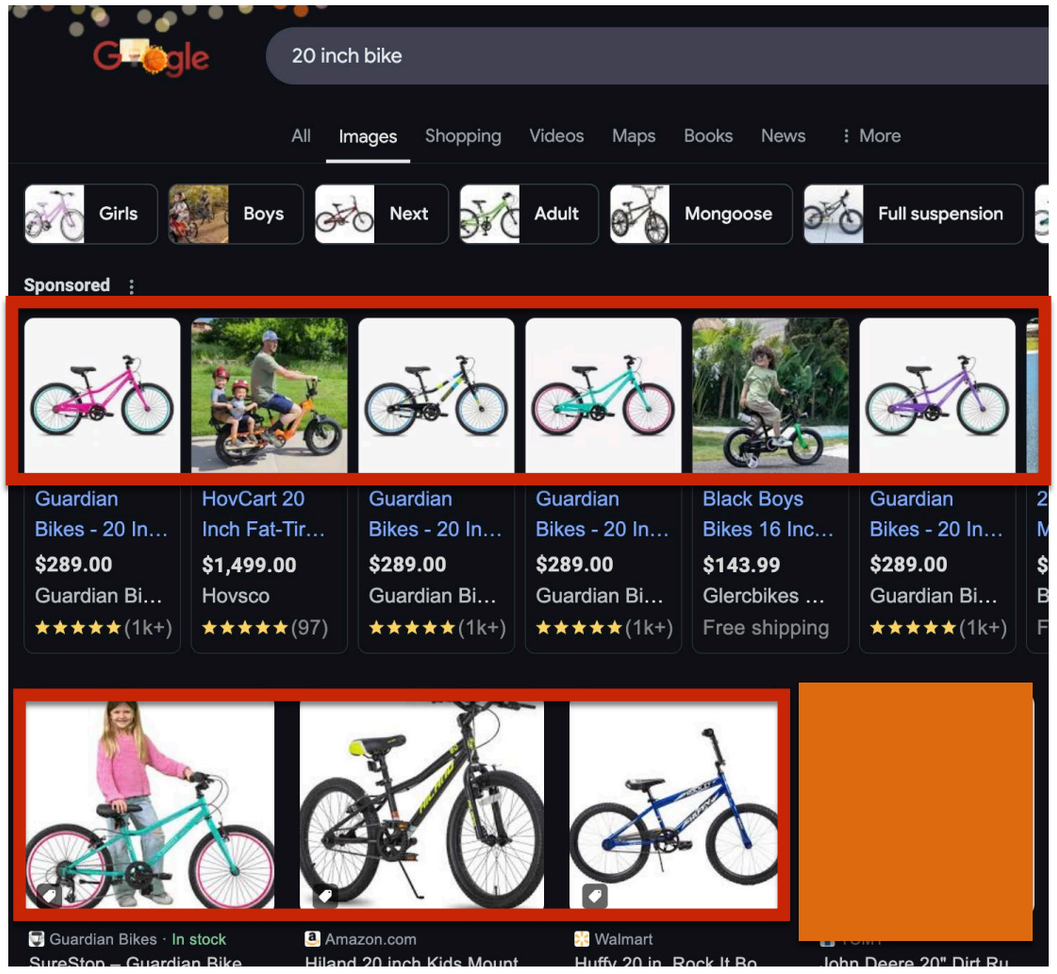


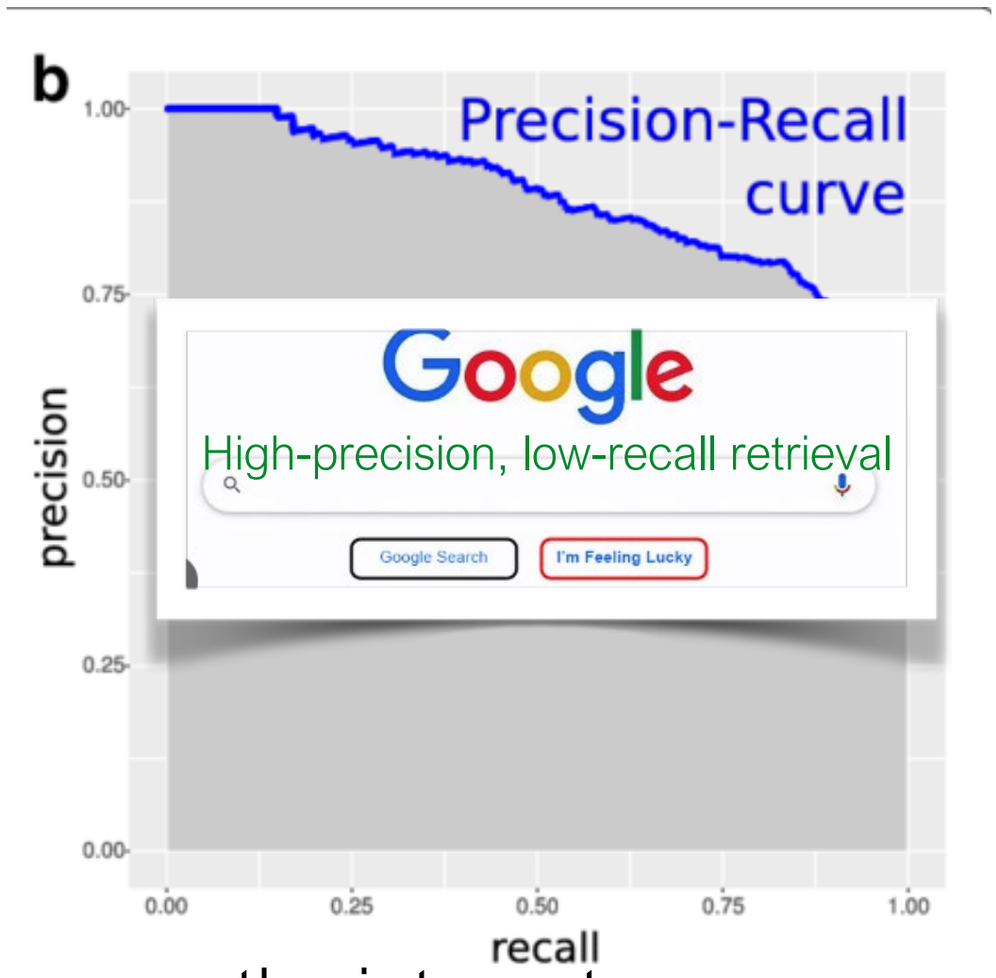
Let's find all the `bike` images on the internet

1. Assume algorithm produces a `bike` confidence for each internet image; different downstream apps might operate at different confidence thresholds

Challenge: how to evaluate imbalanced classification tasks?

(naively computing errors won't work; classifying everything as negative will reduce errors)





Let's find all the `bike` images on the internet

1. Assume algorithm produces a `bike` confidence for each internet image; different downstream apps might operate at different confidence thresholds

2. Construct a ***precision-recall*** curve

For each confidence threshold, count true positives, false positives and false negatives (or missed detections).

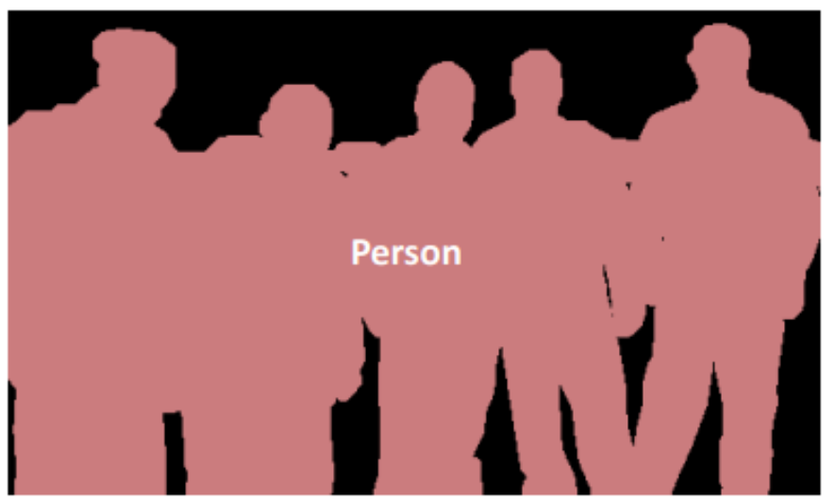
Precision = TP/(TP + FP) = 9/(9+1) = .9

Recall = TP/(TP + FN) = 9/(9+1000000000) = .005

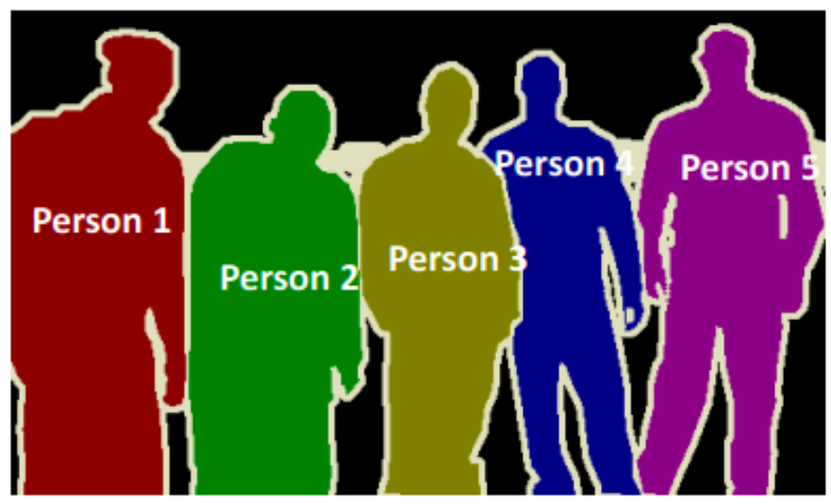
3. Summarize algorithm accuracy with ***area under curve*** ("average precision")

Application to object detection / instance segmentation

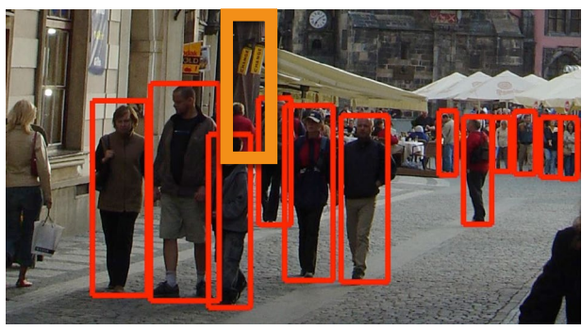
Object Detection



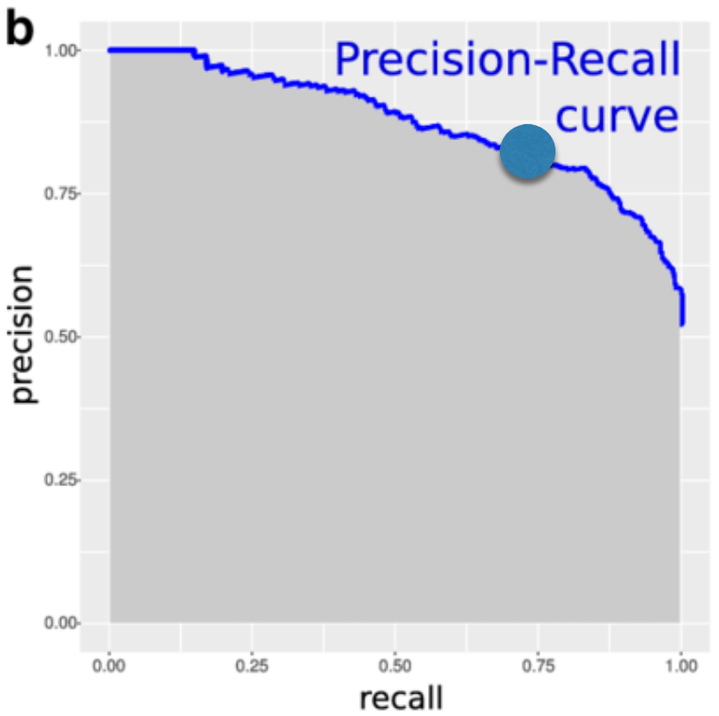
Semantic Segmentation



Instance Segmentation



Precision = TP/(TP + FP)
Recall = TP/(TP + FN)



Datasets have different goals…

- Some are object-centric (e.g. Caltech, ImageNet)
- Others are scene-centric (e.g. LabelMe, SUN’09)
- The process of benchmark and dataset curation is of increasing value…

NeurIPS 2025 Datasets & Benchmarks Track Call for Papers

The **NeurIPS Datasets and Benchmarks track** serves as a venue for high-quality publications on highly valuable machine learning datasets and benchmarks crucial for the development and continuous improvement of machine learning methods. Previous editions of the Datasets and Benchmarks track were highly successful and continuously growing (accepted papers 2021, 2002, and 2023, and best paper awards 2021, 2022, 2023 and 2024. Read our original blog post for more about why we started this track, and the 2025 blog post announcing this year's track updates.

Dates and Guidelines

Please note that the Call for Papers of the NeurIPS2025 Datasets & Benchmarks Track this year **will follow the Call for Papers of the NeurIPS2025 Main Track, with the addition of three track-specific points:**

- Single-blind submissions
- Required dataset and benchmark code submission
- Specific scope for datasets and benchmarks paper submission

The dates are also identical to the main track:

- **Abstract submission deadline:** May 11, 2025 AoE
- **Full paper submission deadline:** May 15, 2025 AoE (all authors must have an OpenReview profile when submitting)
- **Technical appendices and supplemental materials deadline:** May 22, 2025 AoE
- **Author notification:** Sep 18, 2025 AoE
- **Camera-ready:** Oct 23, 2025 AoE

Accepted papers will be published in the NeurIPS proceedings and presented at the conference alongside the main track papers. As such, we aim for an equally stringent review as in the main conference track, while also allowing for **track-specific guidelines**, which we introduce below. For details on everything else, e.g. formatting, code of conduct, ethics review, important dates, and any other submission related topics, please refer to the main track CFP.

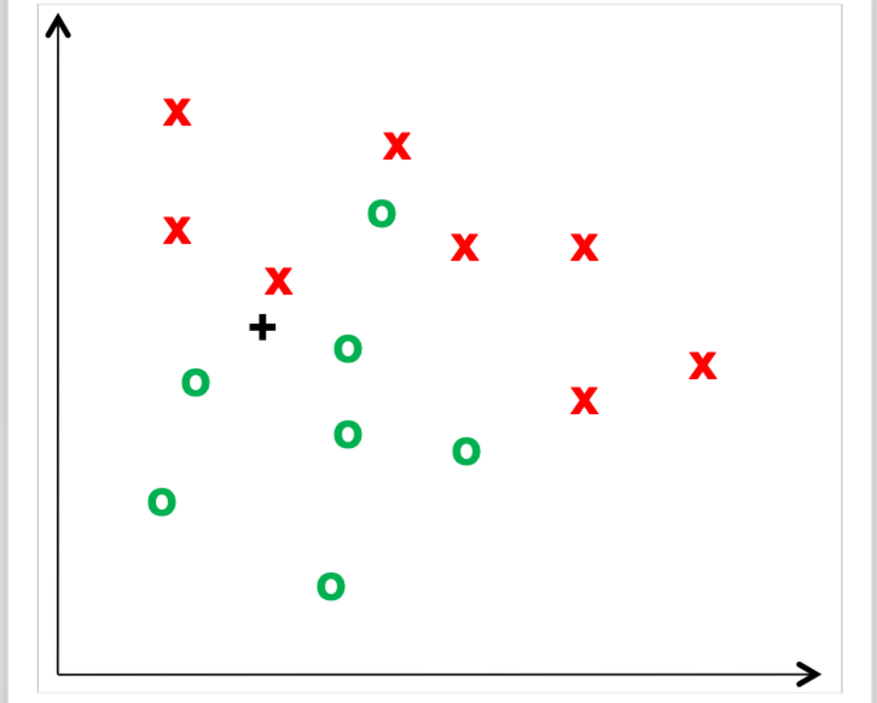
OpenReview

Submit at: https://openreview.net/group?id=NeurIPS.cc/2025/Datasets_and_Benchmarks_Track

The site will start accepting submissions on April 3, 2025 (at the same time as the main track).

Note: submissions meant for the main track should be submitted to a different OpenReview portal, as shown here. Papers will not be transferred between the main and the Datasets and Benchmarks tracks after the submission is closed.

