

Intro to

Computer Vision

with Prof. Kosta Derpanis

Machine learning



$$\mathbf{x} \in \mathbb{R}^N$$



$m \times n$

—
—



$3mn \times 1$

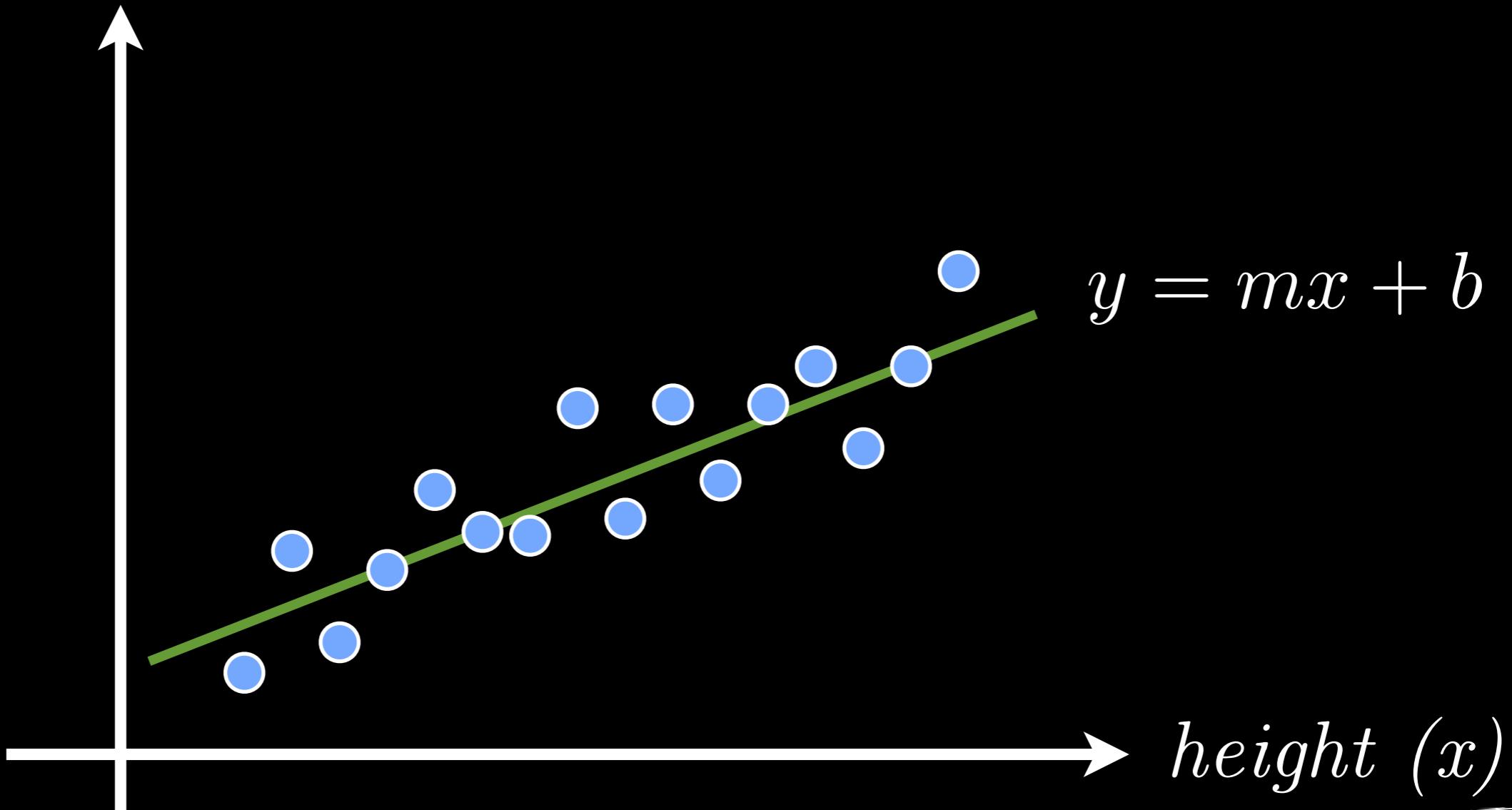
5

canonical learning problems

1

regression

weight (y)

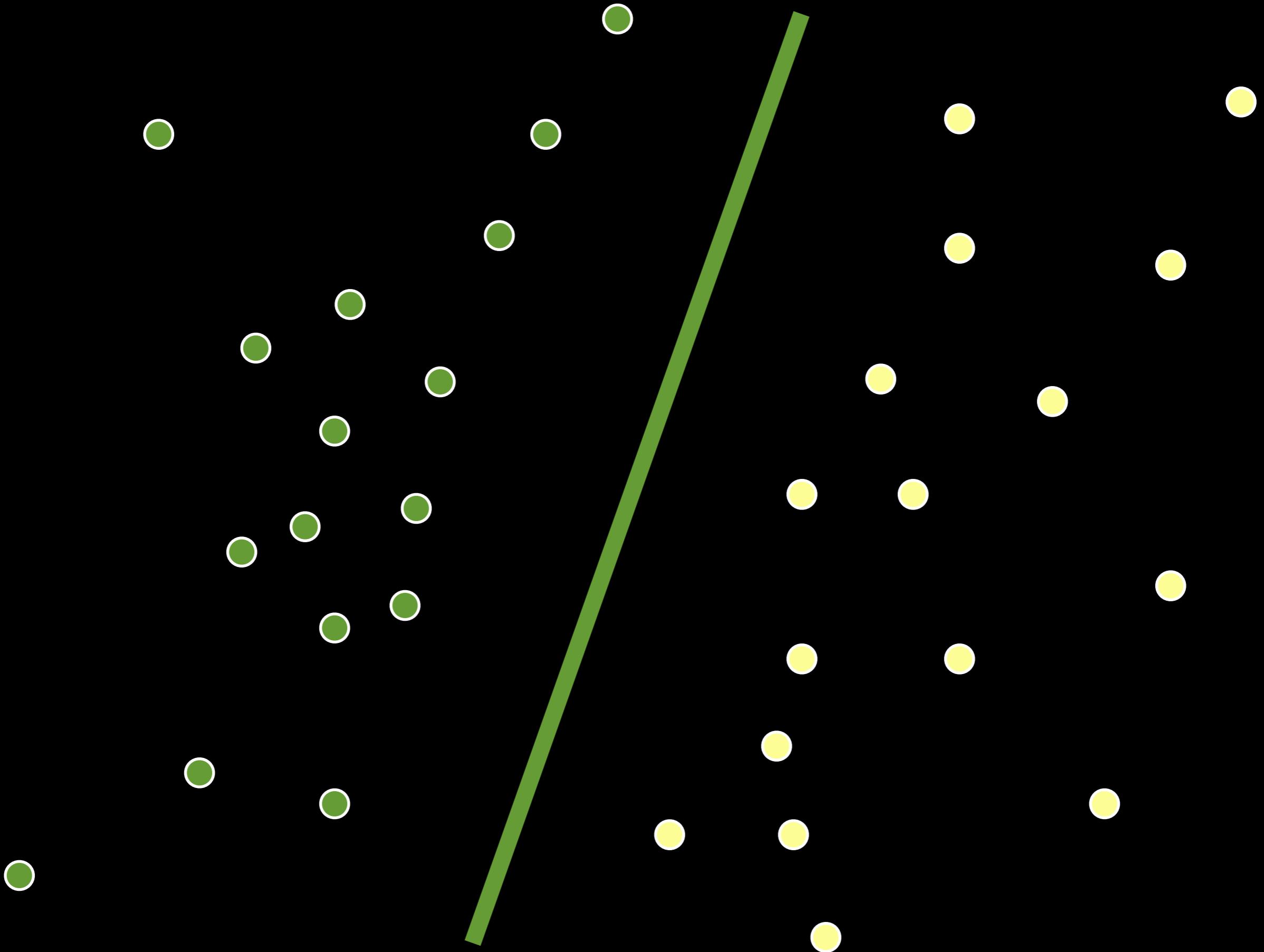


estimate the parameters of the “best” fitting function

2

classification

assign input to one of two or more discrete classes

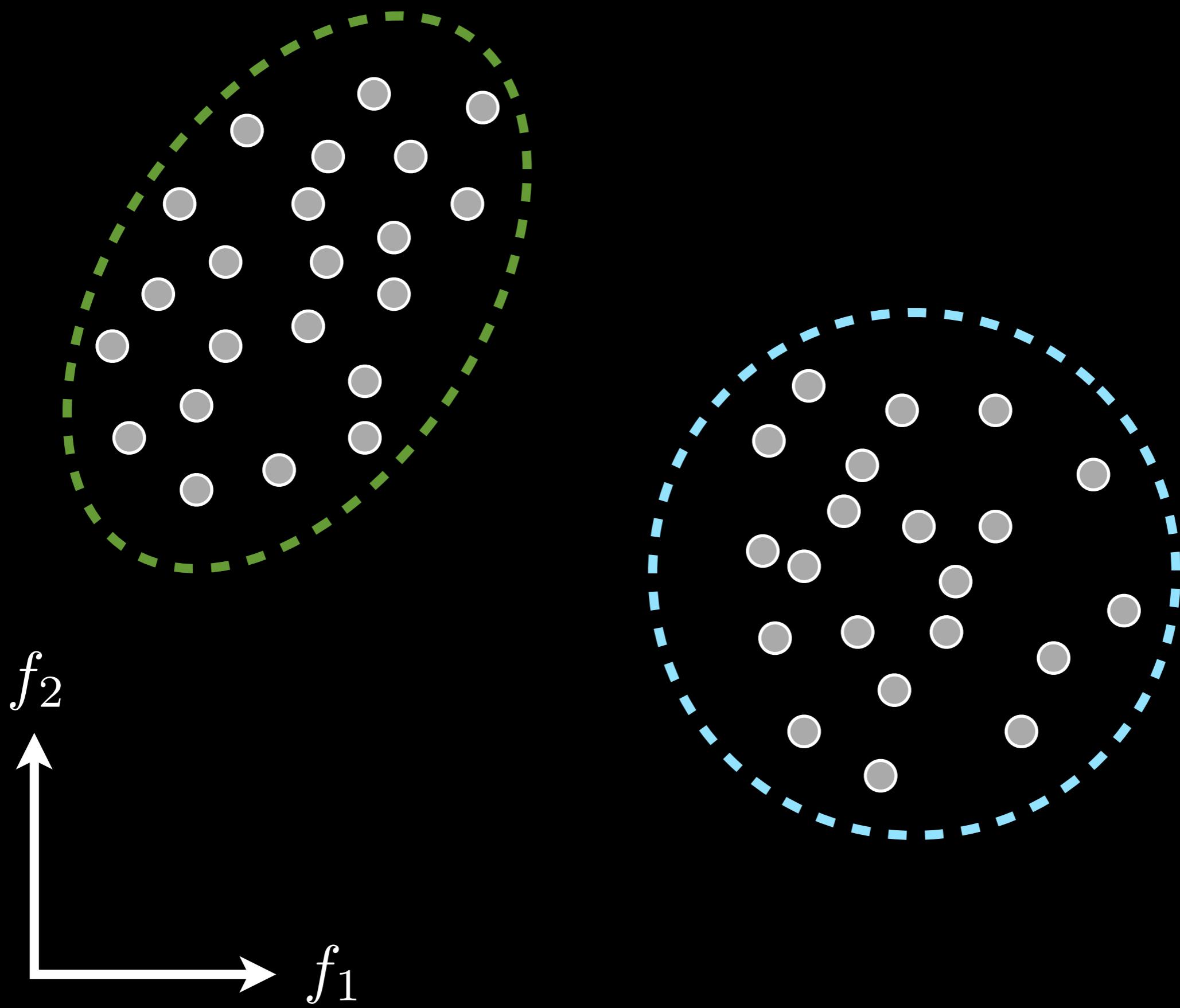




3

clustering

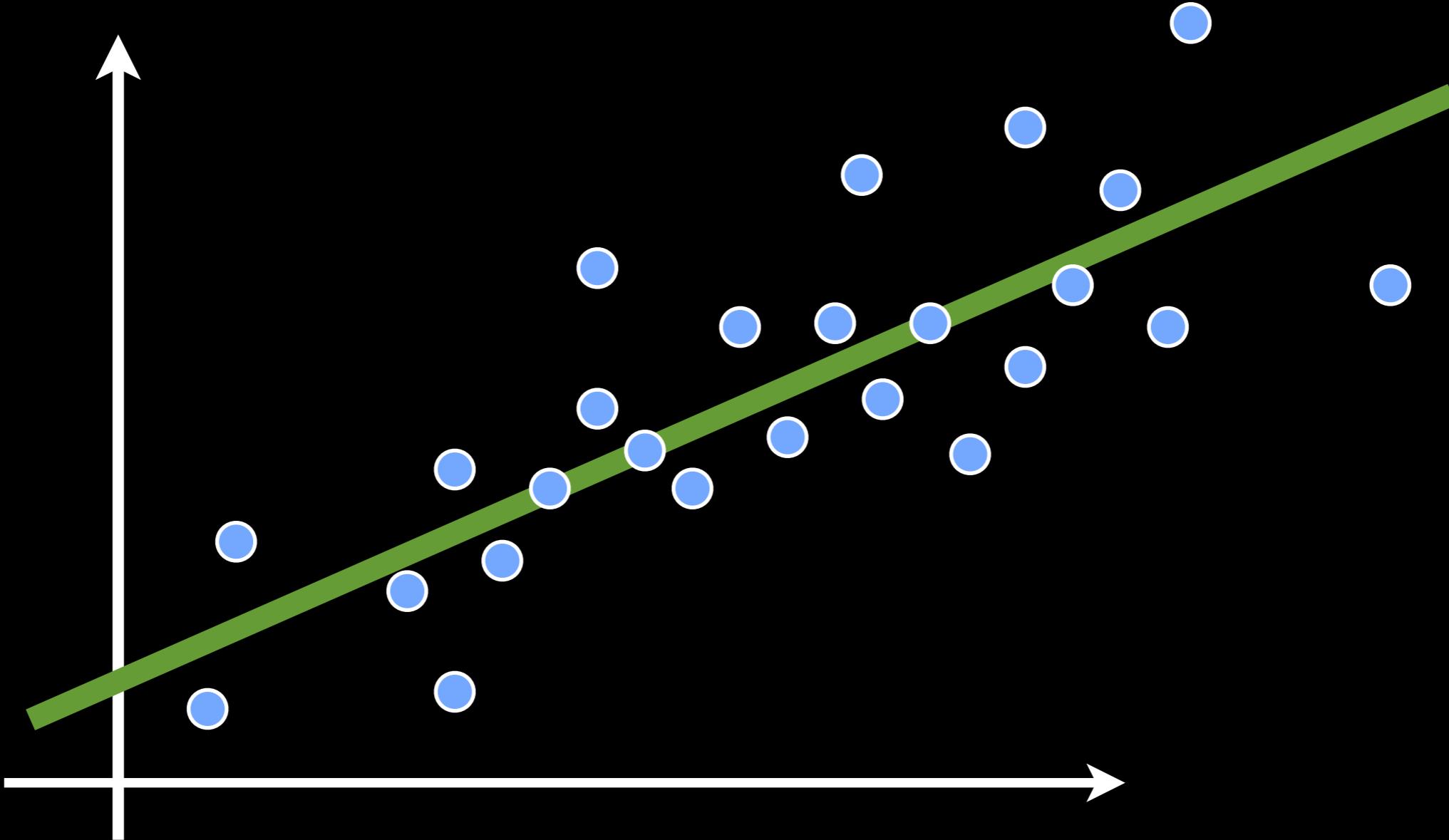
find hidden discrete structure in unlabelled data