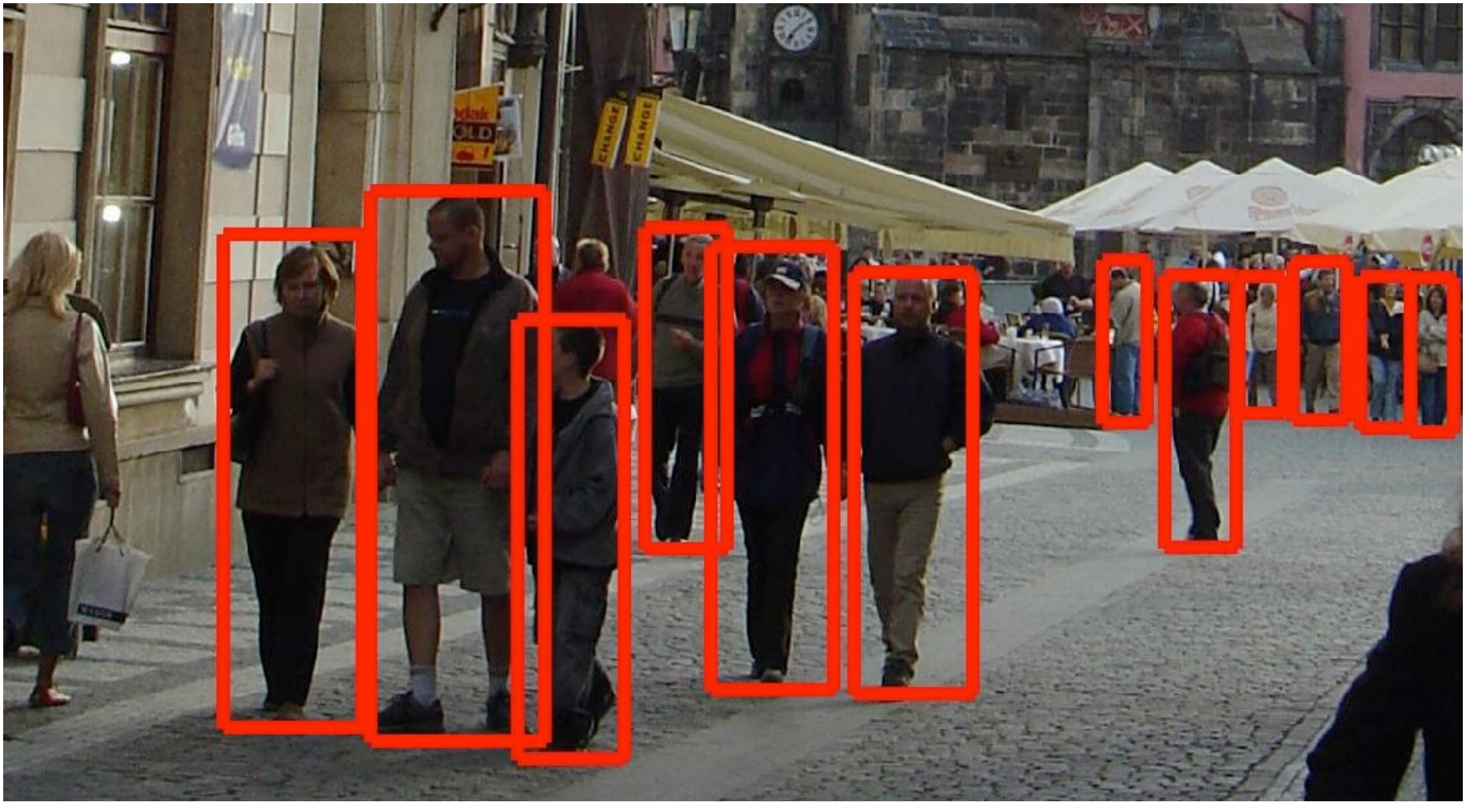
Linear classification



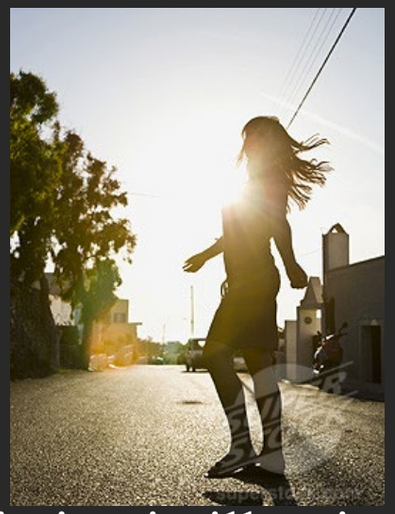
Outline

- Motivation
- How to define problem?
  - Classifiation / VQA / (panoptic) segmentation
  - Dataset (bias)
  - Metrics / losses
- **Case example: finding people**
- Statistical classification

Goal: detecting pedestrians



Thought experiment: let's build a person detector.
Why is this difficult?



variation in illumination

variation in appearance

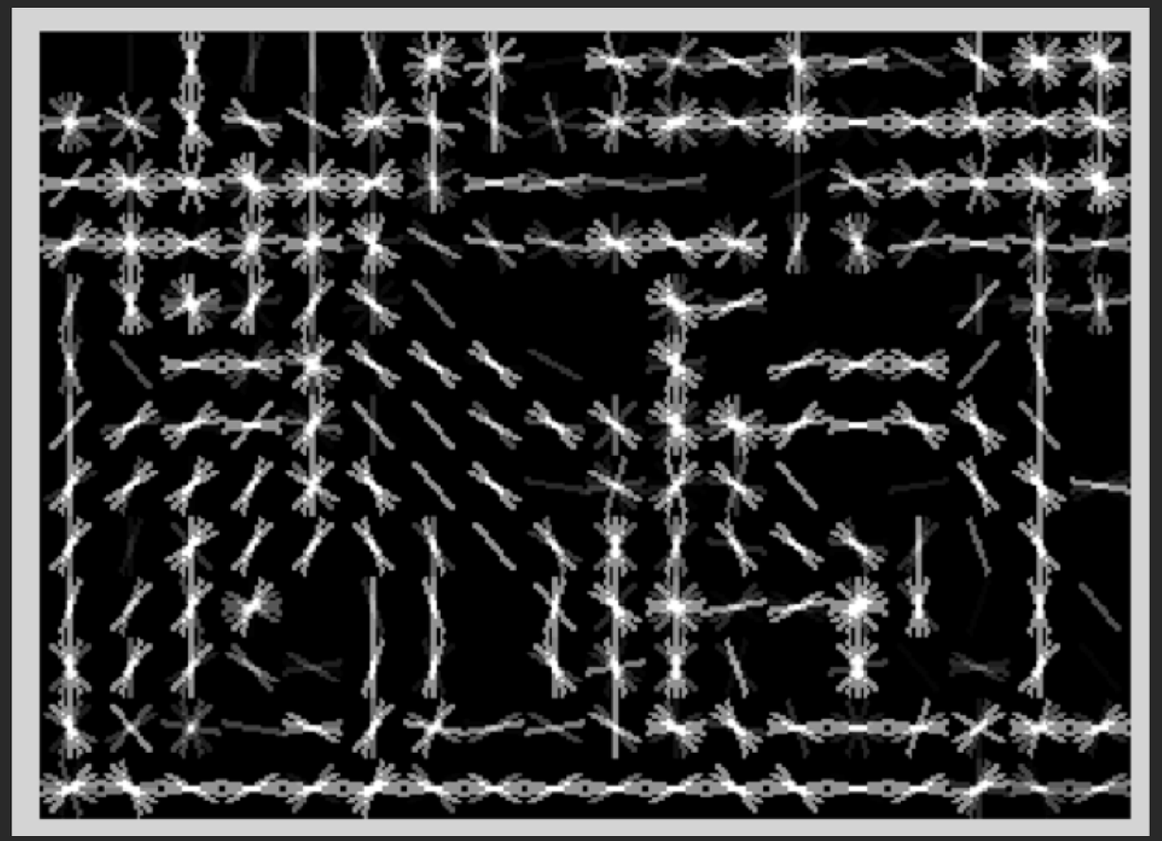
variation in pose, viewpoint

occlusion & clutter

Classic "nuisance factors" for general object recognition

Main idea:
use “invariant edge features”

Histograms of oriented gradients; bin 9 gradient orientations over 8x8 pixel neighborhoods



Neat approach to visualizing an edge histogram:
draw an oriented edge with the histogram weight

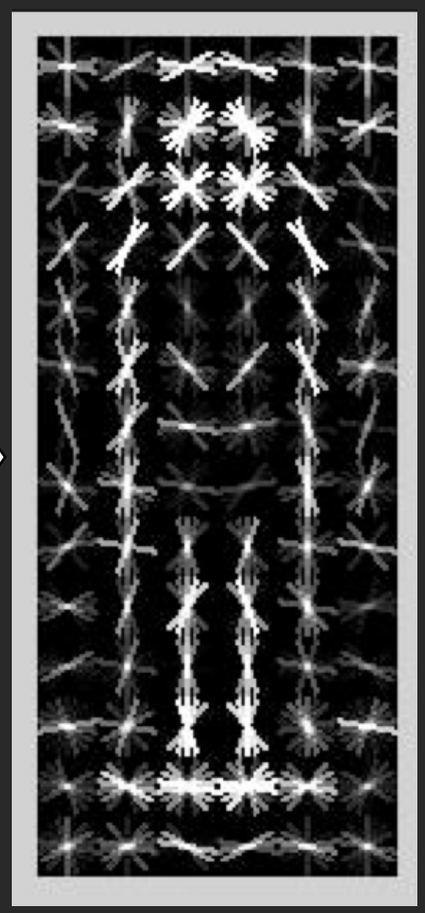
(Dalal & Triggs 05)

Template classifiers

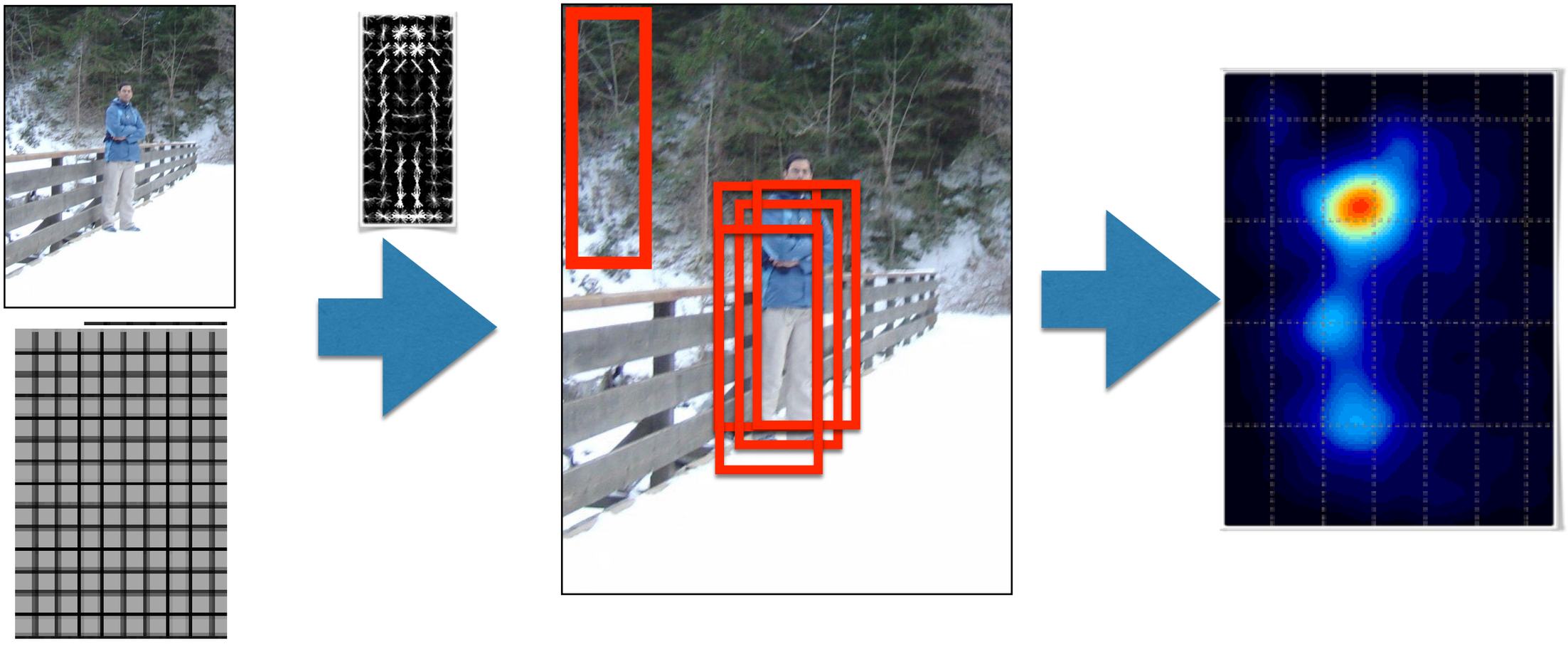
pos

neg

...

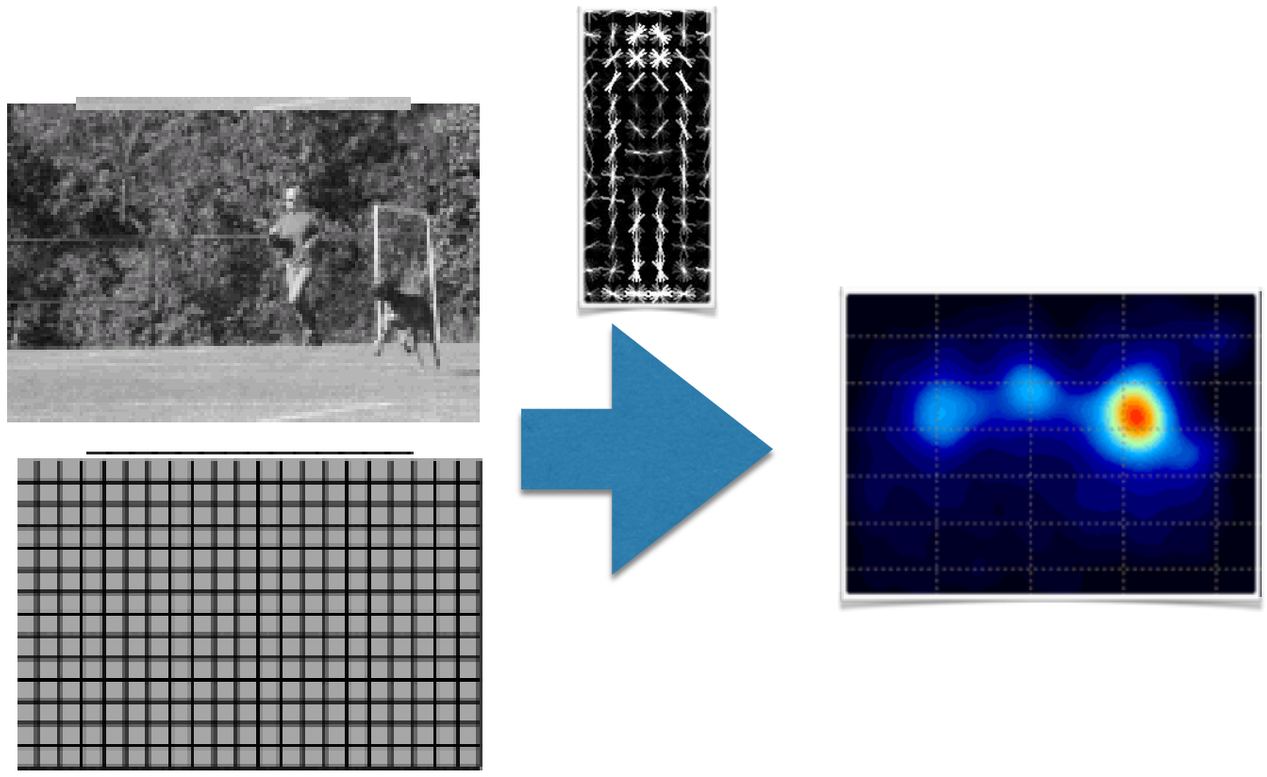


Template scoring

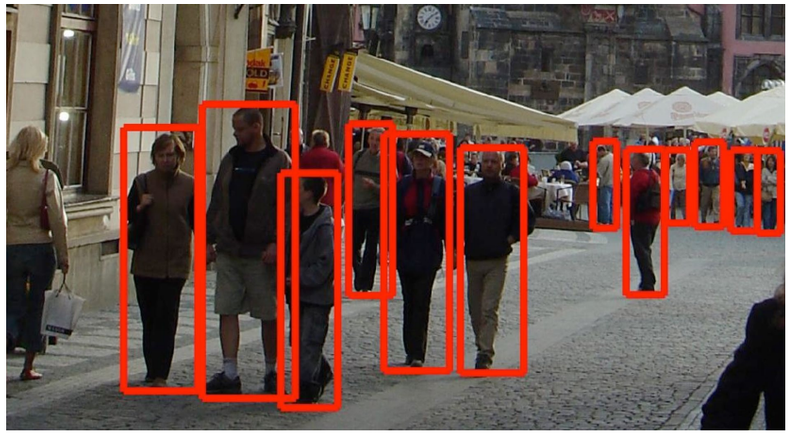


How can we fix output from naive thresholding?