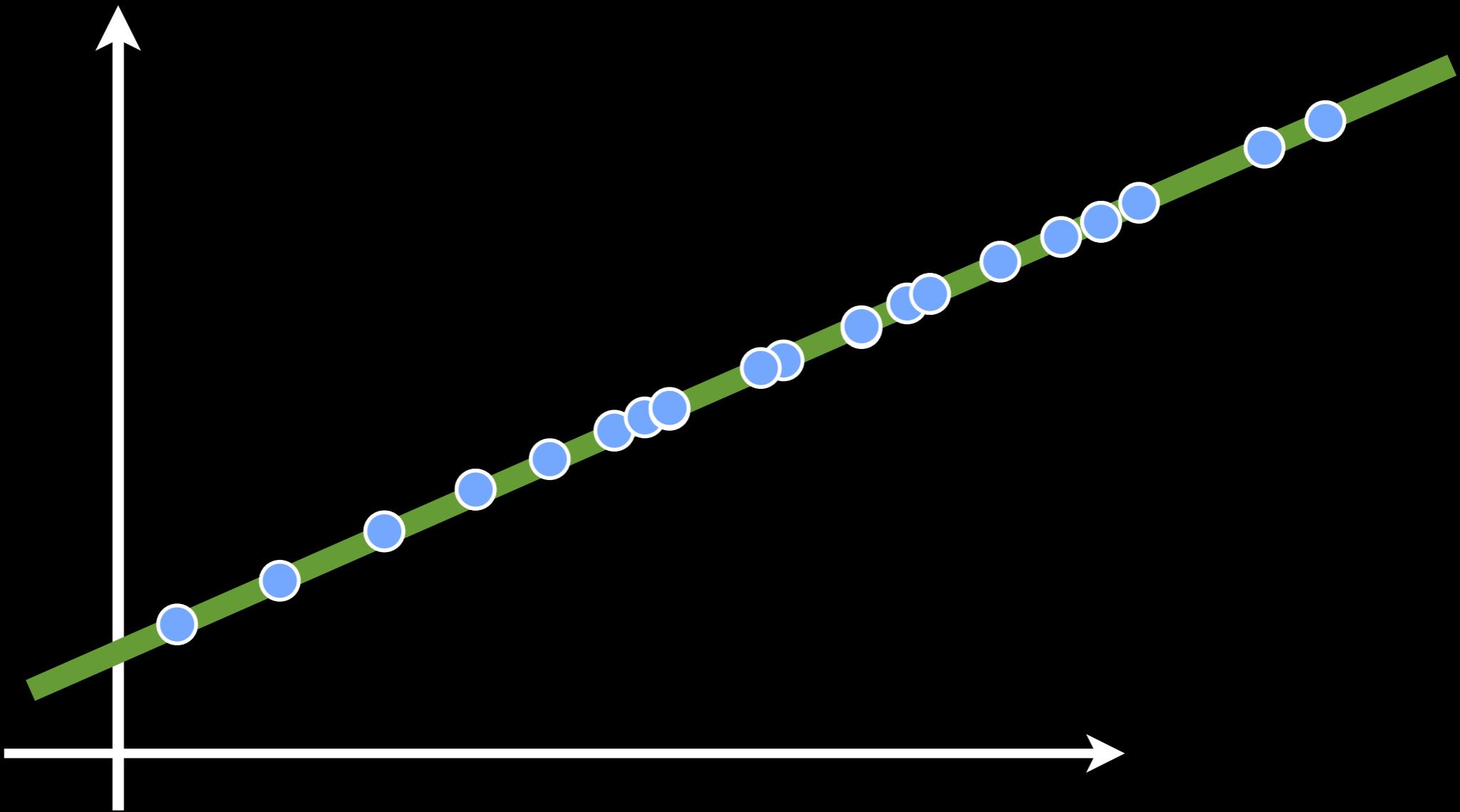


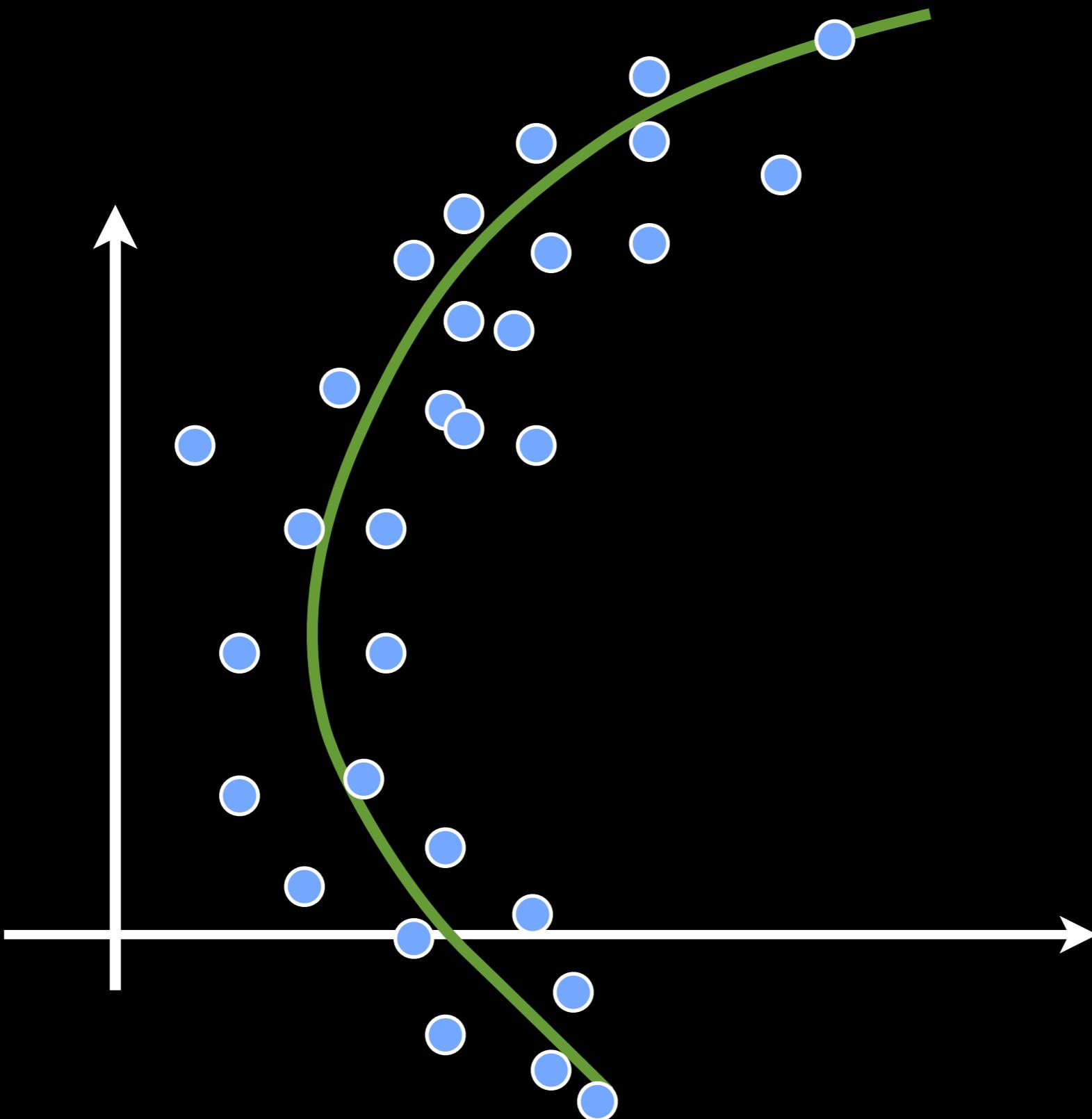
4

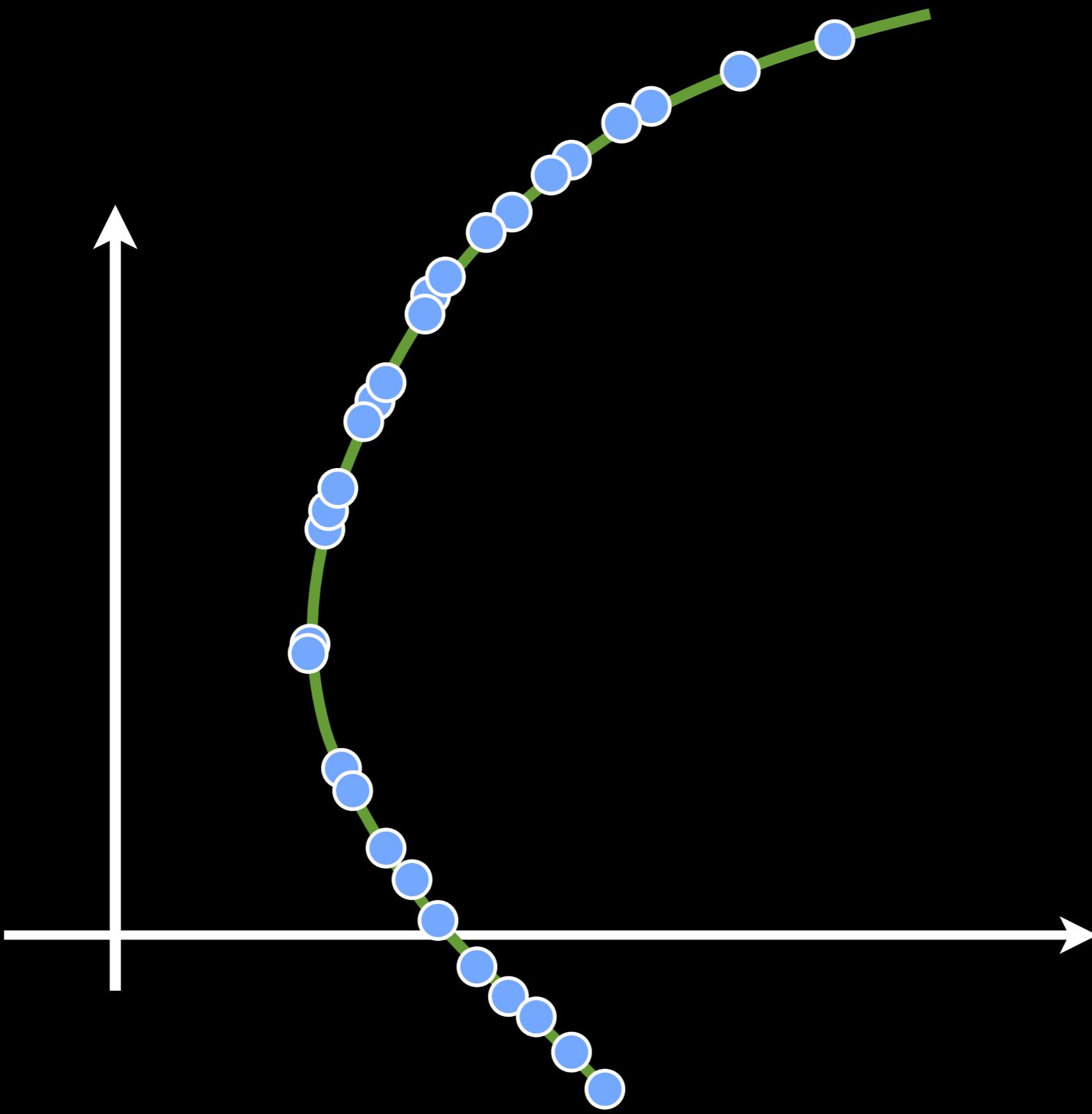
dimensionality reduction

find hidden continuous structure in data









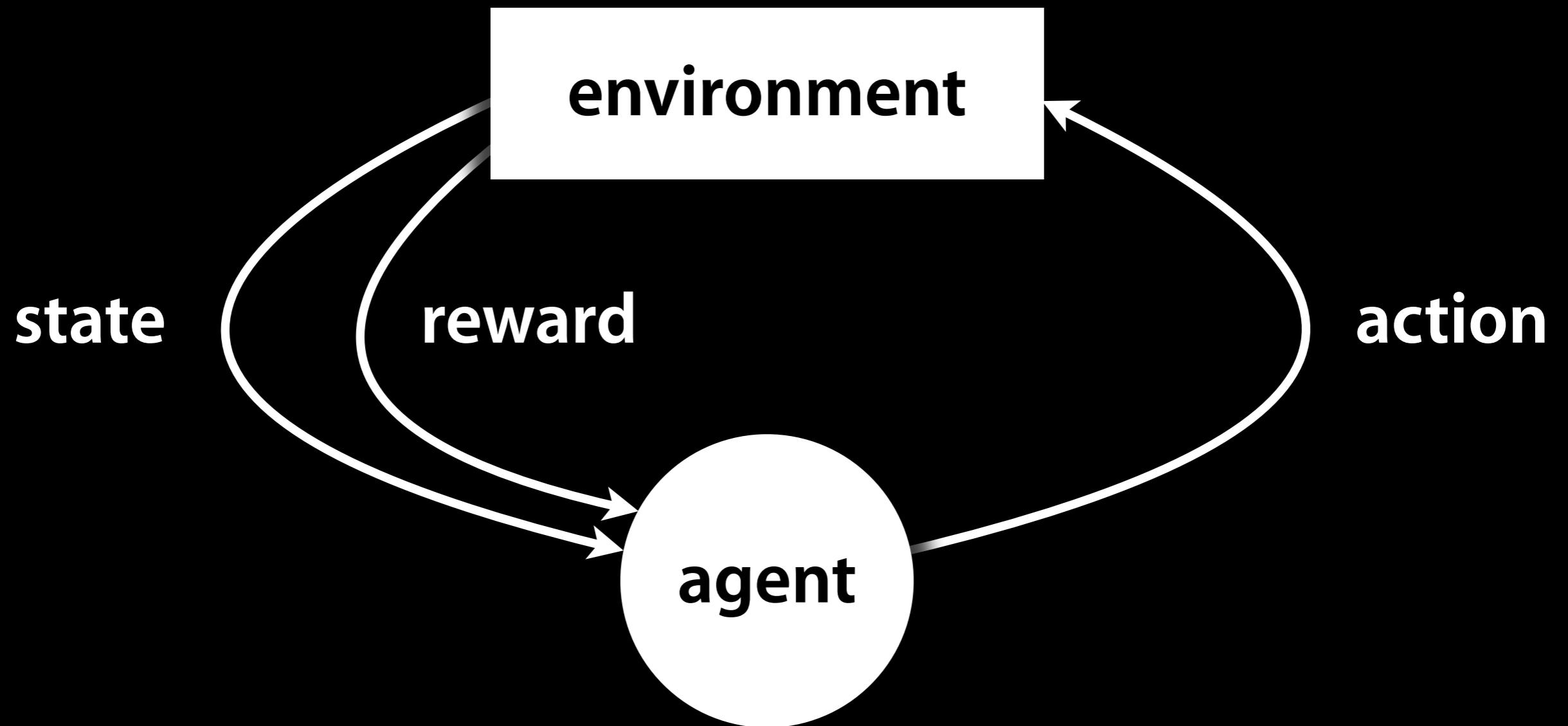
Summary

	supervised	unsupervised
discrete	classification	clustering
continuous	regression	dimensionality reduction

5

reinforcement learning

learning by interacting with environment



240 minutes of training



Building

Image Recognition



image classification

**assign an image to a known class based
on the objects present in the image**



What objects are present in the image?