Nonmaximal suppression

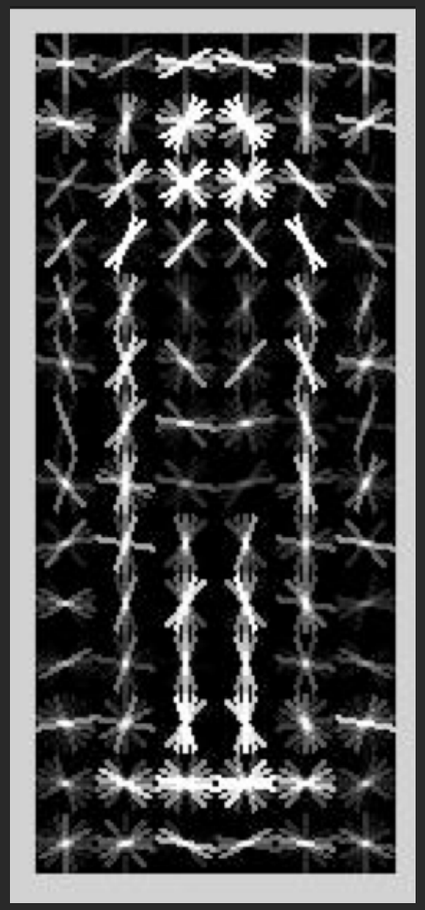


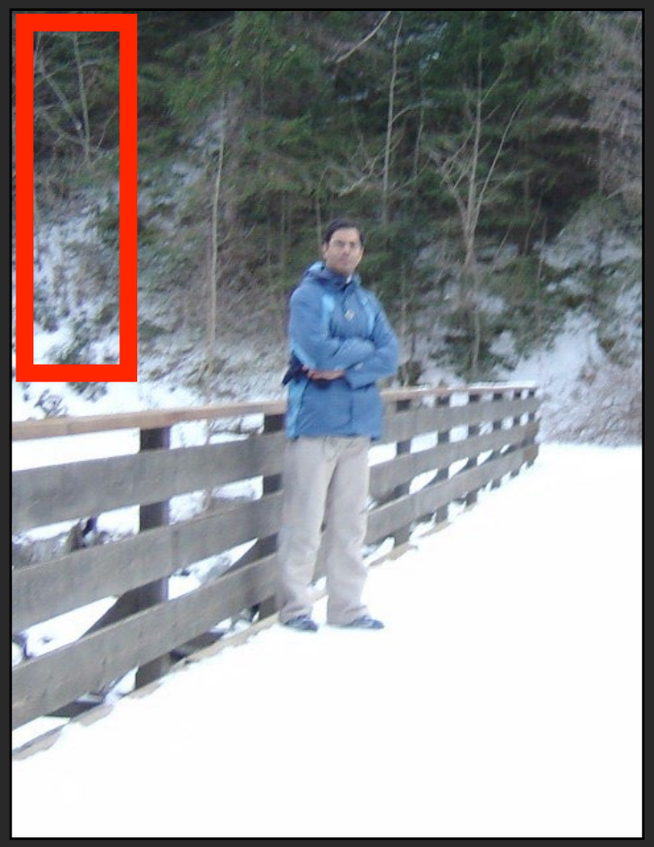
1. Find highest scoring location
2. Zero-out responses that overlap
3. Repeat until highest remaining score is below a threshold

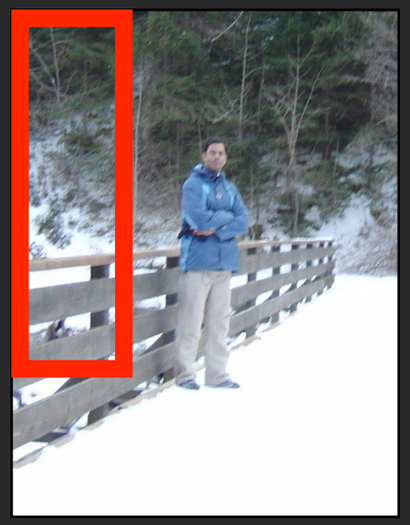


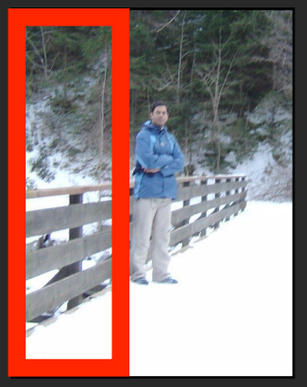
How do we find different-size people?

Run template over Gaussian pyramid!





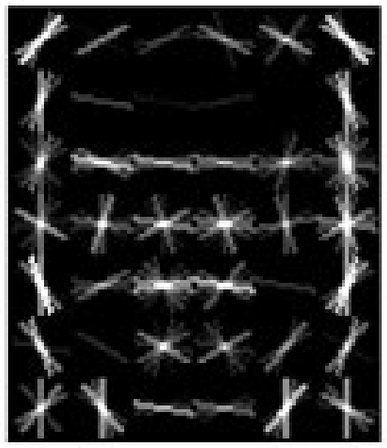




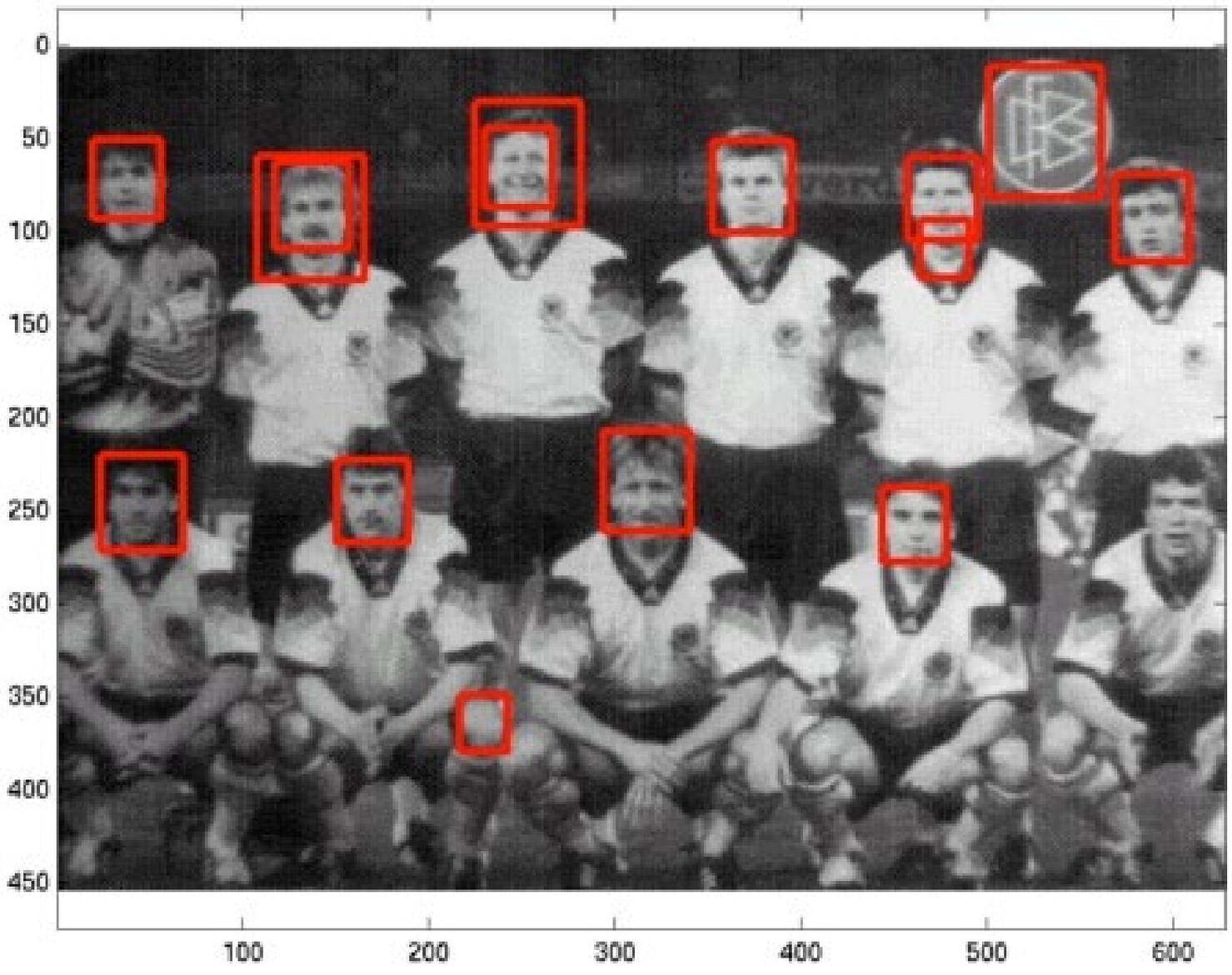
Pedestrian detection

Dalal and Triggs “Histograms of Gradients”

Face detection



template



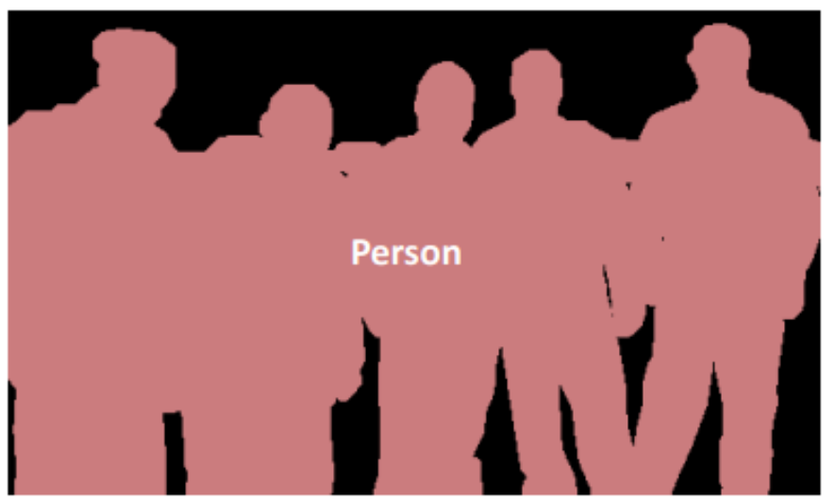
Challenge: how to evaluate detectors?

For now, let's look at simpler task of *imbalanced* image classification, instead of object detection

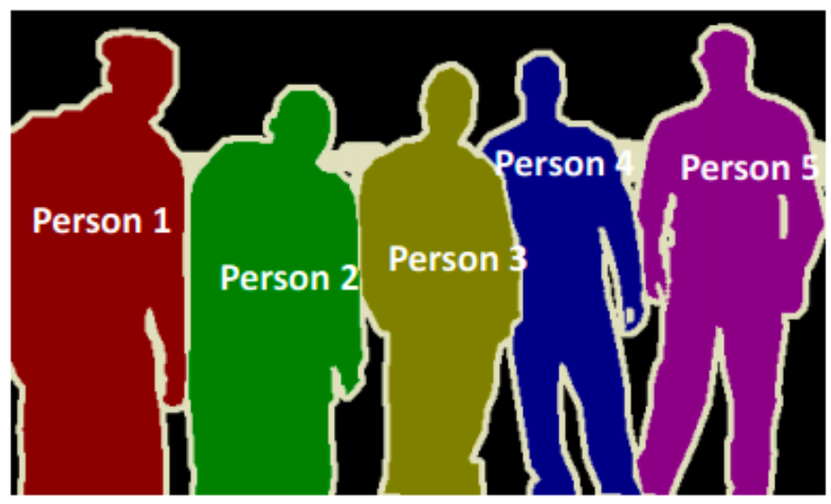
Here, computing errors is difficult since classifying everything as negative will reduce error

Application to object detection / instance segmentation

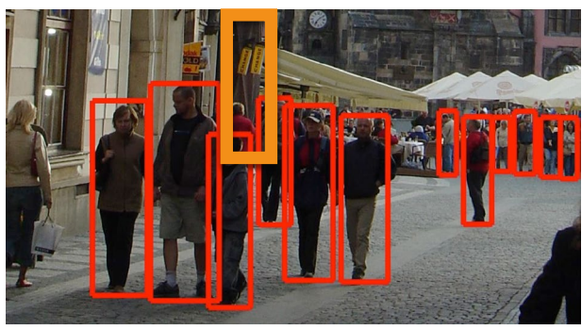
Object Detection



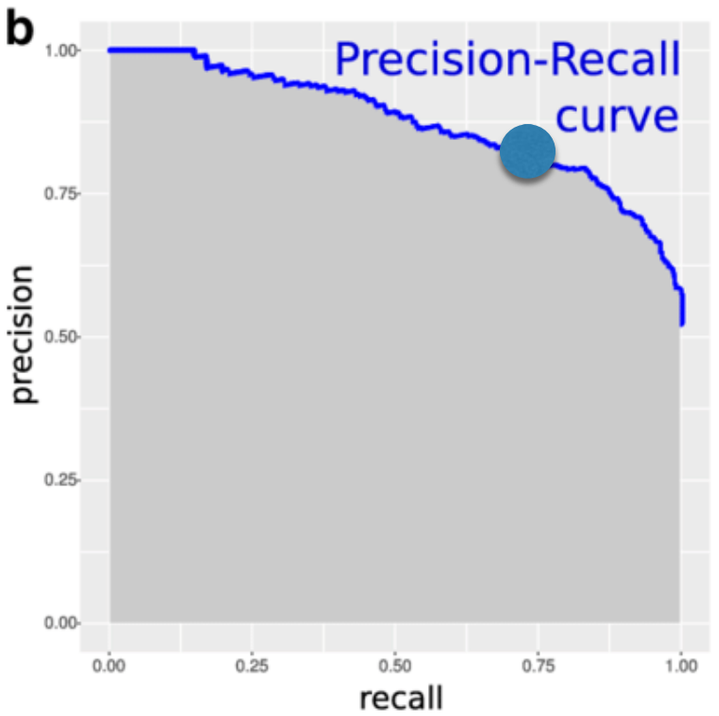
Semantic Segmentation



Instance Segmentation

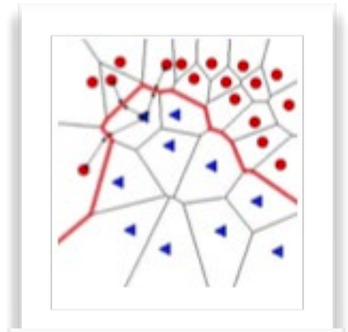


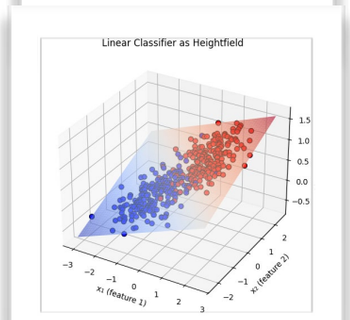
Precision = TP/(TP + FP)
Recall = TP/(TP + FN)

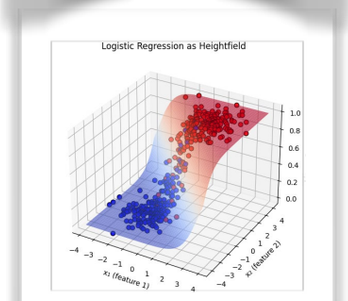


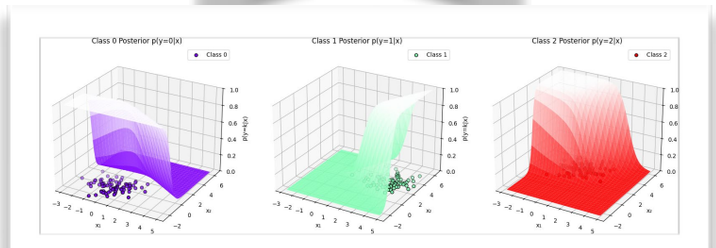
Statistical classification

- Nearest neighbors
- Linear classification
    - Linear regression
    - Linear discriminant analysis (LDA)
    - Logistic regression (& sigmoids)
    - Softmax regression / classification
- Loss minimization
    - least squares vs cross-entropy loss
    - optimizers (SGD), regularizer, classifier, loss function





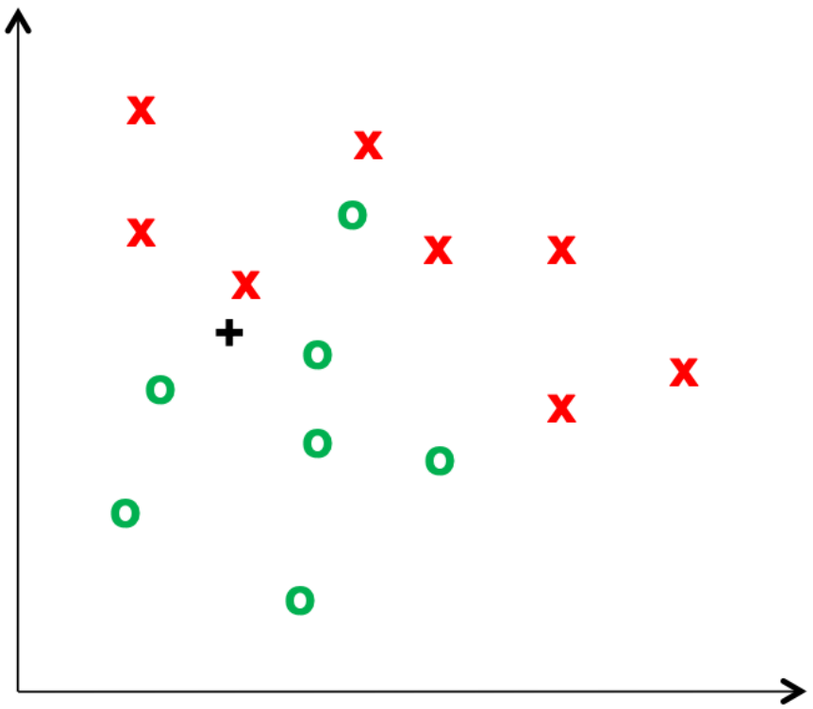




Statistical classification

$x_i \in R^N$

$y_i \in \{-1,1\}$



?