ImageNet Challenge



ImageNet Challenge

1000 categories, a million images

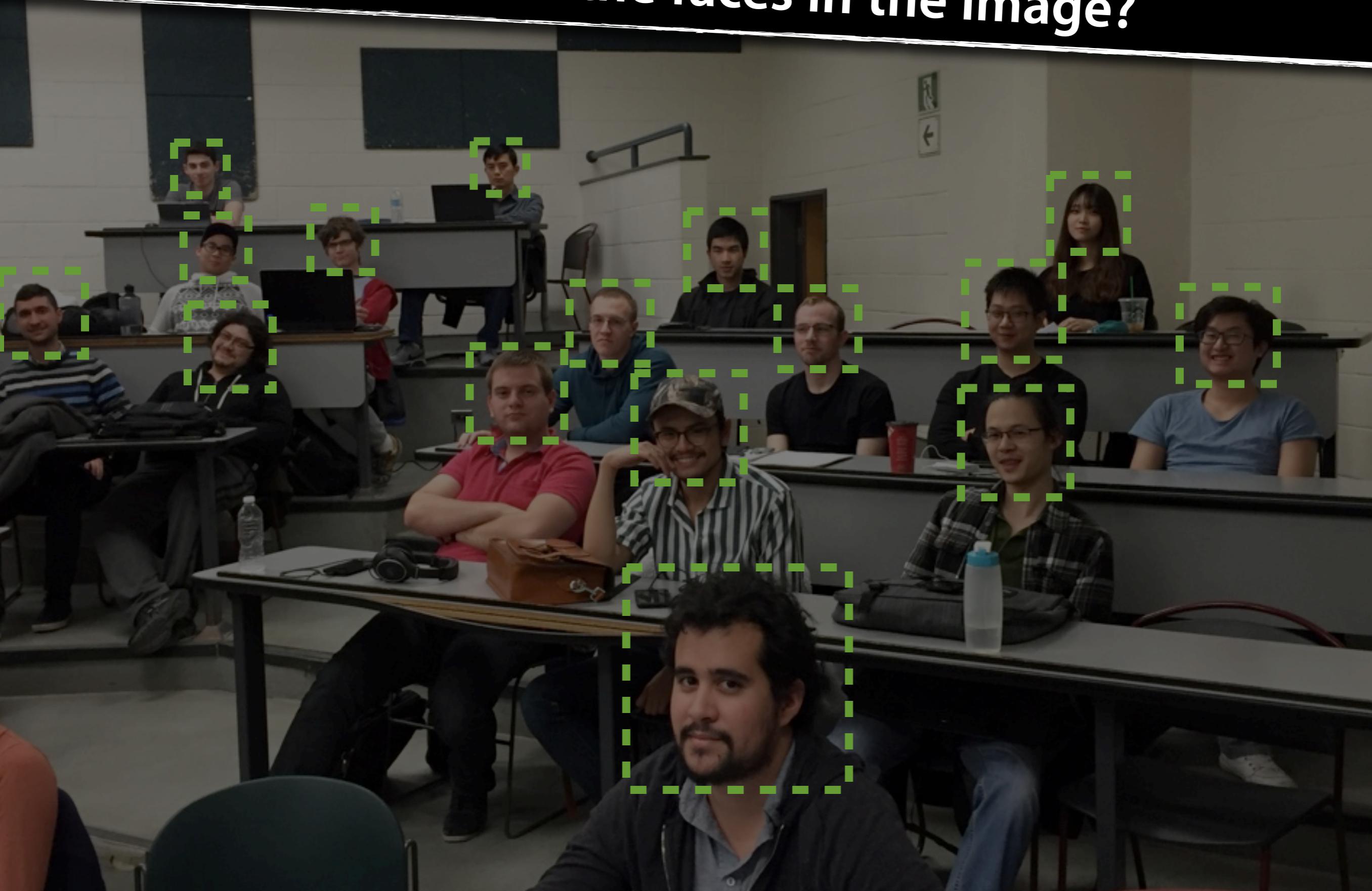
scene classification

**assign an image to a known class based
on the scene in the image**

object detection

**locate where all object instances of
known classes are present in the image**

Where are all the faces in the image?



semantic segmentation

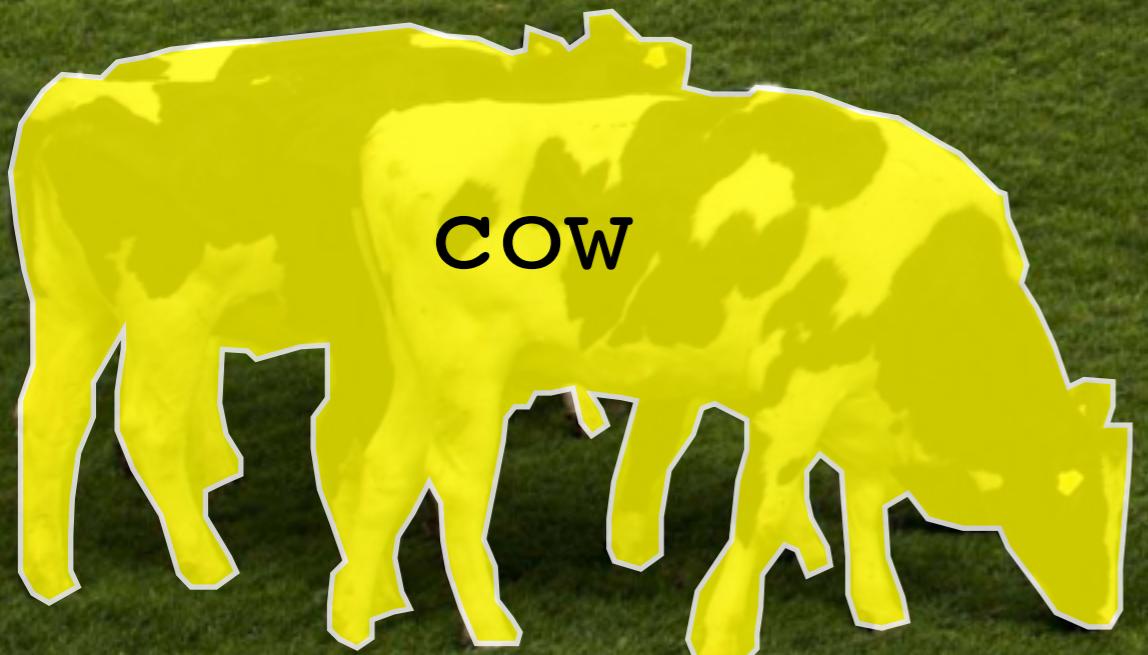
assign each image pixel to a known class



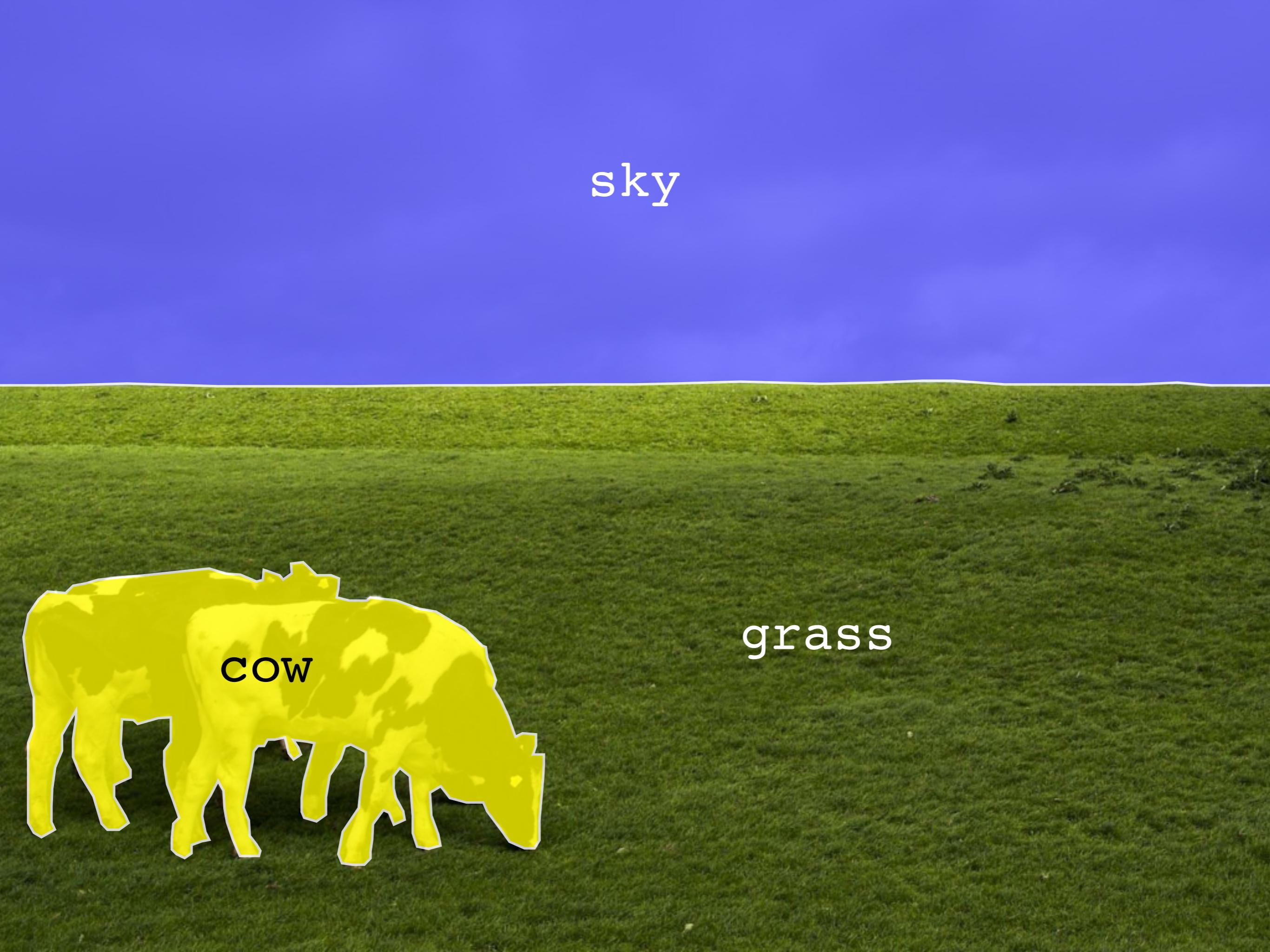
Label each pixel by semantic category



sky



COW



grass

image captioning

**provide a short sentence describing the
image**

"a group of zebras drinking water"