Recall N = 10x5x9 ≈ 500 for typical HOG templates

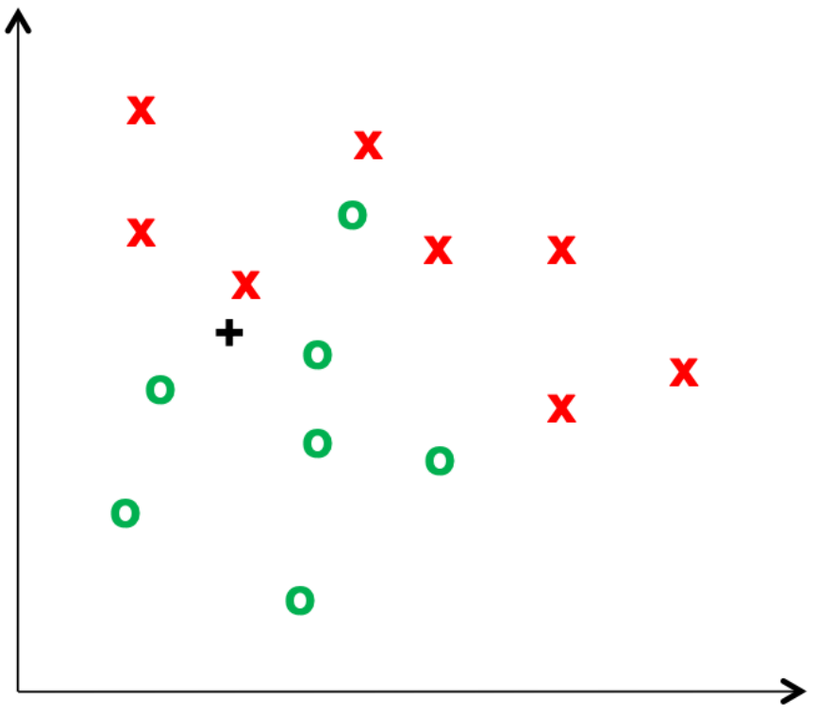
Given training points (x_i,y_i), learn function f(x) that predicts a label {-1,1}

Notation alert: for this lecture, I ***won't*** bold the (500-dim) vector x

Statistical classification

$x_i \in R^N$

$y_i \in \{-1,1\}$



?

Version 0: nearest neighbor classification

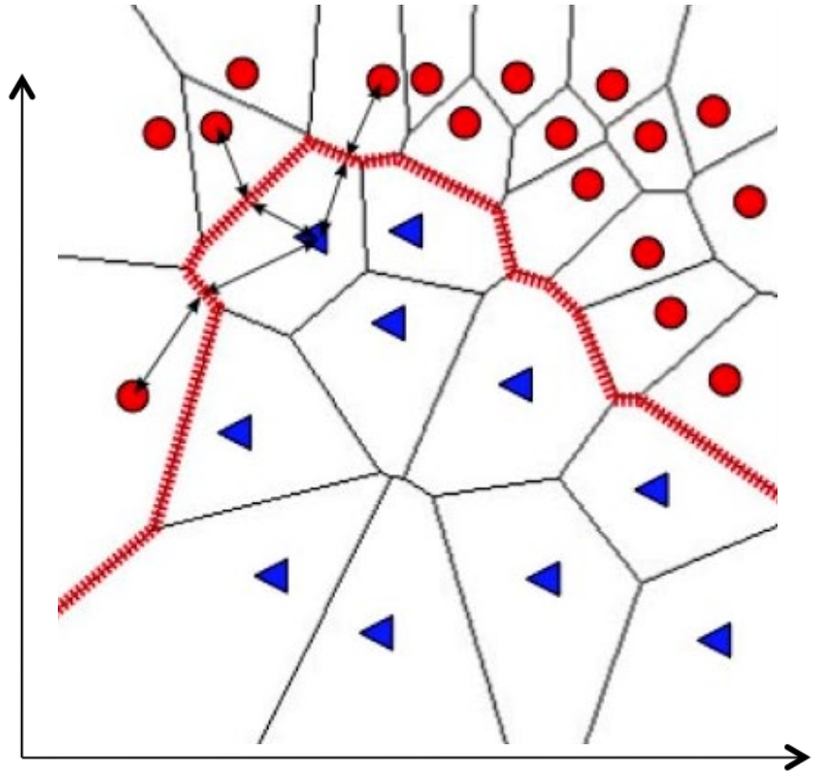
$$\text{Prediction}(x) = y_{i*} \quad \text{where} \quad i^* = \arg\min_i ||x - x_i||^2$$

- Train time: 0-cost
- Test time: ***really*** expensive
- Trivially handles multiple classes
- Surprisingly effective!

Statistical classification

$x_i \in R^N$

$y_i \in \{-1,1\}$



?

Version 0: nearest neighbor classification

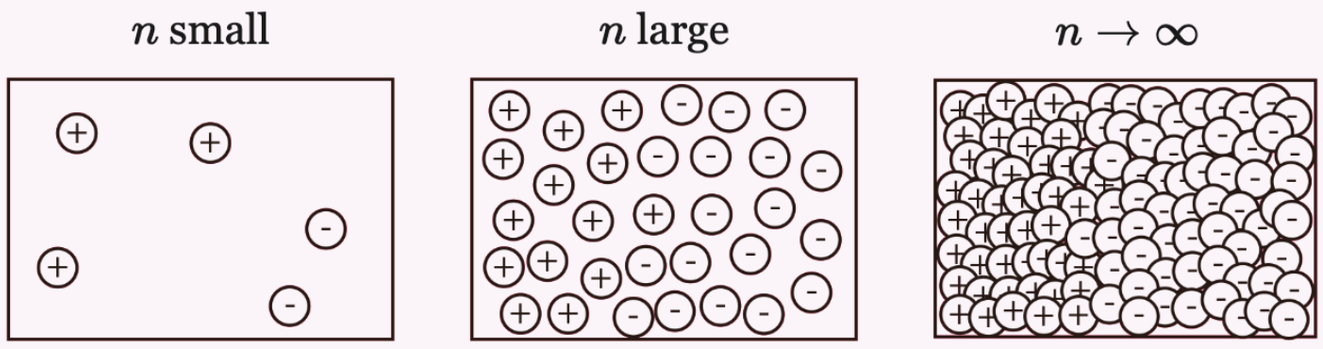
$$\text{Prediction}(x) = y_{i^*} \quad \text{where} \quad i^* = \arg\min_i ||x - x_i||^2$$

Decision boundary can be arbitrarily complex!

With big data, nearest neighbors is *optimal*

1-NN Convergence Proof

Cover and Hart 1967[1]: As $n \to \infty$, the 1-NN error is no more than twice the error of the Bayes Optimal classifier. (Similar guarantees hold for $k > 1$.)



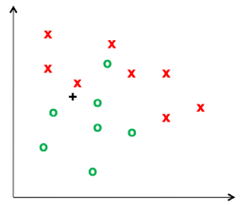
Let $\mathbf{x}_{\text{NN}}$ be the nearest neighbor of our test point $\mathbf{x}_{\text{t}}$. As $n \to \infty$, $\text{dist}(\mathbf{x}_{\text{NN}}, \mathbf{x}_{\text{t}}) \to 0$, i.e. $\mathbf{x}_{\text{NN}} \to \mathbf{x}_t$. (This means the nearest neighbor is identical to $\mathbf{x}_{\text{t}}$.) You return the label of $\mathbf{x}_{\text{NN}}$. What is the probability that this is not the label of $\mathbf{x}_{\text{t}}$? (This is the probability of drawing two different label of $\mathbf{x}$)

https://www.cs.cornell.edu/courses/cs4780/2022sp/notes/LectureNotes03.html

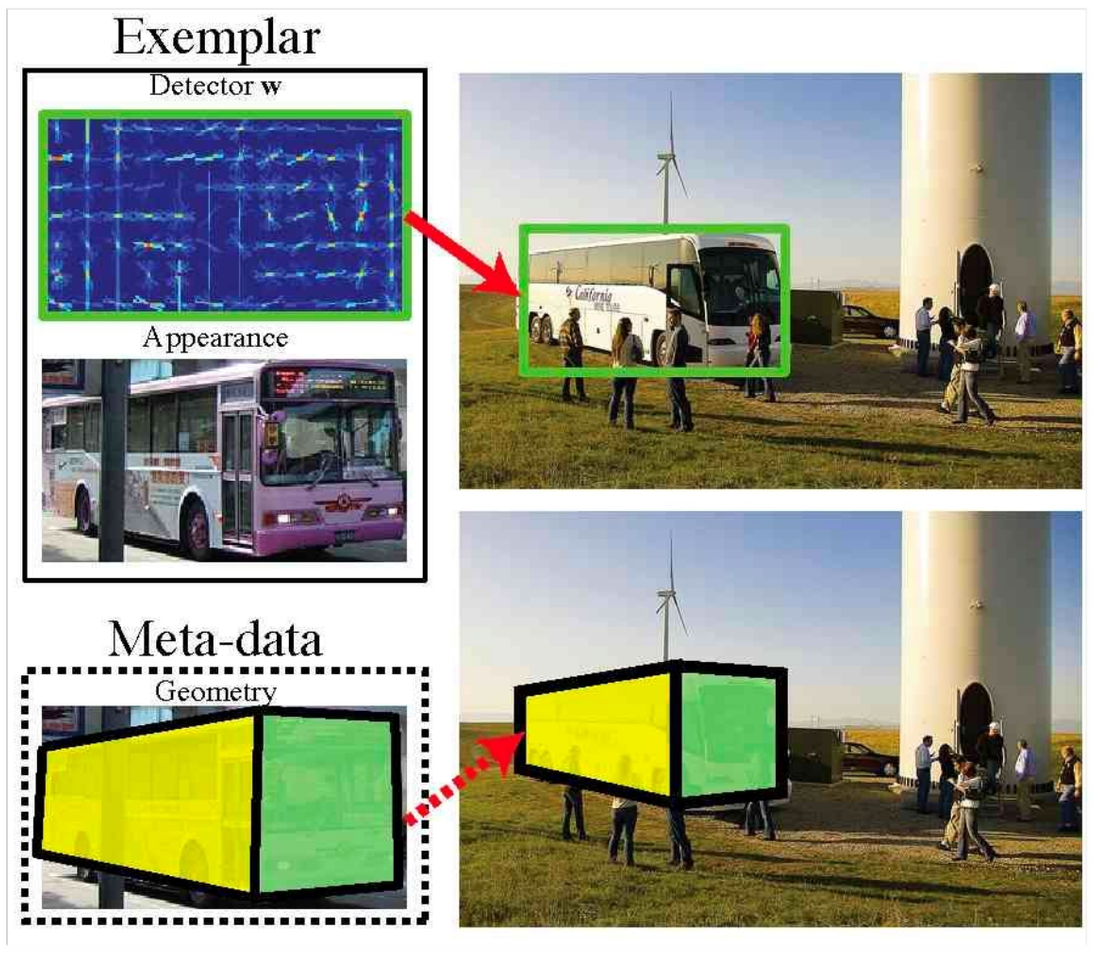
Intuition: NN (with infinite data) returns back maximum of posterior p(y=1|x) or p(y=0|x). Worst case is posterior = .5.

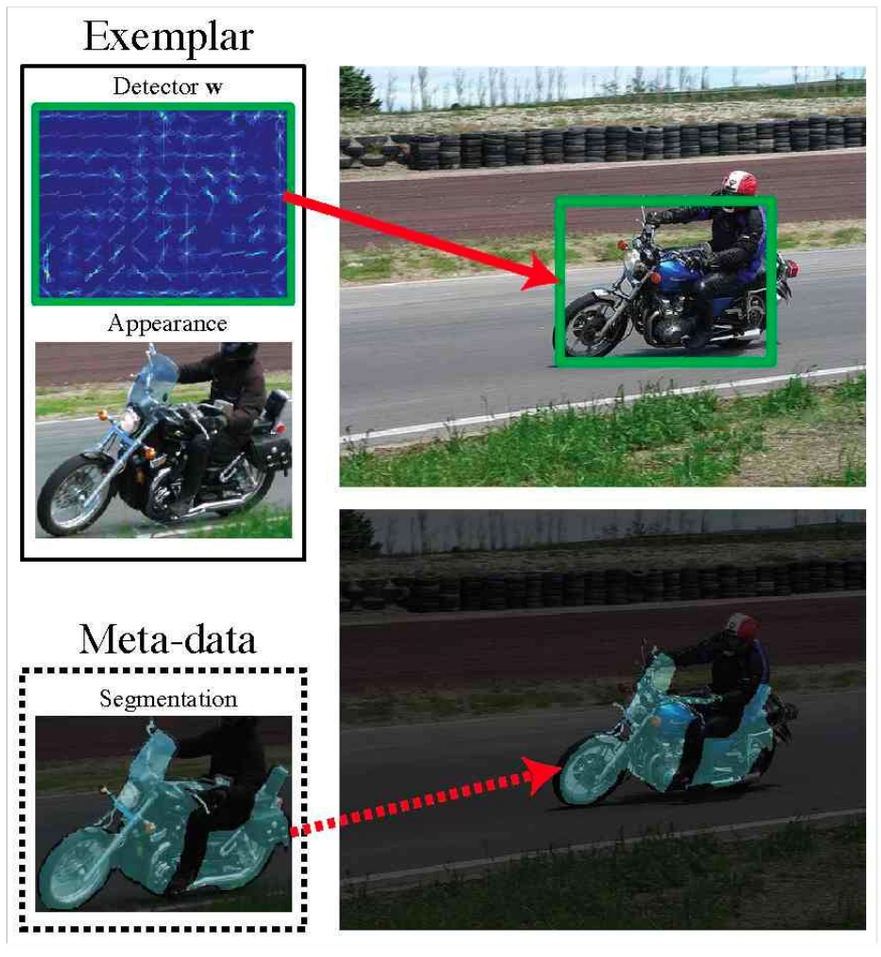
Classification “by retrieval”

$$\text{Prediction}(x) = y_{i*} \quad \text{where} \quad i^* = \arg\min_i ||x - x_i||^2$$



Ask not what is this, but “what is this like”?



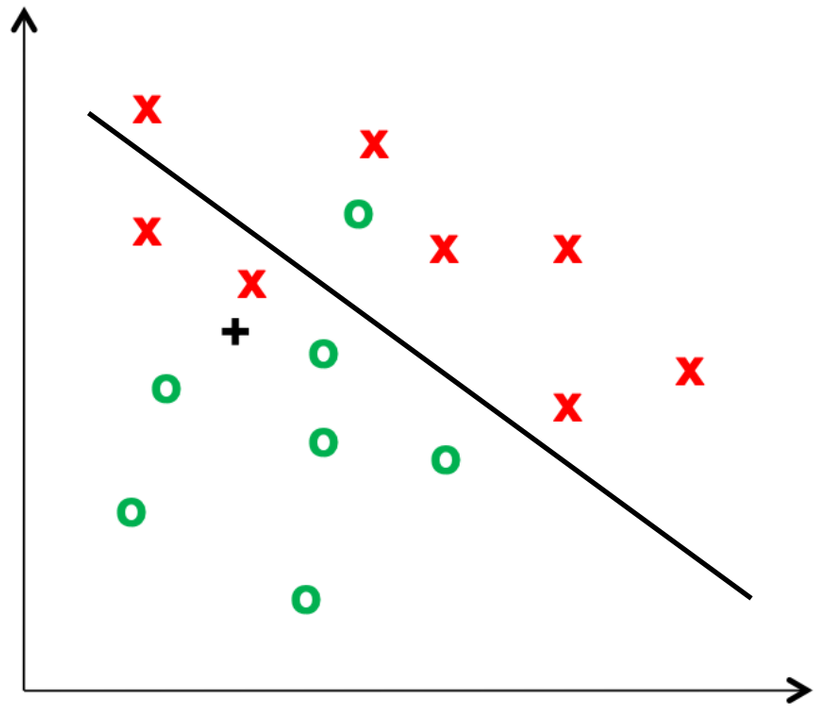


(in my view) this is the heart of all data-driven learning

Statistical classification

$x_i \in R^N$

$y_i \in \{-1,1\}$



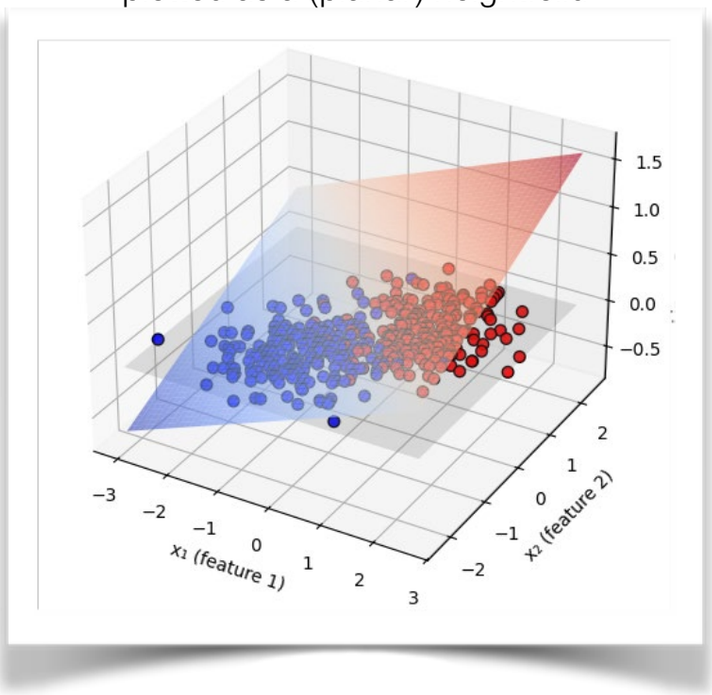
Given training points (x_i,y_i), learn function f(x) that predicts a label {-1,1}

Version 0: nearest neighbor classification

Version 1: linear classification (we'll spend rest of lecture on)

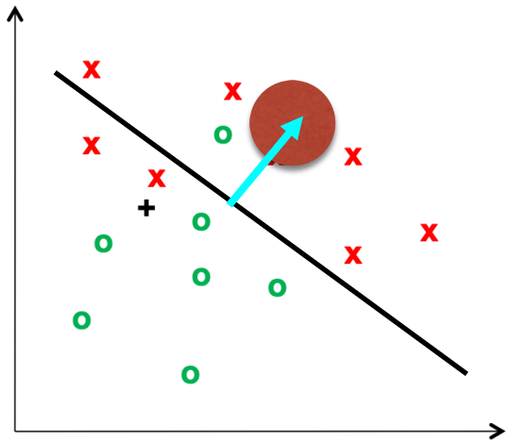
Building intuition for linear classification

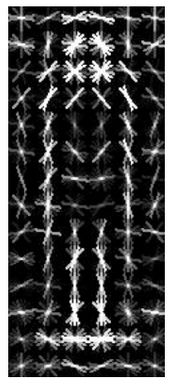
Score for each 2D input can be plotted as a (planar) heightfeild



$x_i \in R^N$

$y_i \in \{-1,1\}$





μ

Compute class centroid

$$\mu = \frac{1}{|P|}\sum_i x_i$$

Score the “person-ness” of example ‘x’ by computing its distance to centroid

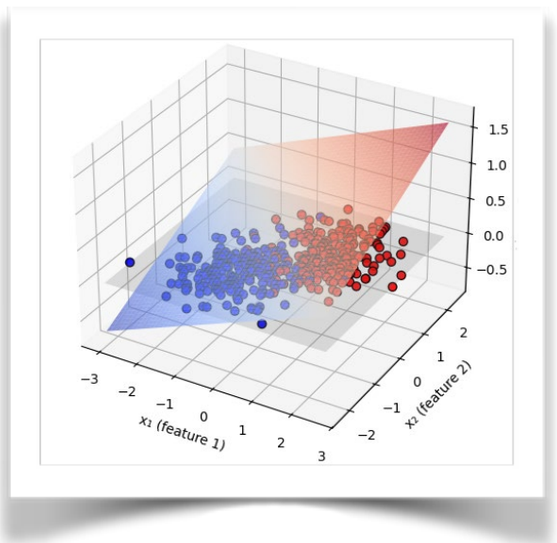
$$\frac{1}{2}||x-\mu||^2 = \frac{1}{2}x^T x - x^T\mu + \frac{1}{2}\mu^T\mu \leq \text{threshold}$$

Given normalized x features ($x^T x = 1$), dot-product should give equivalent rankings.

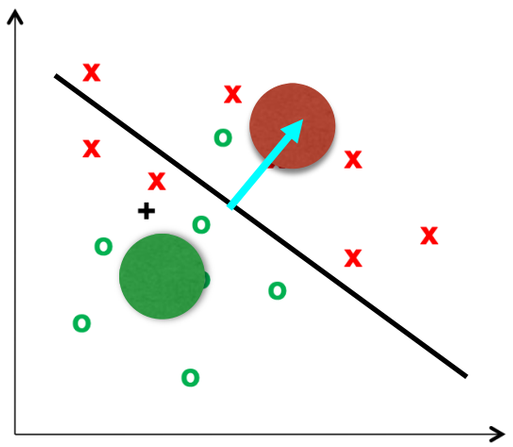
$$w^T x > threshold, w = \mu$$

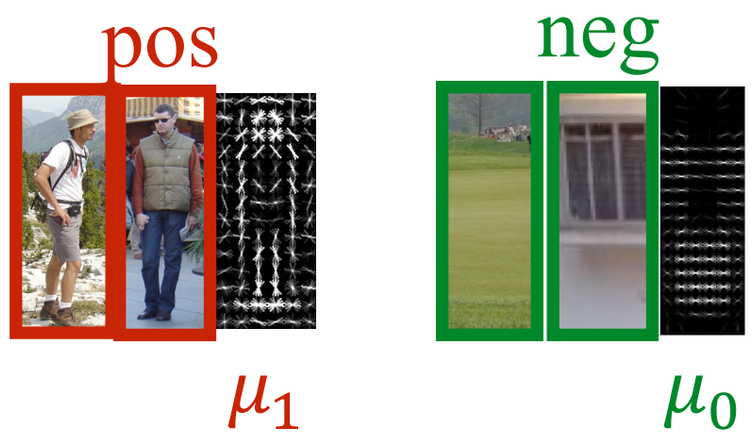
[ignore constants (for now) since they can be absorbed into threshold]

Building intuition for linear classification



$$x_i \in R^N$$
$$y_i \in \{-1,1\}$$





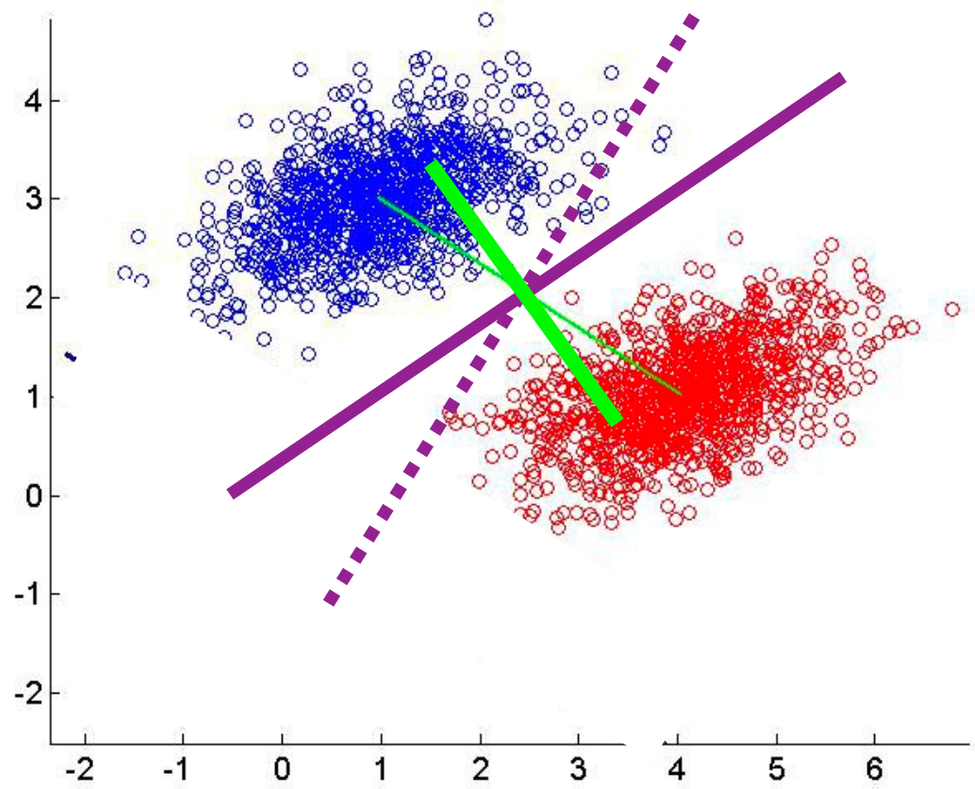
Score distance-to-person-centroid ***relative*** to distance-to-bg-centroid

$$\frac{1}{2}(||x-\mu_1||^2 - ||x-\mu_0||^2) = -x^T\mu_1 + \frac{1}{2}\mu_1^T\mu_1 + x^T\mu_0 - \frac{1}{2}\mu_0^T\mu_0$$
$$= -x^T(\mu_1-\mu_0) + \frac{1}{2}(\mu_1^T\mu_1 - \mu_0^T\mu_0)$$

$$w^Tx + b \geq 0 \qquad w = \mu_1 - \mu_0, \quad b = -\frac{1}{2}(\mu_1^T\mu_1 - \mu_0^T\mu_0)$$

[b compensates for skewed dot products arising from one class centroid being larger than the other; b=0 if both centroids have same norm]

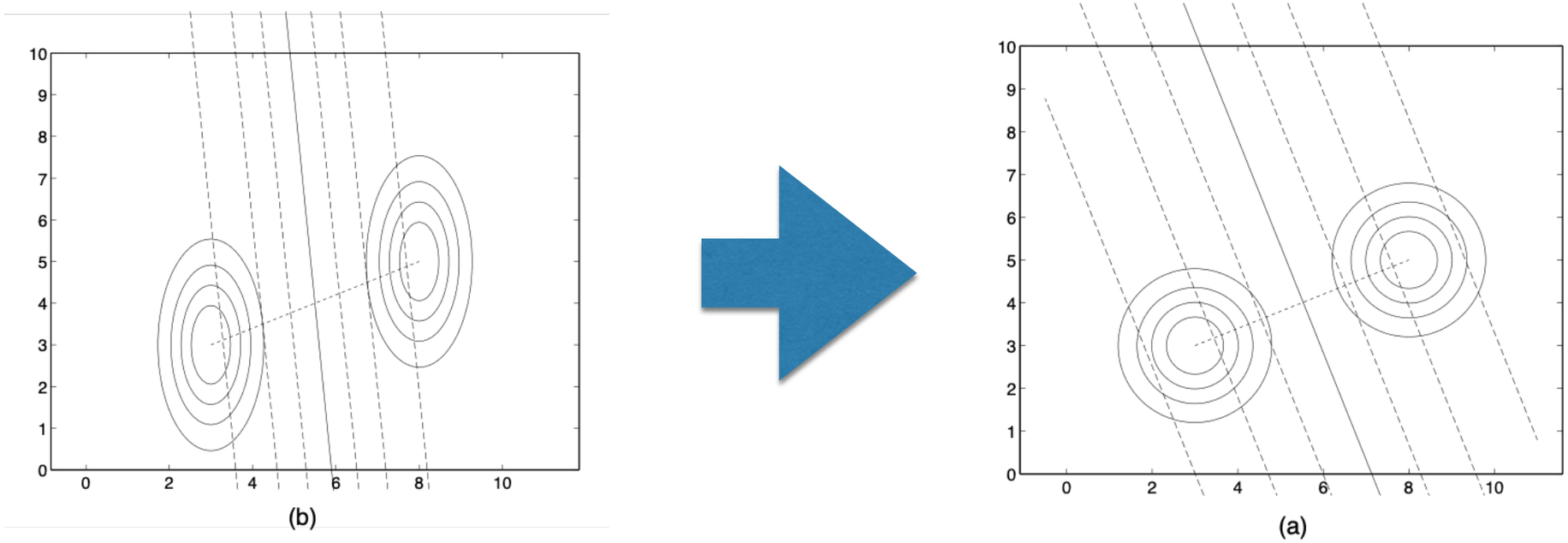
Skewed class distributions



If data is skewed along particular dimensions, we should measure centroid distances (and construct the resulting decision boundary) differently

How do we account for skew?

Skewed class distributions



Conceptually, let's apply a geometric warp to R^D that ***whitens*** (or deskews) the data…

$$\Sigma = \frac{1}{N}\sum_i^N \tilde{x}_i \tilde{x}_i^T \; where \; \tilde{x}_i = x_i - \mu_{y_i} \quad \text{(centered pts)}$$

Can either compute centroids in whitened space or apply Σ transformation to weight vector (both produce same classifier!)

$$x' \rightarrow \Sigma^{-\frac{1}{2}} x$$

$$(\mu_1 - \mu_0)^T x' \geq threshold$$

$$\Longleftrightarrow$$

$$w = \Sigma^{-1}(\mu_1 - \mu_0)$$

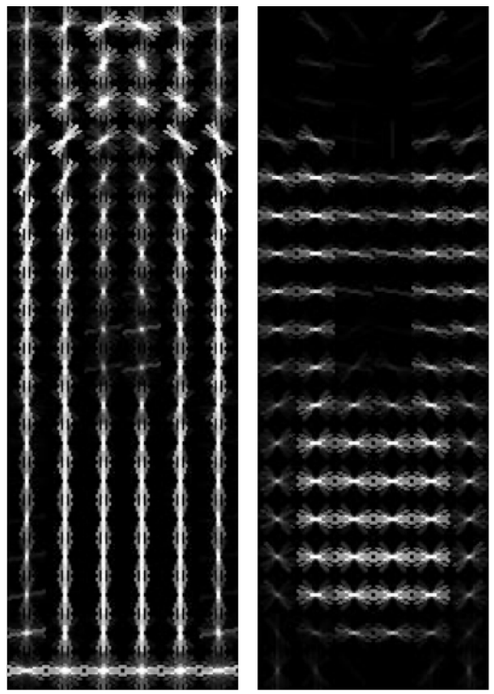
$$[\Sigma^{-1}(\mu_1 - \mu_0)]]^T x \geq threshold$$

Excercise for reader: computing Σ on warped data {x'} will yeild a Identity matrix

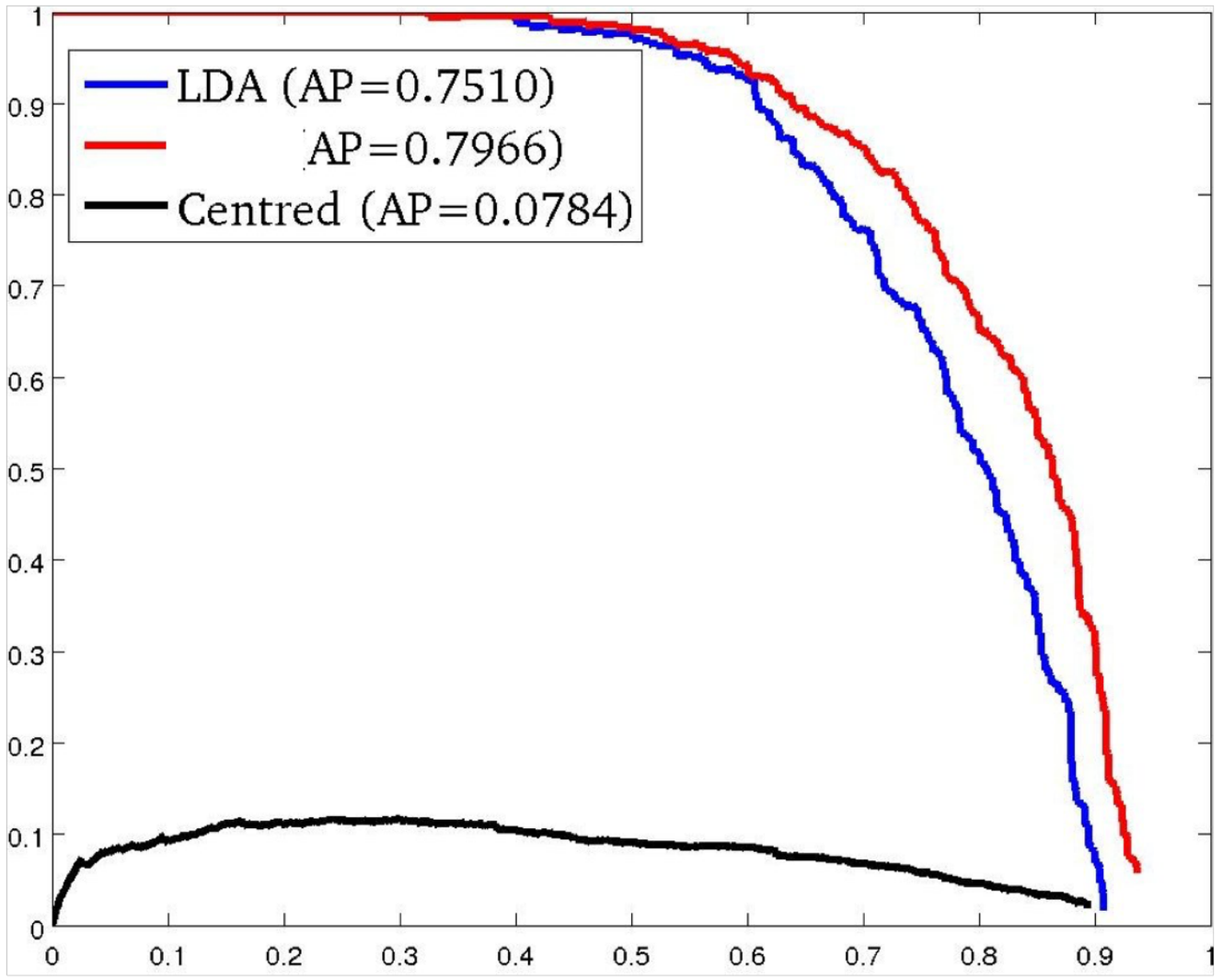
This technique is known as linear discriminant analysis (LDA) or Fischer Discriminant Analysis (FDA)
[though its typically derived from a variance minimization perspective which ignores the bias]