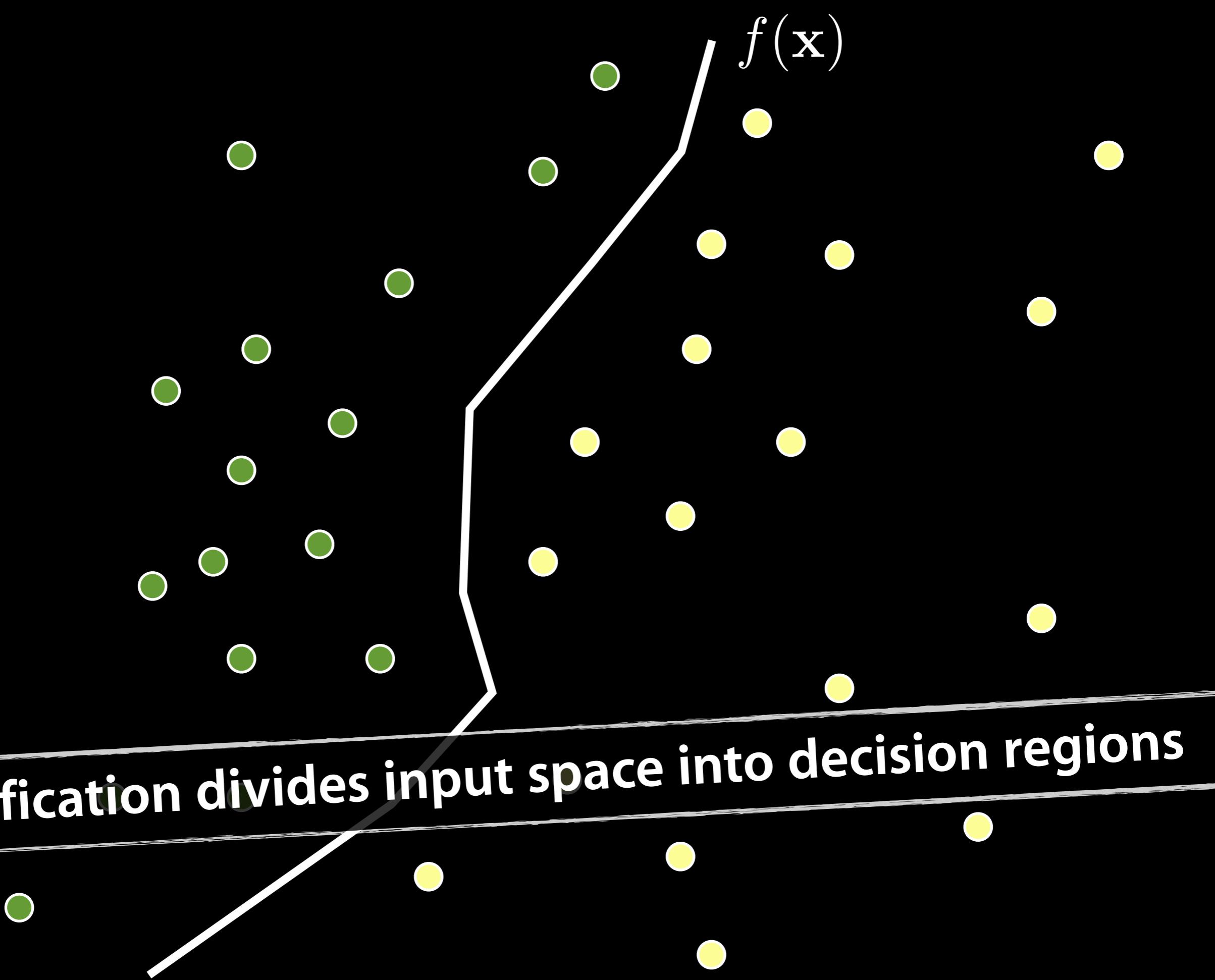
Classification

Given a training set of N observations

$$\{(\mathbf{x}_i, y_i)\}_{i=1}^N \quad \text{where} \quad \mathbf{x}_i \in \mathbb{R}^D \quad \text{and} \quad y_i \in \{-1, 1\}$$