The numbers

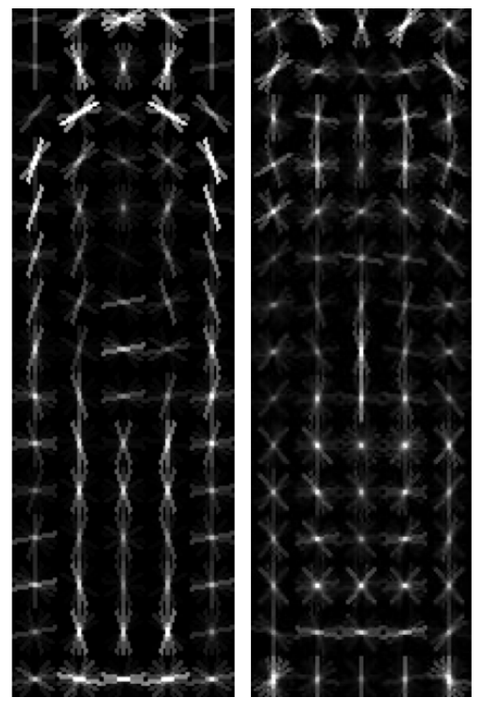
Centered



$w = \mu_1 - \mu_0$



Whitened LDA



$w = \Sigma^{-1}(\mu_1 - \mu_0)$

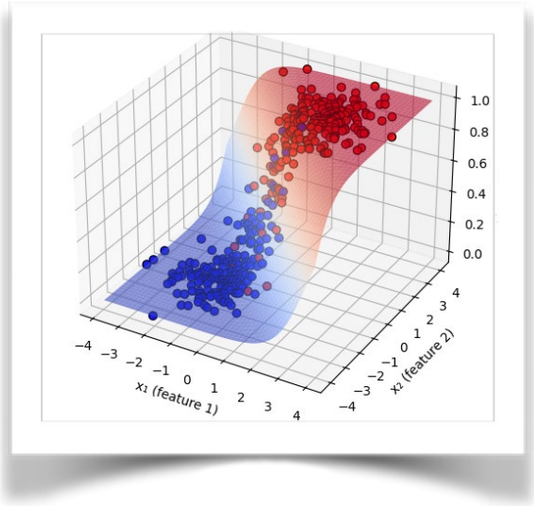
(a) AP

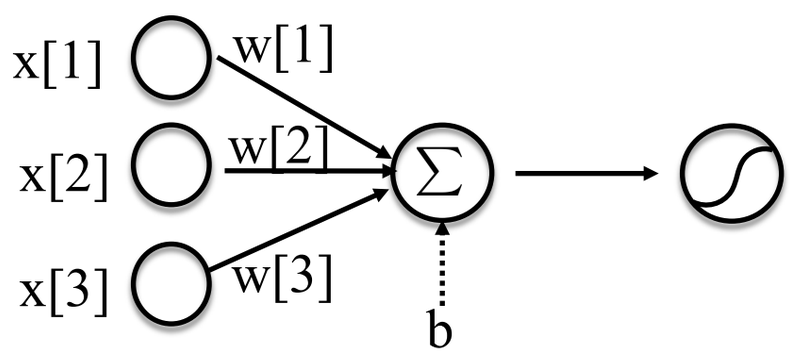
Whitening (or decorrelating) gets us 90% of the way there…

[foreshadowing of red curve: train with logistic regression]

Class-conditional Gaussians

$$p(y=1|x) = \text{sigmoid}(w^T x + b)$$

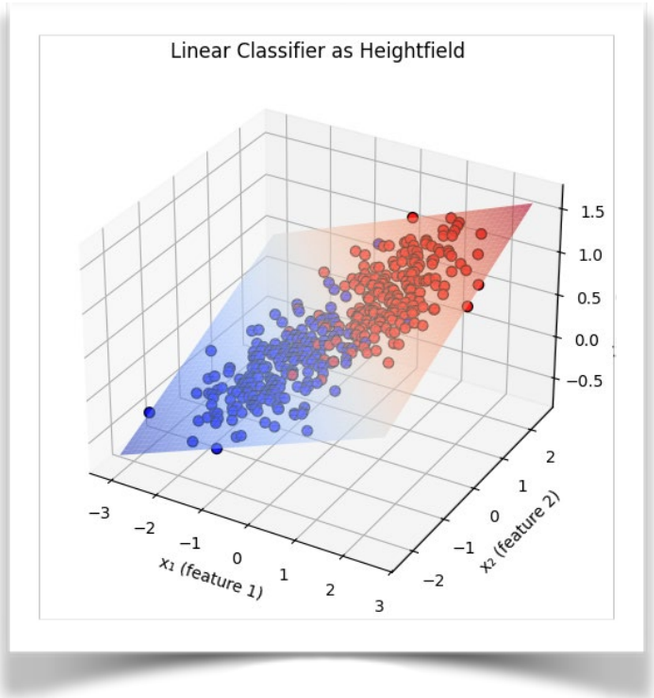




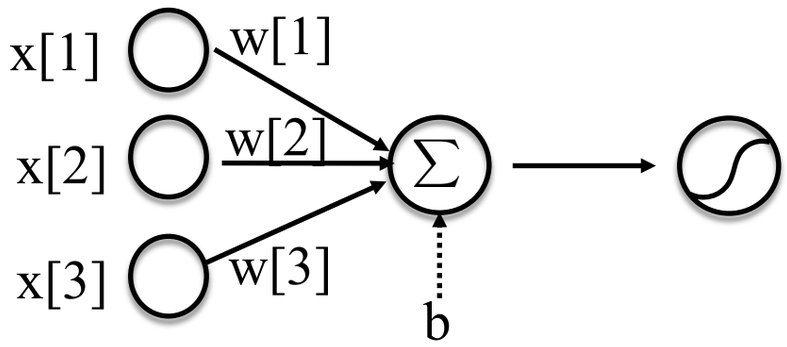
$$p(y=1|x) > .5 \quad \text{when} \quad \text{when} \quad w^T x + b > 0, \quad w = \Sigma^{-1}(\mu_1 - \mu_0)$$

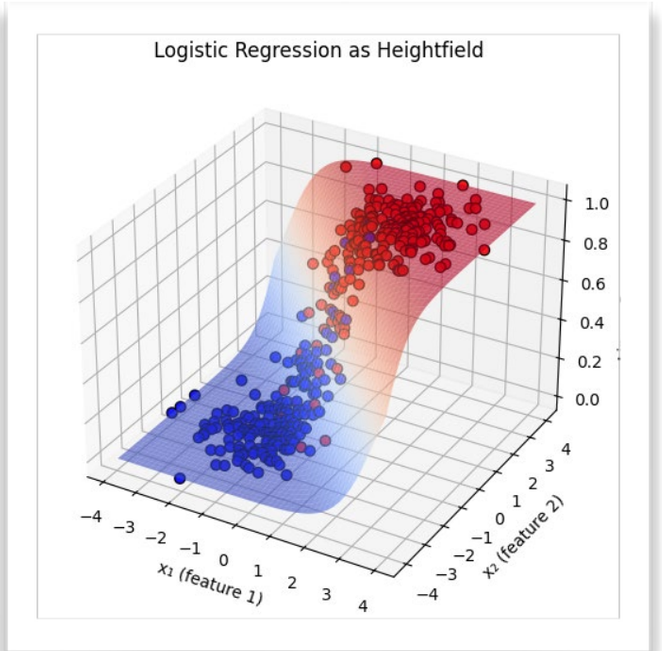
Core building block of all contemporary networks

Alternative: instead of fitting generative model (μ, Σ, θ), directly tune parameters (w,b) to minimize classification loss



$$x_i \in R^N$$
$$y_i \in \{0,1\}$$





$$p(y = 1|x) = \text{sigmoid}(w^T x + b)$$

Search for parameters 'w,b' to maximize log probability of the data $\{(x_i,y_i)\}$

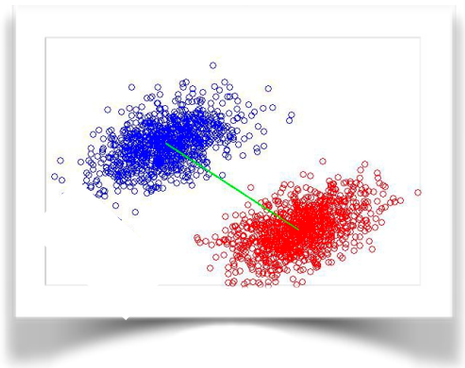
$$\max_{w,b} \sum_i \log p(y = y_i|x_i)$$

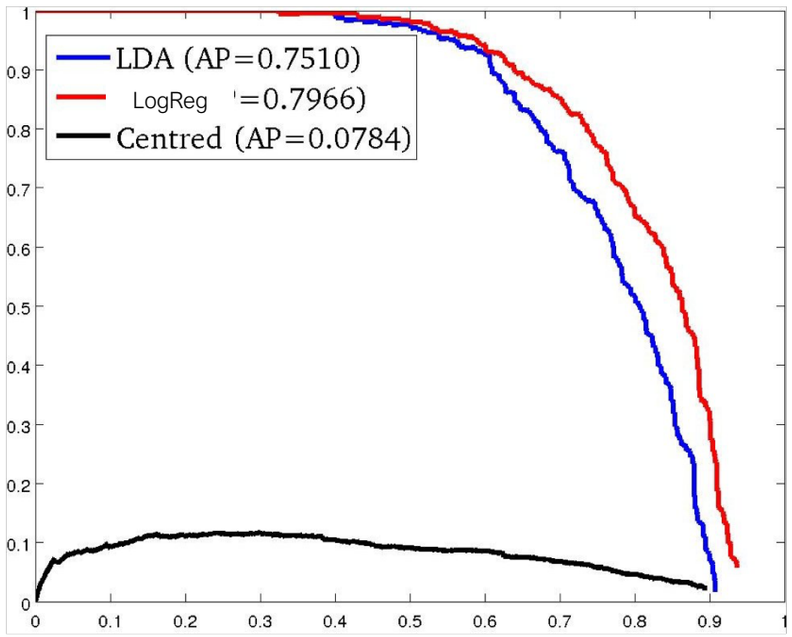
Gory detail: $\log p(y = y_i|x_i) = y_i \log(\hat{y}) + (1 - y_i)\log(1 - \hat{y})\, where \hat{y} = \frac{1}{1 + e^{-(w^T x_i + b)}}$

Lots of other names: **binary cross-entropy loss**, maximum likelihood, negative log-liklihood (NLL), log-loss, logistic regression

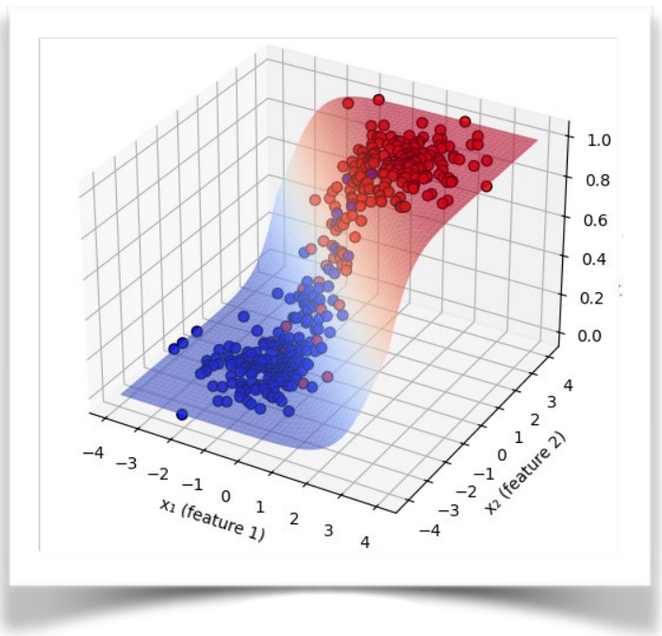
This perspective of loss minimization is the heart of all contemporary machine learning
(though I still like the generative bit; particularly
with recent innovations in generative modeling!)

Class Centroids vs LDA vs Logistic Regression

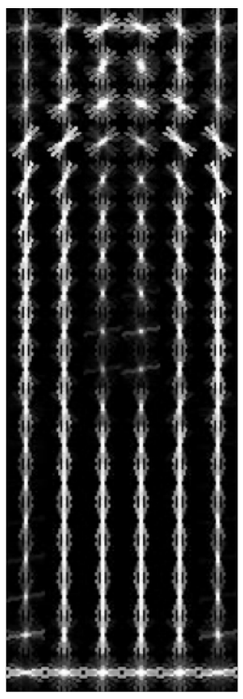




(a) AP

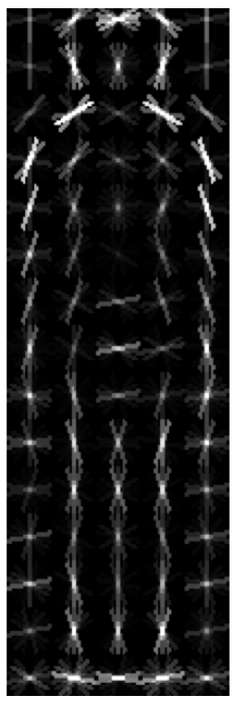


Centered



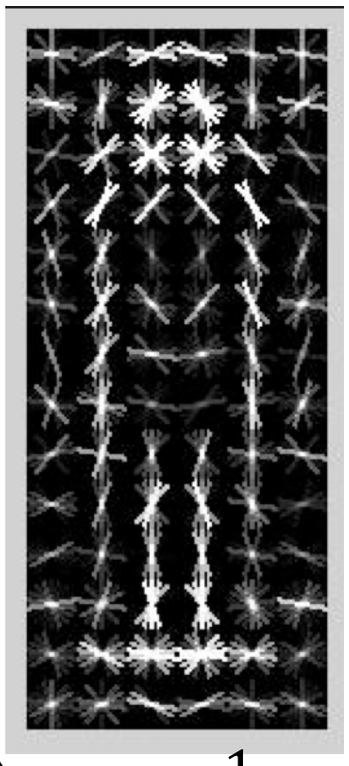
$$w = \mu_1 - \mu_0$$

LDA



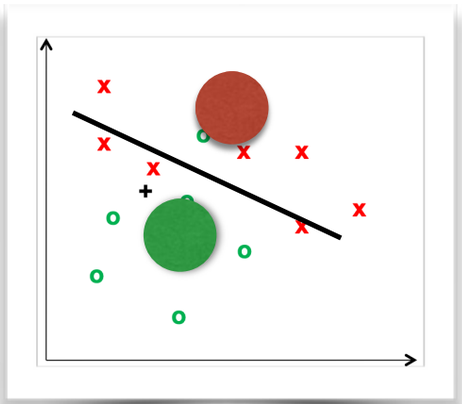
$$w = \Sigma^{-1}(\mu_1 - \mu_0)$$

LogReg



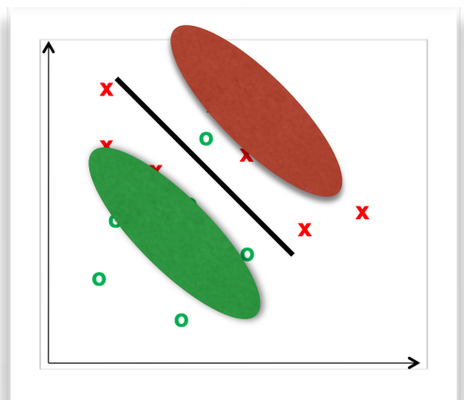
$$\hat{y} = \frac{1}{1 + e^{-(w^T x)}}$$

Look back at linear classification



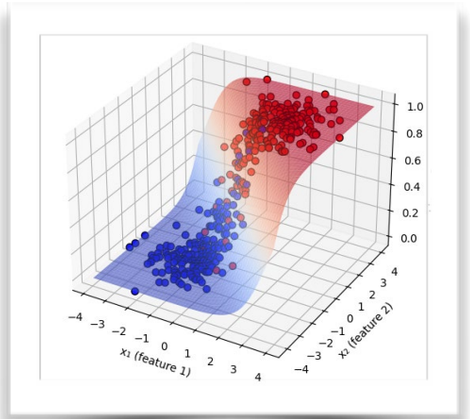
$$w = \mu_1 - \mu_0$$

Pay attention to centroids



$$w = \Sigma^{-1}(\mu_1 - \mu_0)$$

Pay attention to correlations



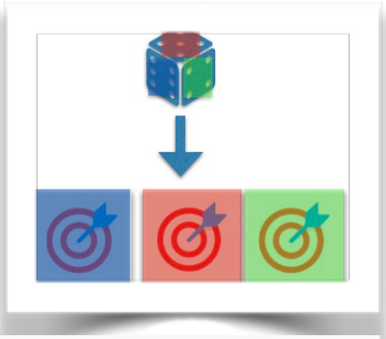
$$\hat{y} = \frac{1}{1 + e^{-(w^T x)}}$$

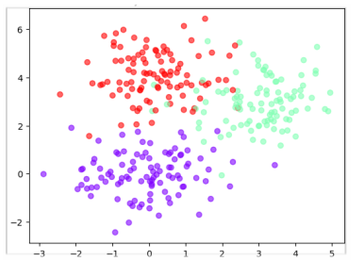
Pay attention to
points near boundary

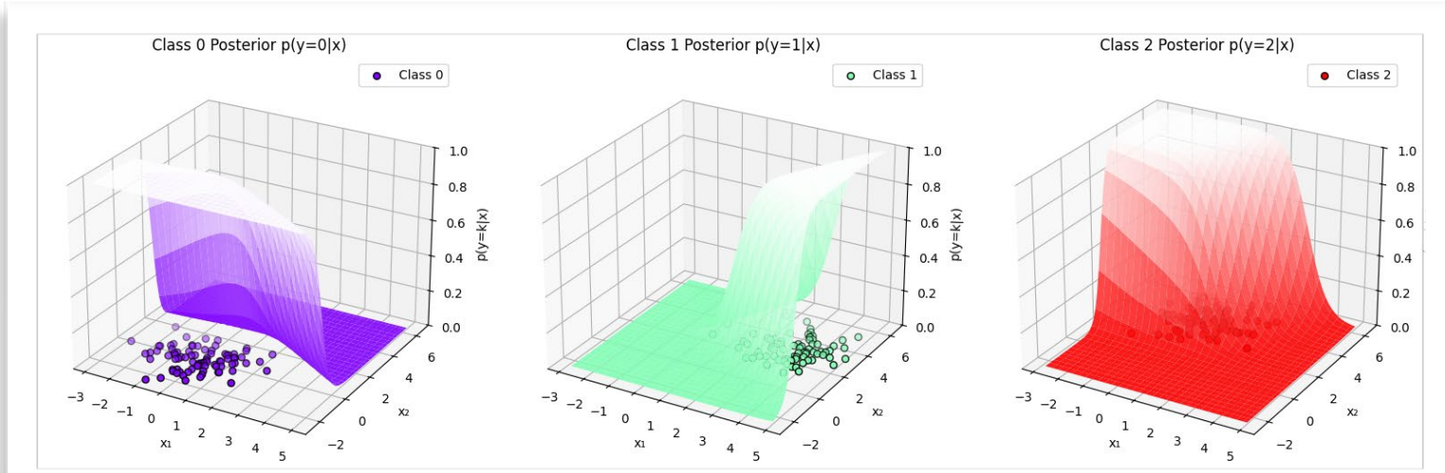
Softmax: extending logistic regression to multiple classes

Same derivation of class-conditional Gaussians for K classes (assuming all have same covariance)

```
n_samples = 300
priors = [0.2, 0.5, 0.3]
means = [
    np.array([0, 0]),
    np.array([4, 4]),
    np.array([0, 5])
]

# Store linear transform A such that
# the covariance matrix Sigma = A@A.^T
A = np.array([[1.0, 0.3], [0.3, 1.0]])

# Containers
X = []
y = []

# --- Sample one point at a time ---
for _ in range(n_samples):
    # Sample class index from prior
    c = np.random.choice(len(priors), p=priors)

    # Sample z ~ N(0, I)
    z = np.random.randn(2)

    # Transform: x = A z + mu
    x = A @ z + means[c]

    X.append(x)
    y.append(c)
```







$$p(y = k|x) = softmax_k(z_k) = \frac{e^{z_k}}{e^{z_0} + e^{z_1} + e^{z_2}} \quad where z_k = w_k^T x + b_k$$

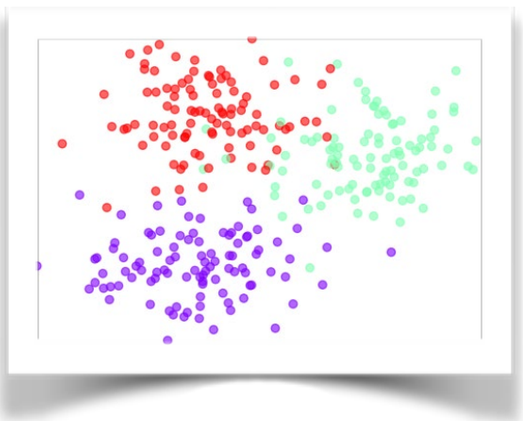
Kx1 vector of ***logits*** Kx1 vector

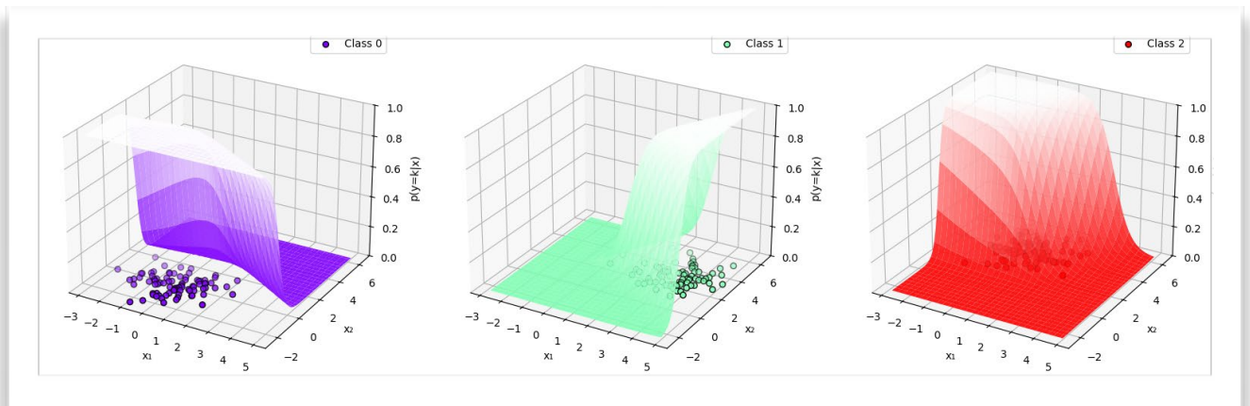
$$z = Wx + b$$

KxD matrix Dx1 feature

Cross-entropy loss: fit parameters (W,b) to maximize log $p(y=y_i|x_i)$ over data pairs (x_i,y_i)

Implementing softmax





$$p(y = k|x) = \underset{\text{k}}{softmax}(z_k) = \frac{e^{z_k}}{e^{z_0} + e^{z_1} + e^{z_2}}$$

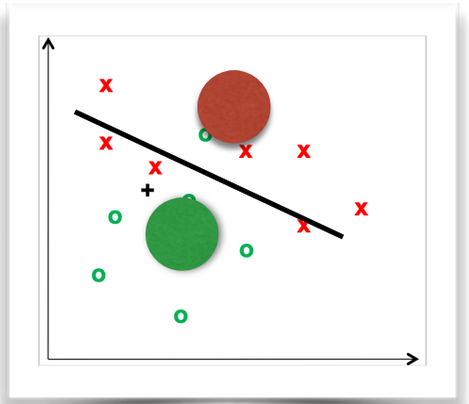
Note there's a degree of freedom (DOF):

$$\underset{\text{k}}{softmax}(z_k) = \frac{e^{z_k}}{e^{z_0} + e^{z_1} + e^{z_2}} = \underset{\text{k}}{softmax}(z_k - \alpha) = \frac{e^{z_k - \alpha}}{e^{z_0 - \alpha} + e^{z_1 - \alpha} + e^{z_2 - \alpha}}$$

Exploit DOF to improve numerical stability; choose α such that largest term in denominator becomes 1

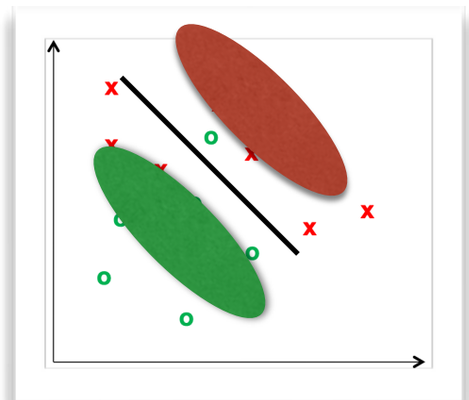
```
# --- Softmax Regression ---
def softmax(z):
    e_z = np.exp(z - np.max(z))
    return e_z / np.sum(e_z)
```

Look back at linear classification



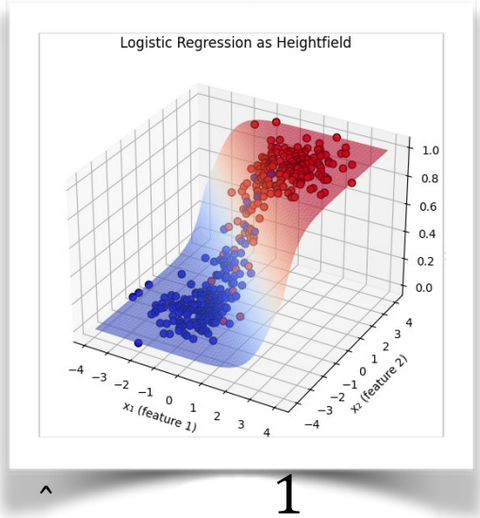
$$w = \mu_1 - \mu_0$$

Pay attention to centroids



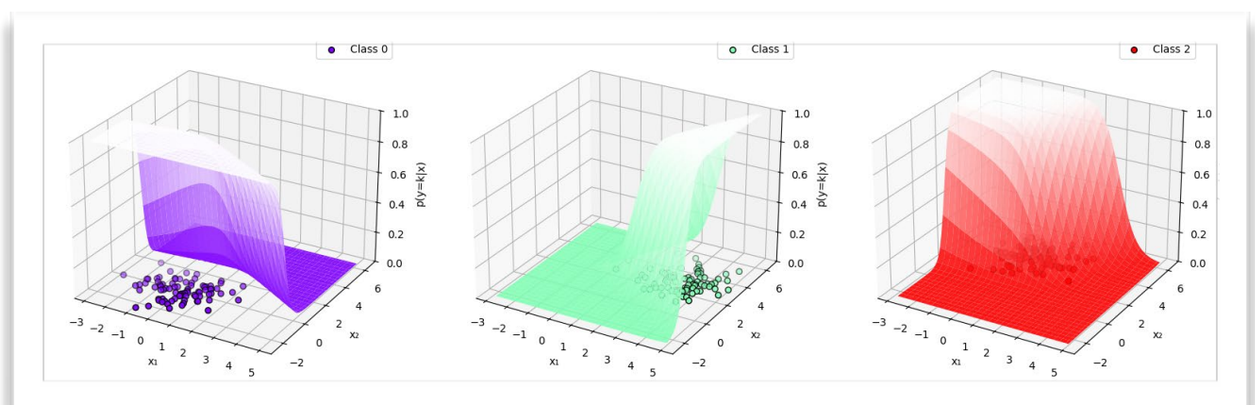
$$w = \Sigma^{-1}(\mu_1 - \mu_0)$$

Pay attention to correlations



$$\hat{y} = \frac{1}{1 + e^{-(w^T x)}}$$

Pay attention to points near boundary



$$\hat{y} = \frac{e^{z_k}}{e^{z_0} + e^{z_1} + e^{z_2}}, z = Wx$$

Modular view of ML

$$\min_w \lambda R(w) + \sum_i loss(y_i, f_w(x_i))$$

$$R_{L2}(w) = \frac{1}{2}||w||^2 \quad \text{(L2 regularization)}$$

$$R_{L1}(w) = \sum_j |w_j| \quad \text{(L1/sparse regularization)}$$

$$loss_{01}(y, \hat{y}) = I(y \neq \hat{y}) \quad \text{0-1 loss}$$

$$loss_{squared}(y, \hat{y}) = (y - \hat{y})^2 \quad \text{squared loss}$$

$$loss_{\text{cross\_entropy}}(y, \hat{y}) = \sum_k y[k] \log p(\hat{y} = k) \quad \text{cross-entropy loss}$$

$$f_w(x_i) = w^T x_i + b \quad \text{Linear classifier}$$

$$f_w(x_i) = CNN_w(x_i) \quad \text{Nonlinear classifier}$$

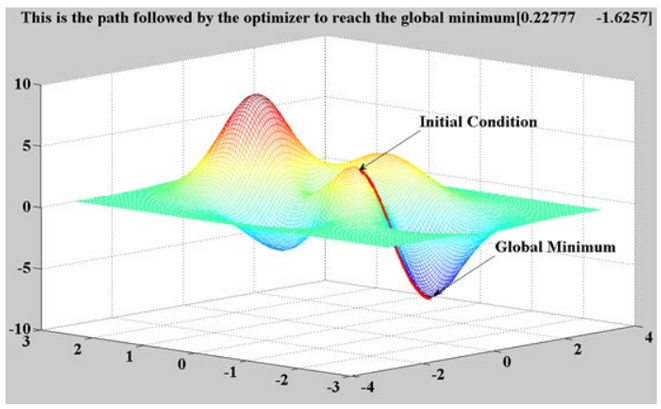
Optimizer =**(stochastic) gradient descent**
=least squares

Workhorse optimizer: gradient descent

$$\min_{w} L(w) \text{ where } \quad L(w) = \frac{1}{2}||w||^2 + \frac{1}{2}\sum_i (f_w(x_i) - y_i)^2$$

$$w \mathrel{-}= learnRate * (w + \sum_i (f_w(x_i) - y_i)\frac{\partial f_w(x_i)}{\partial w})$$

Weight decay + loss error + model gradient



Early 2000's: obsession with convex $L(w)$ where gradient descent would find global minima

Turns out local minima isn't that big of a deal!

Real workhorse optimizer: stochastic gradient descent (SGD)

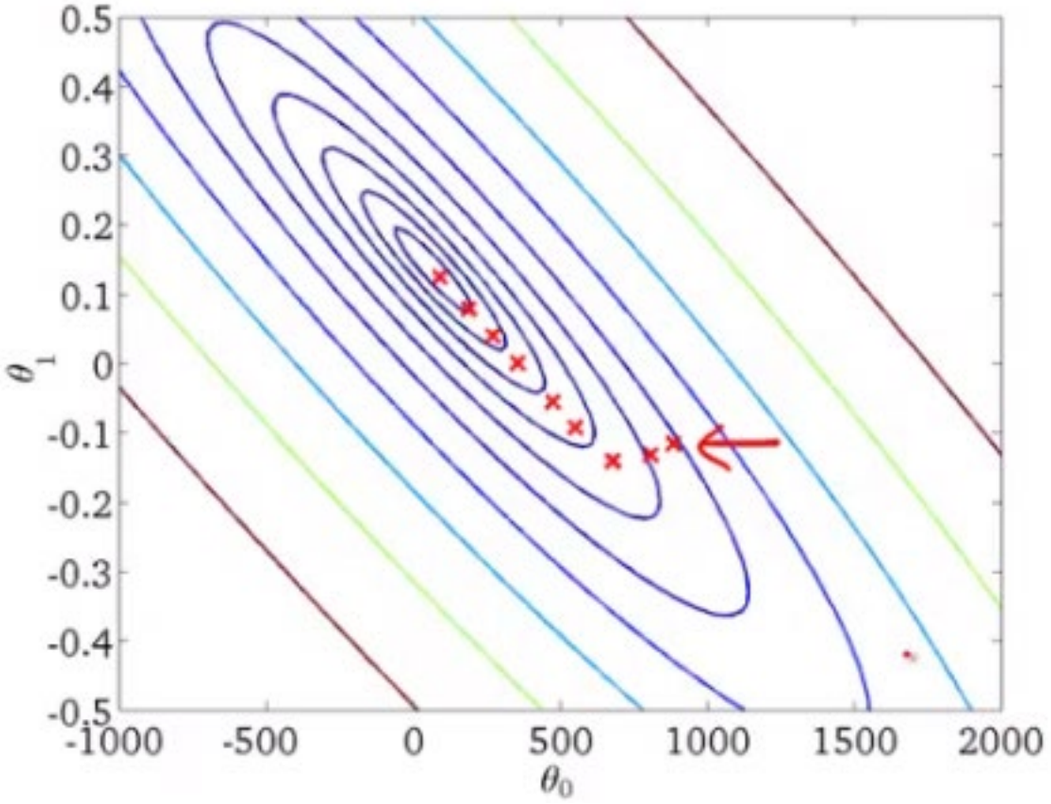
drop weight decay for notational simplicity

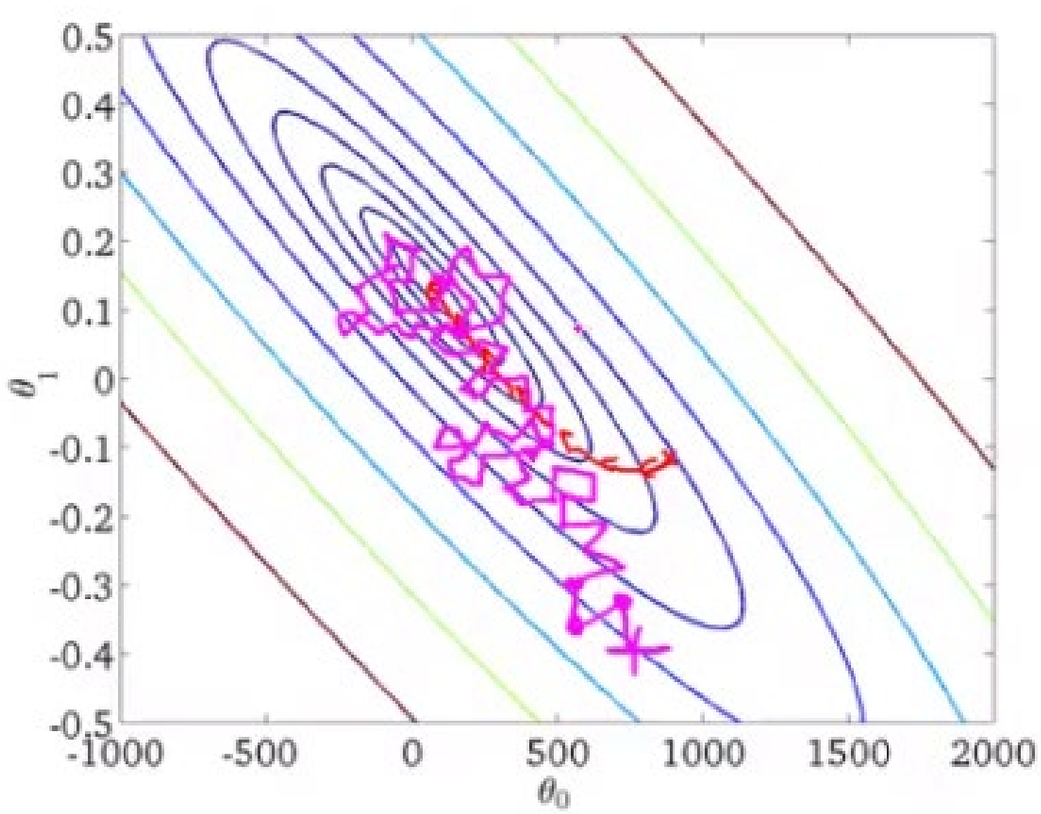
$$w \mathrel{-}= learnRate * (\sum_i (f_w(x_i) - y_i) \frac{\partial f_w(x_i)}{\partial w})$$