$$f(\mathbf{x})$$

classification divides input space into decision regions



Support Vector Machines

Restricted Boltzmann Machine

Naive Bayes

Bayesian Network

Convolutional Neural Network

Randomized Forest

Boosted Decision Tree

AdaBoost

K-Nearest Neighbour

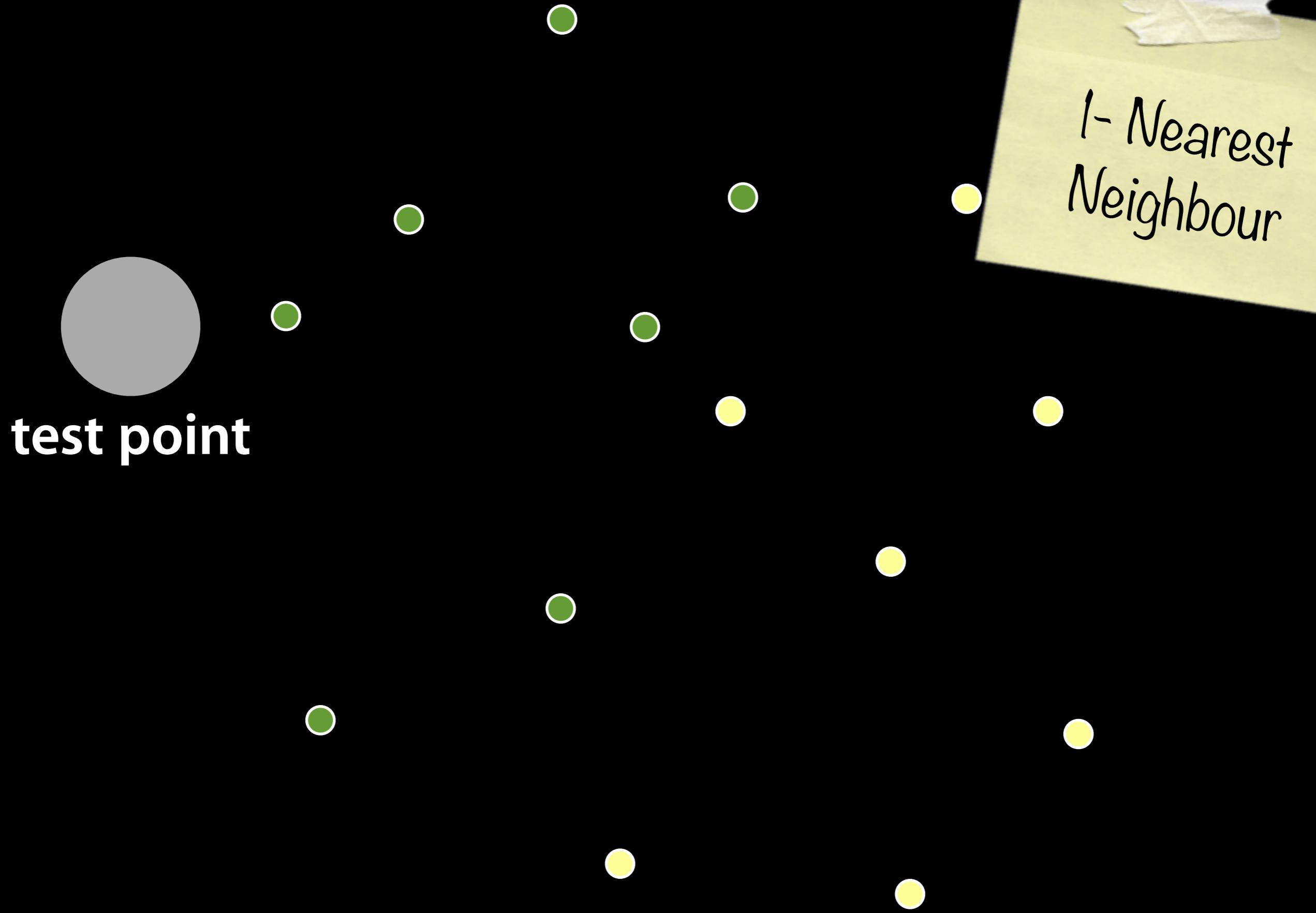
Decision Tree

K-Nearest Neighbour Classification



**For each test point, find the K nearest samples
in the training data**

**Classify the point according to the majority vote
of their class label**



test point

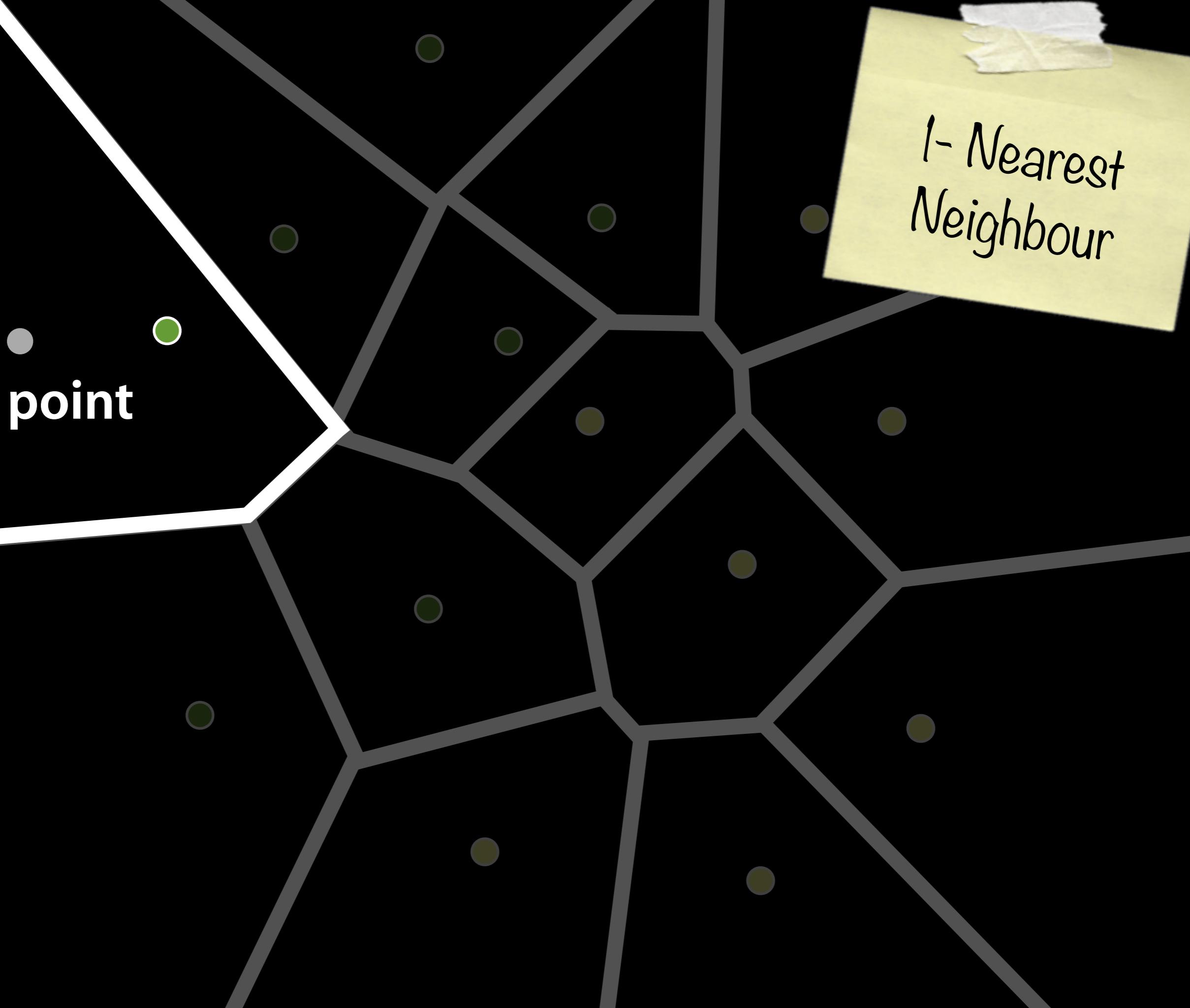
1-Nearest
Neighbour

test point

1-Nearest
Neighbour

1-Nearest
Neighbour

test point



test point

1-Nearest
Neighbour



3- Nearest
Neighbour

test point

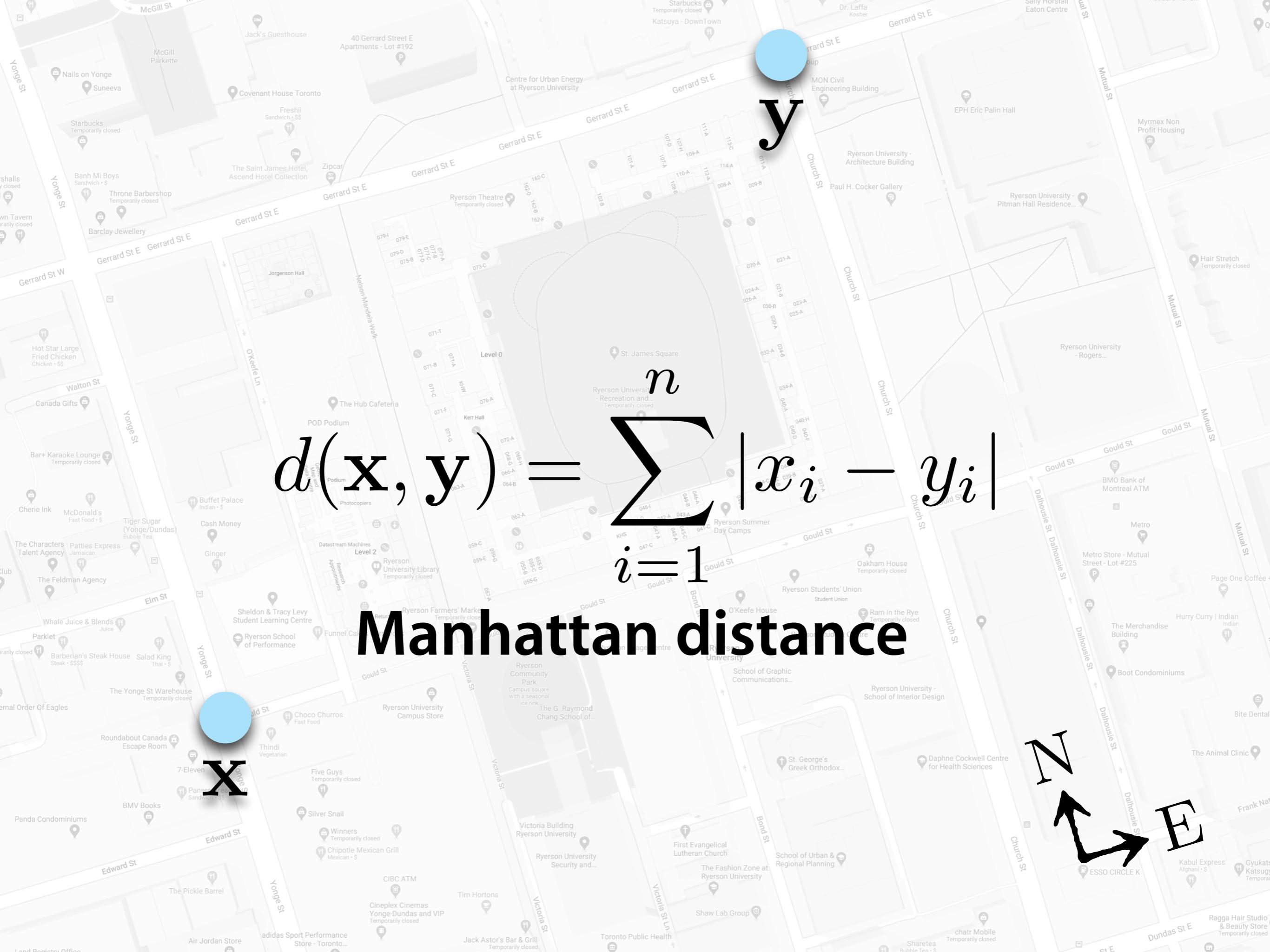
classify point according to the majority vote of labels