Real gradient requires summing over entire dataset

$$w \mathrel{-}= learnRate * (f_w(x_i) - y_i) \frac{\partial f_w(x_i)}{\partial w}$$

Stochastic gradient is computed over a single example



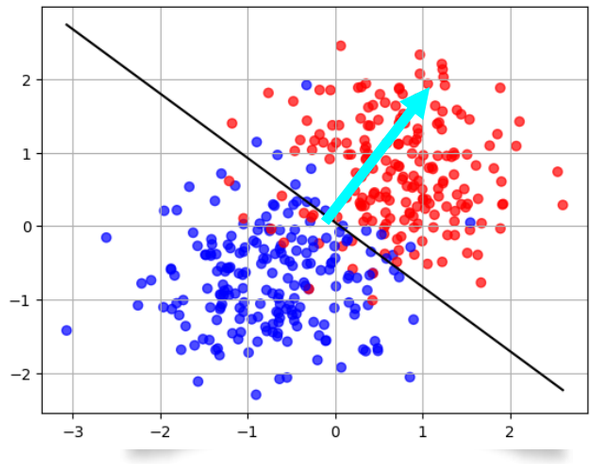


Trivially “out-of-core”: most contemporary models are trained in this manner

Important special cases of SGD:

linear regression (mean-squared error)

Linear Regression



$x_i \in R^D$

$y_i \in R$

$$L_i = \frac{1}{2}(\hat{y} - y_i)^2 \; where \; \hat{y} = w^T x_i$$

$$\frac{\partial L_i}{\partial w}$$

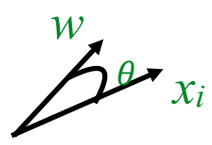
$$= (\hat{y} - y_i)x_i$$

$$= learnRate * (\hat{y} - y_i)x_i$$

$$w \mathrel{+}= learnRate * (y_i - y)x_i$$

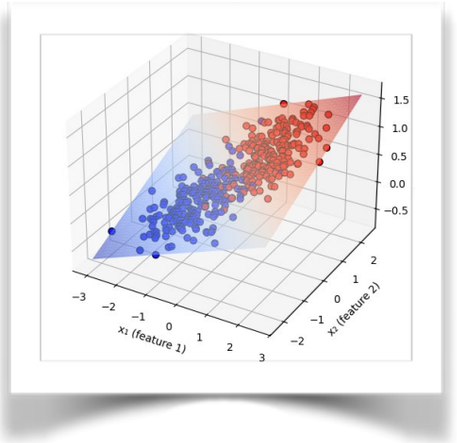
(if w *under*-predicts target y_i for sample x_i, *add* x_i to w)

$w^T x = ||w||||x_i||\cos\theta$



Important special cases of SGD:
linear regression (mean-squared error) & logistic regression (binary cross-entropy)

Linear Regression



$y_i \in R$

$\hat{y} \in R$

$$L_i = \frac{1}{2}(\hat{y} - y_i)^2 \; where \; \hat{y} = w^T x_i$$

$$\frac{\partial L_i}{\partial w}$$

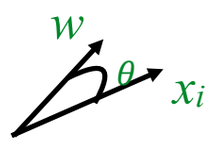
$$= (\hat{y} - y_i)x_i$$

$$= learnRate * (\hat{y} - y_i)x_i$$

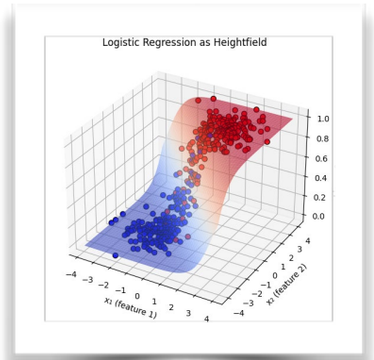
$$w \mathrel{+}= learnRate * (y_i - \hat{y})x_i$$

(if we ***under***-predict the target y_i for sample x_i, ***add*** sample x_i to w)

$w^T x = ||w||||x_i||\cos\theta$



Logistic Regression



$y_i \in \{0,1\}$

$\hat{y} \in [0,1]$

$$L_i = -\log p(y_i|x_i)$$

$$= -[y_i \log(\hat{y}) + (1 - y_i)\log(1 - \hat{y})] \; where \; \hat{y} = \frac{1}{1 + e^{-w^T x_i}}$$

$$\frac{\partial L_i}{\partial w}$$

$$= (\hat{y} - y_i)x_i$$

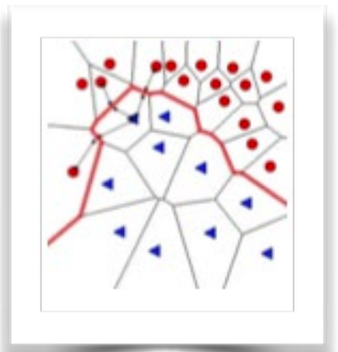
$$= learnRate * (\hat{y} - y_i)x_i$$

$$w \mathrel{+}= learnRate * (y_i - \hat{y})x_i$$

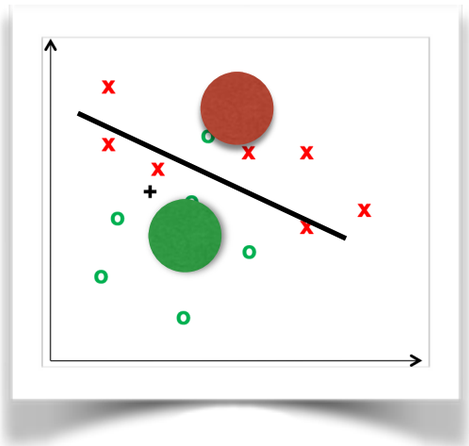
Key change: ***squash*** prediction to [0,1] since target lies in that range

(derivation on next slide)

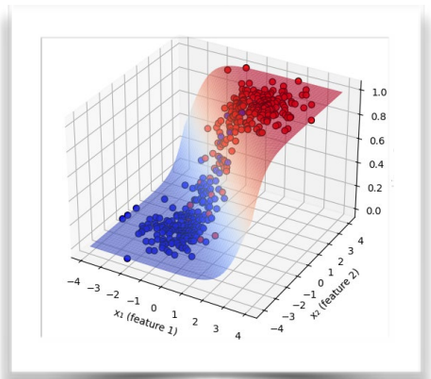
Parting thoughts on statistical classification



$$\min_i ||x - x_i||^2$$



$$w = \mu_1 - \mu_0$$



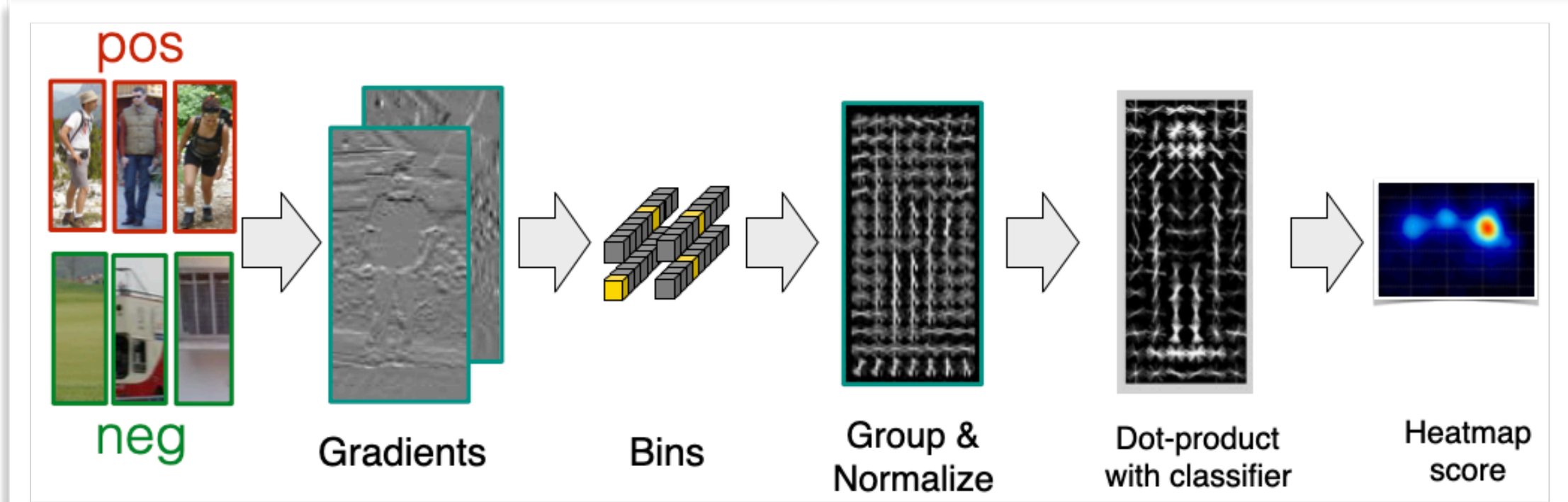
$$\max_w \sum_i y_i \log \hat{p}(y) \; where \hat{y} = \frac{1}{1 + e^{-(w^T x)}}$$

Classification: nearnest neighbors, (whitened) centroids, **sigmoids, softmax**

$$\min_w \lambda R(w) + \sum_i loss(y_i, f_w(x_i))$$

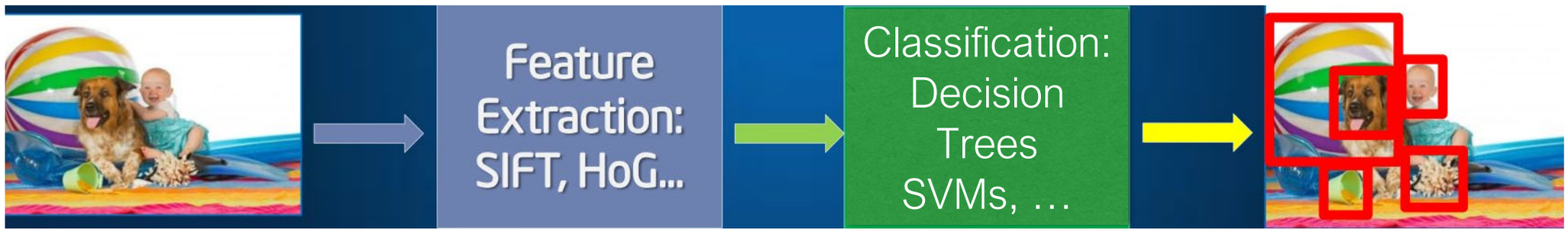
Modular ML: **(SGD) optimizers**, regularizers, loss functions, (non)linear predictors

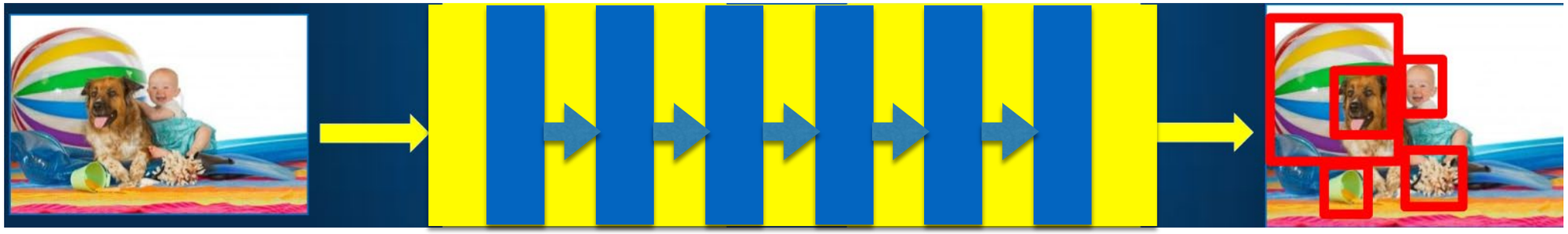
Deep down, we know that vision is really about *features*, not classifiers…



Deep representation learning: *learn* features rather than hand-design them

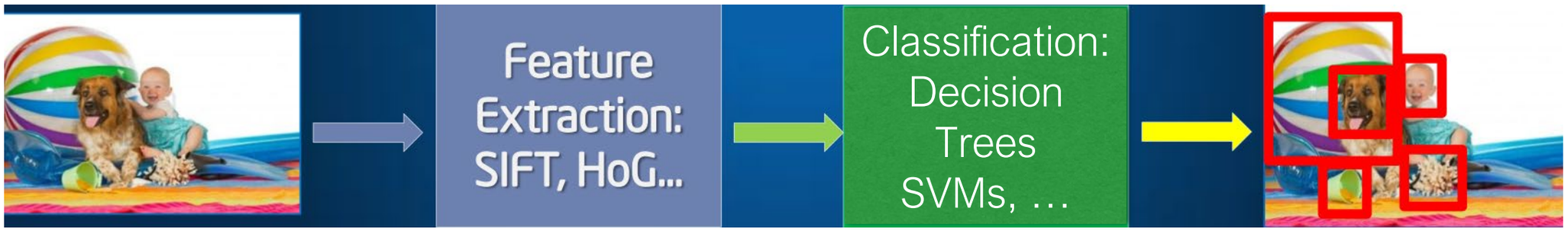
Transition from emphasis on classifiers to features or “representation”

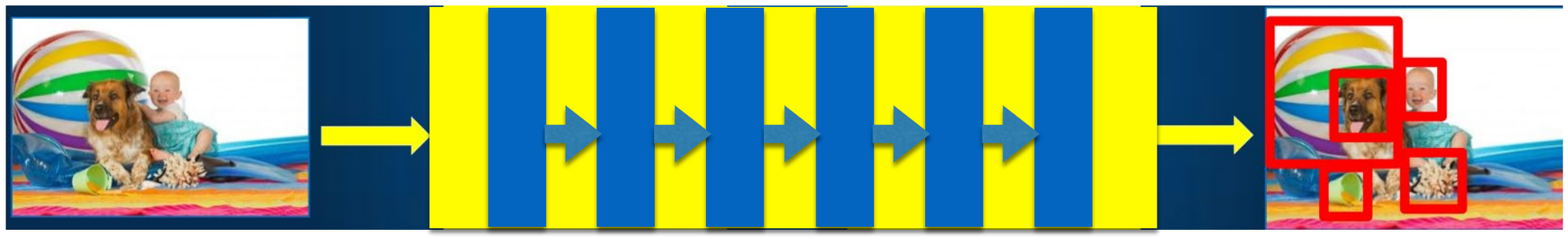




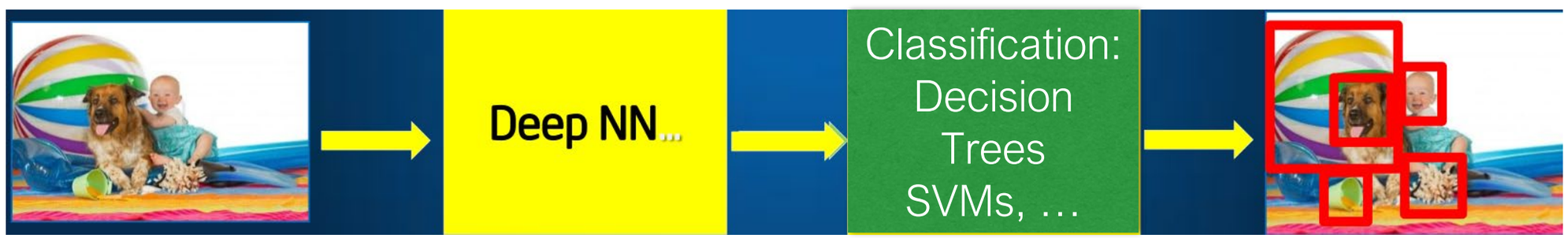
Deep network

Blurs the line between feature and classifier; is this a deep feature extractor with a linear classifier or a shallow feature with a nonlinear classifier?





End-to-end learning



“Off-the-shelf” deep features

My guess: a universal feature extractor is within reach (download DINOV3!)

Implies that “version0” of any deep learning soln ***shouldn’t do any deep learning***; download off-the-self deep feature extractor (Bert, ResNet, CLIP, etc…) and use classic classifiers on top!