Distance Measures



$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

Manhattan distance



$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

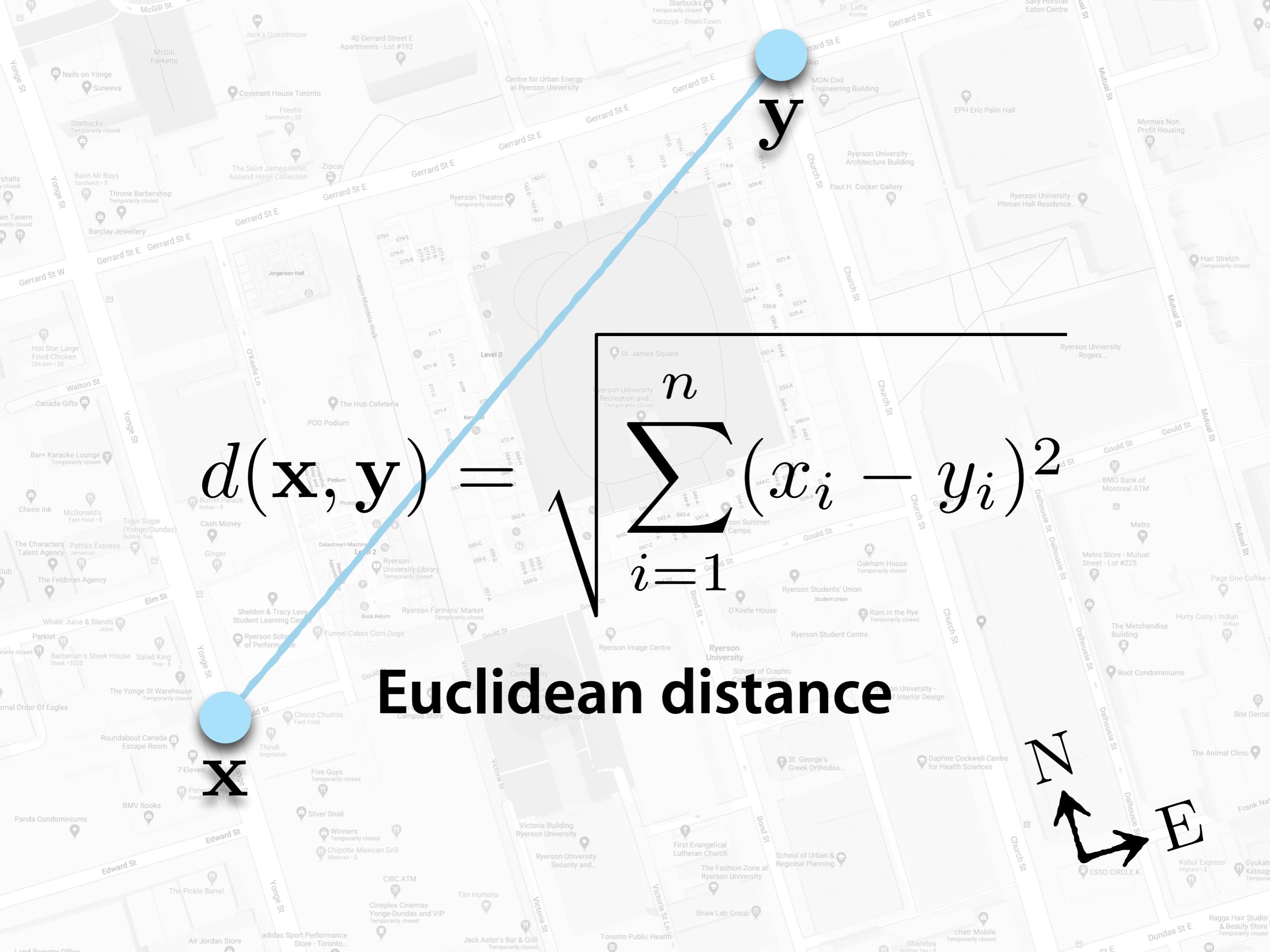
City-Block distance

N
E

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Euclidean distance

N
E



Linear Classification



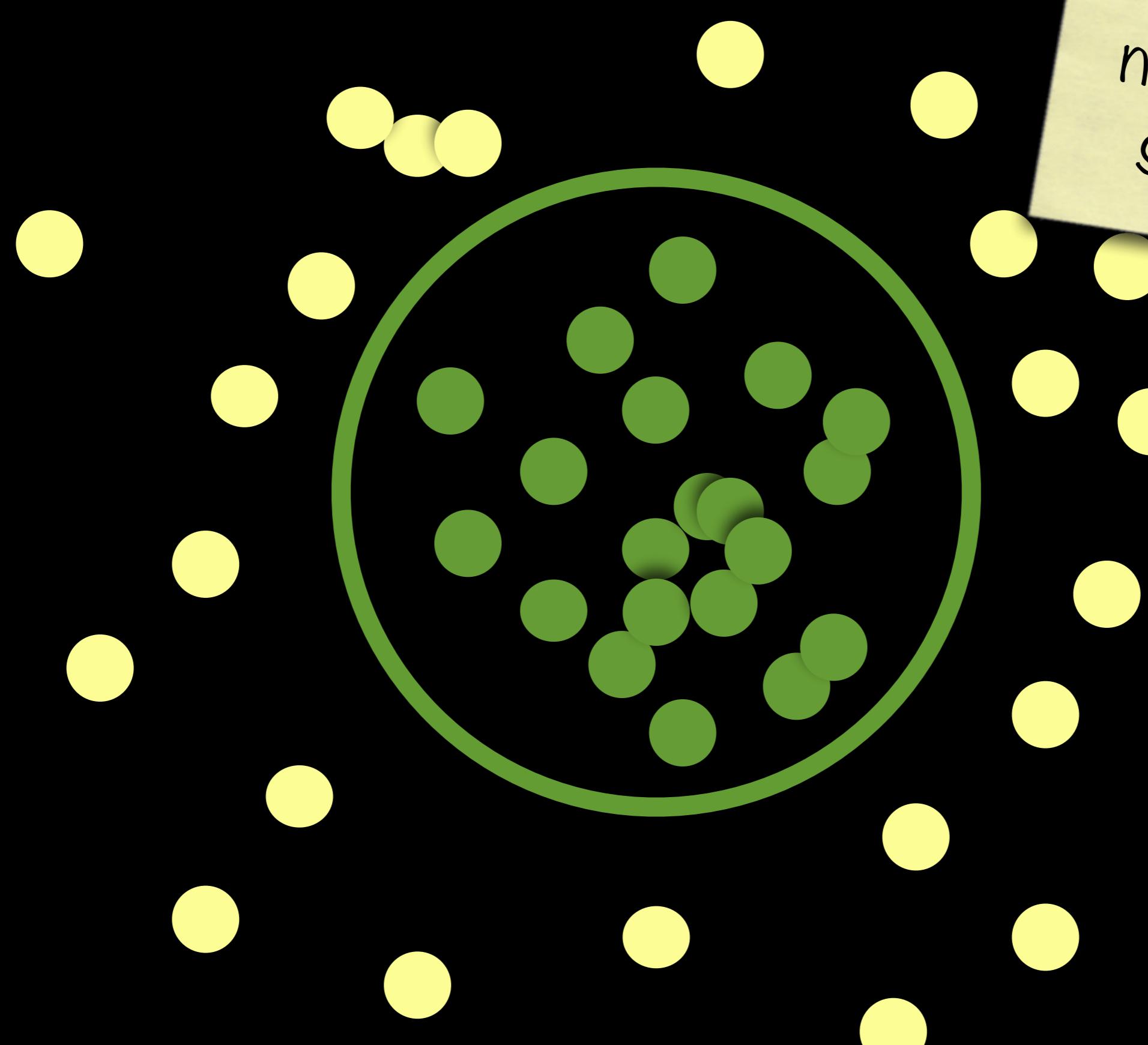
pedestrian



non-pedestrian



Find a linear function that “best” separates the classes



non-linearly
separable

pedestrian

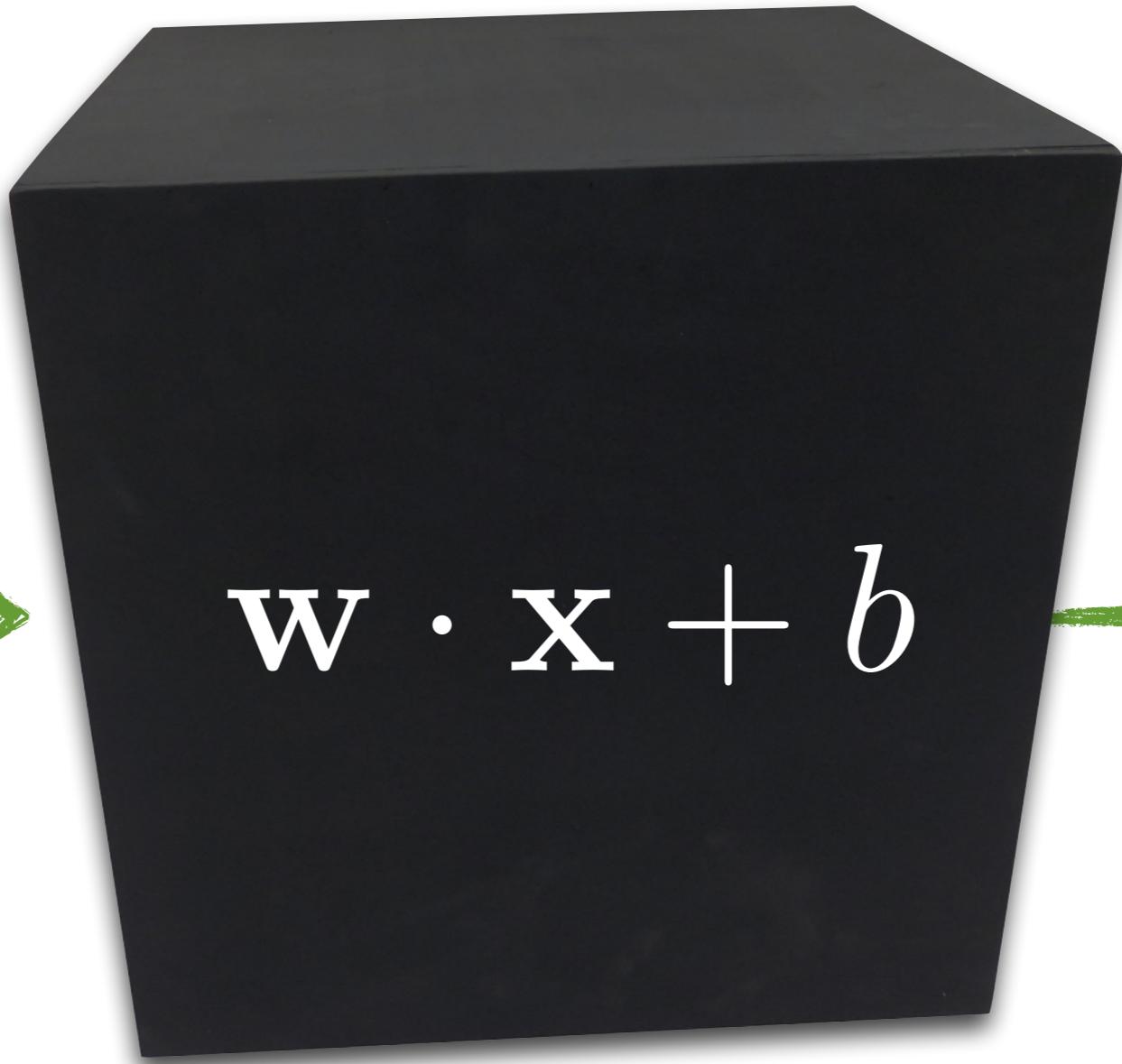
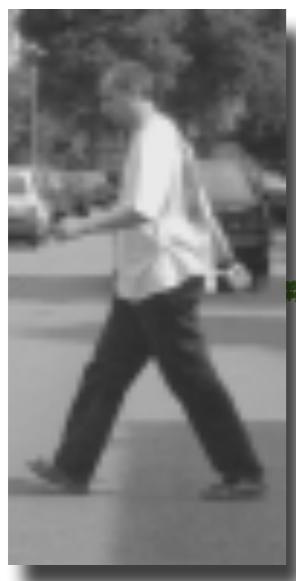


non-pedestrian



At testing, the sign of the score determines the class

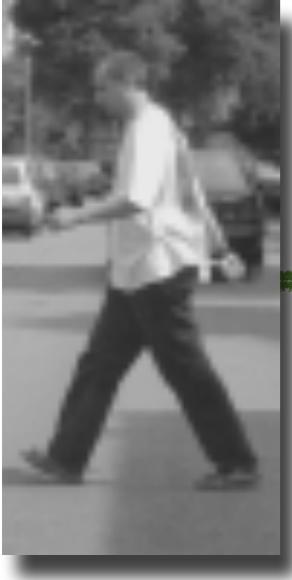
$$f(\mathbf{x}; \mathbf{w}, b) = \mathbf{w} \cdot \mathbf{x} + b$$



$$w \cdot x + b$$

score

x



\rightarrow

$$\begin{pmatrix} 0.2 \\ -0.8 \\ 0.1 \\ 2.0 \end{pmatrix} \cdot \begin{pmatrix} 56 \\ 231 \\ 24 \\ 2 \end{pmatrix} + b + 1.1 \rightarrow -96.8$$

w **x**

What are the best model parameters?

loss function

**quantifies agreement between the
predicted scores and ground truth labels**

scoring function

$$L_i = \max(0, 1 - y_i [\mathbf{w} \cdot \mathbf{x}_i + b])$$

$$L_i = \max(0, 1 - y_i [w \cdot x_i + b])$$

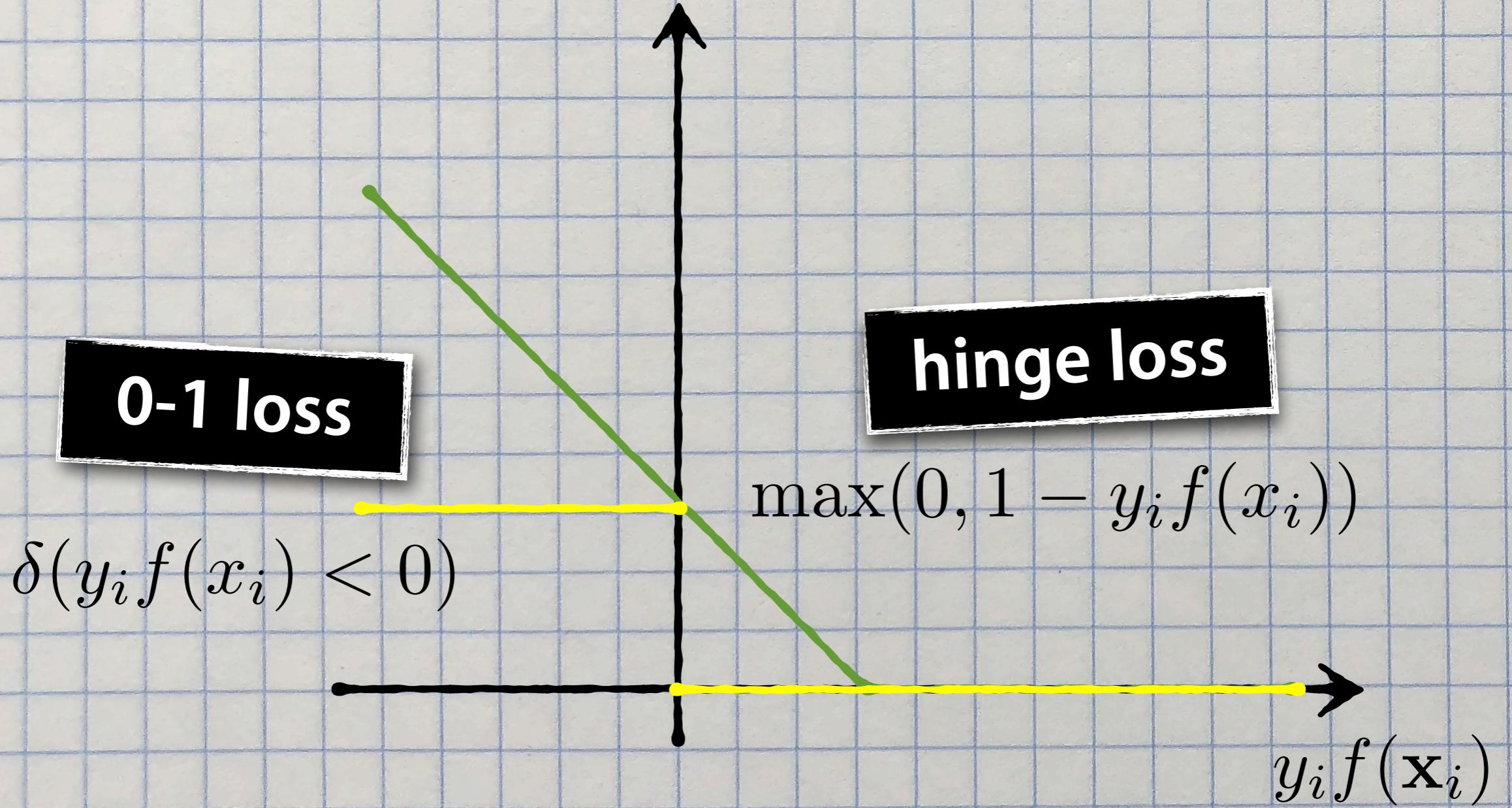
training set label, 1 or -1

loss margin

$$L_i = \max(0, 1 - y_i [\mathbf{w} \cdot \mathbf{x}_i + b])$$

$$L_i = \max(0, 1 - y_i[\mathbf{w} \cdot \mathbf{x}_i + b])$$

hinge loss function



logistic loss

$$\ln(1 + \exp(-y_i f(\mathbf{x}_i)))$$

0-1 loss

$$\delta(y_i f(x_i) < 0)$$

hinge loss

$$\max(0, 1 - y_i f(x_i))$$

hinge and logistic losses are convex approximations of 0-1 loss

$$L_i = \max(0, 1 - y_i[\mathbf{w} \cdot \mathbf{x}_i + b])$$



score	0.3	-30	-96.8
--------------	-----	-----	-------

loss	0.7	0	97.8
-------------	-----	---	------

Overall loss is the average loss over the training set

$$L_i = \max(0, 1 - y_i[\mathbf{w} \cdot \mathbf{x}_i + b])$$



score	0.3	-30	-96.8
--------------	-----	-----	-------

loss	0.7	0	97.8
-------------	-----	---	------

Overall loss is the average loss over the training set

$$L = \frac{1}{N} \sum_{i=1}^N L_i = \frac{1}{N} \sum_{i=1}^N \max(0, 1 - y_i[\mathbf{w} \cdot \mathbf{x}_i + b])$$

$$L = \frac{1}{N} \sum_{i=1}^N \max(0, 1 - y_i [\mathbf{w} \cdot \mathbf{x}_i + b])$$

Loss function admits multiple solutions

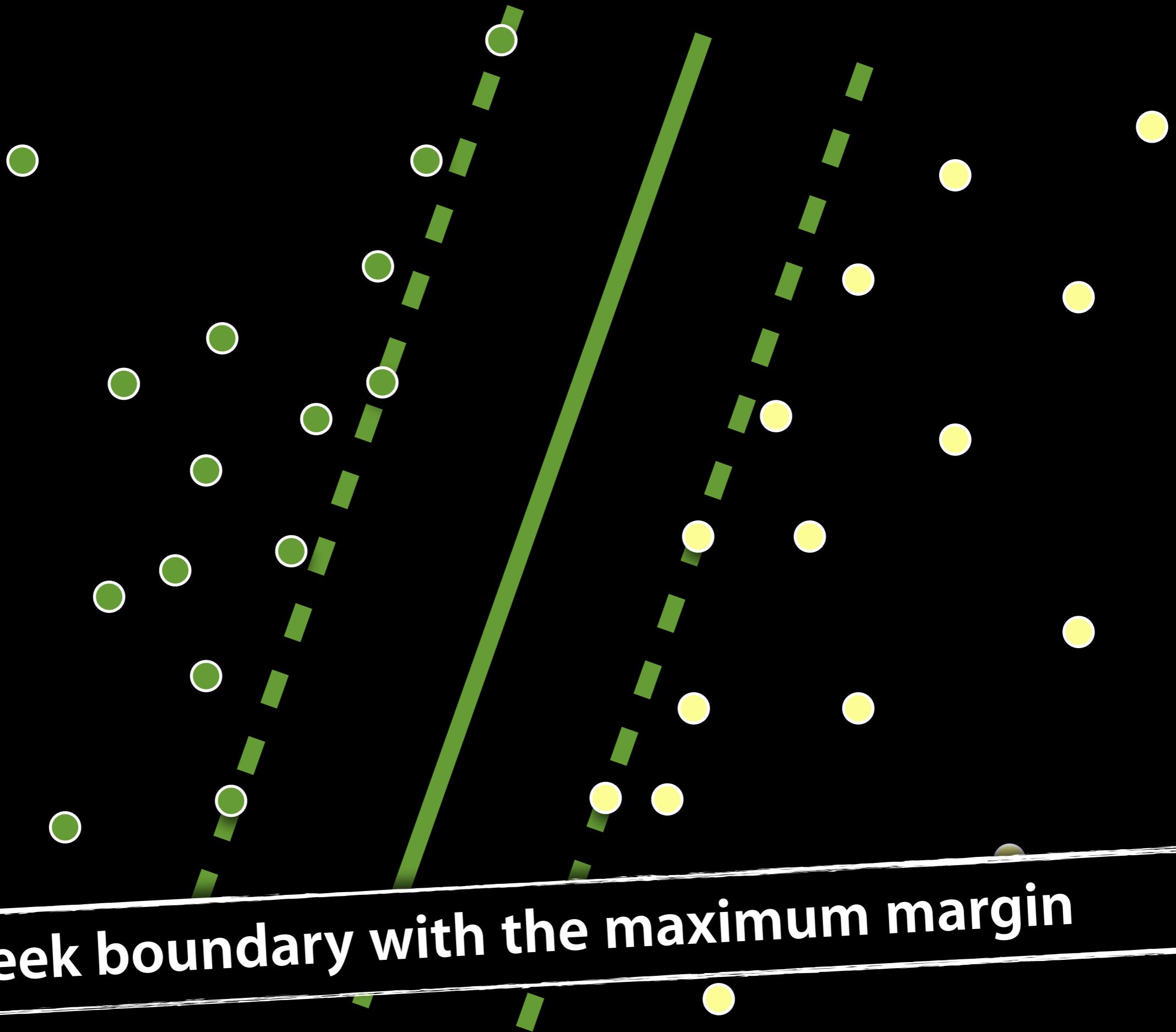
$$L = \frac{1}{N} \sum_{i=1}^N \max(0, 1 - y_i [\mathbf{w} \cdot \mathbf{x}_i + b]) + \lambda R(\mathbf{w})$$

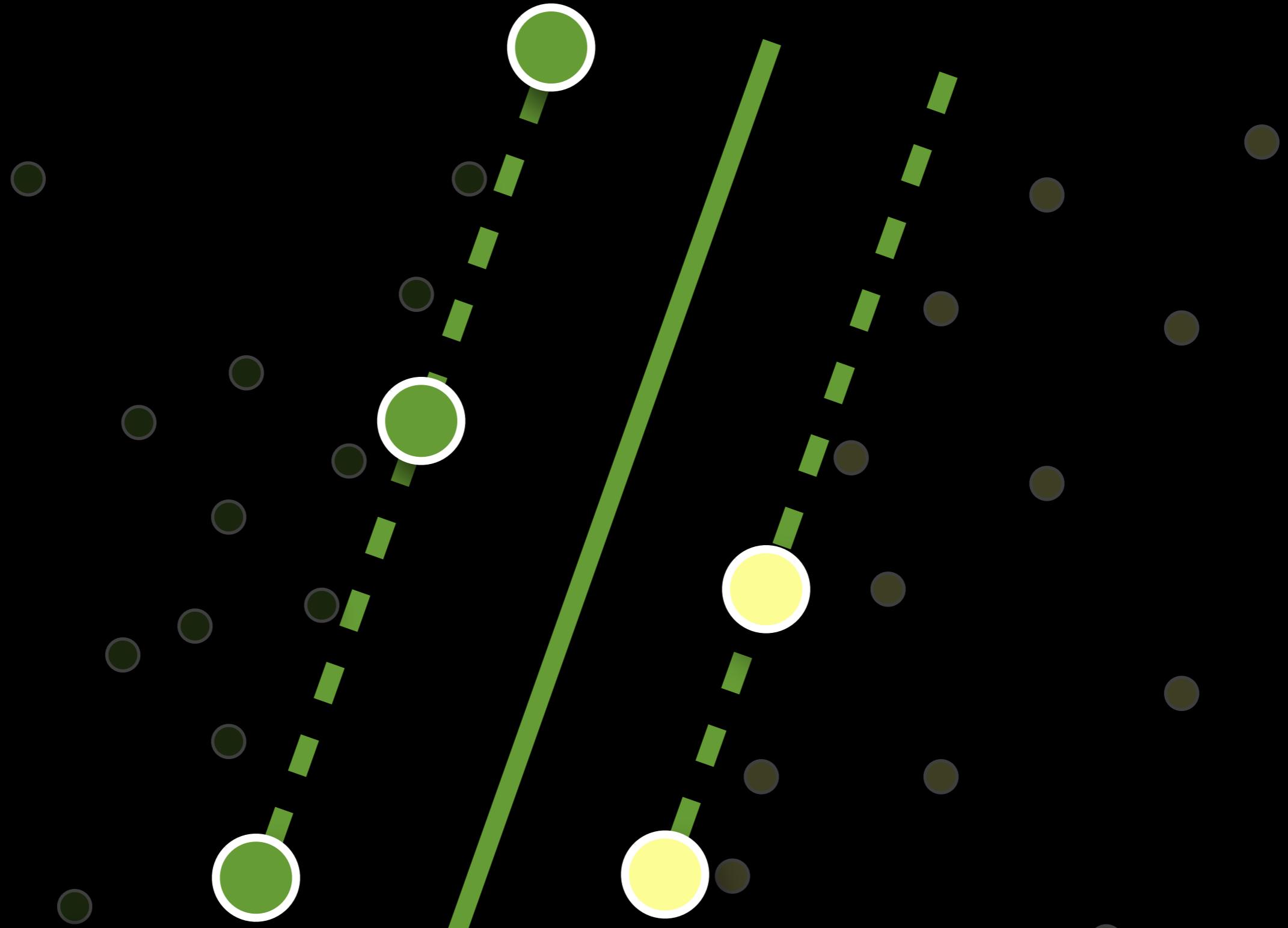
L2 norm is a common regularizer

$$R(\mathbf{w}) = \|\mathbf{w}\|_2^2$$

Support Vector Machine (SVM)

seek boundary with the maximum margin





samples on the margin are called **support vectors**

$$f(\mathbf{x}; \mathbf{w}) = \mathbf{w} \cdot \mathbf{x} + b$$

scoring function

$$L = \frac{1}{N} \sum_{i=1}^N \max(0, 1 - y_i [\mathbf{w} \cdot \mathbf{x}_i + b]) + \lambda R(\mathbf{w})$$

loss function

How do we find the optimal model parameters?

gradient

$$\nabla f(x_1, x_2, \dots, x_N) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_N} \end{pmatrix}$$

points in the direction of greatest rate of increase

gradient

$$\nabla f(x_1, x_2, \dots, x_N) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_N} \end{pmatrix}$$

magnitude is the rate of increase

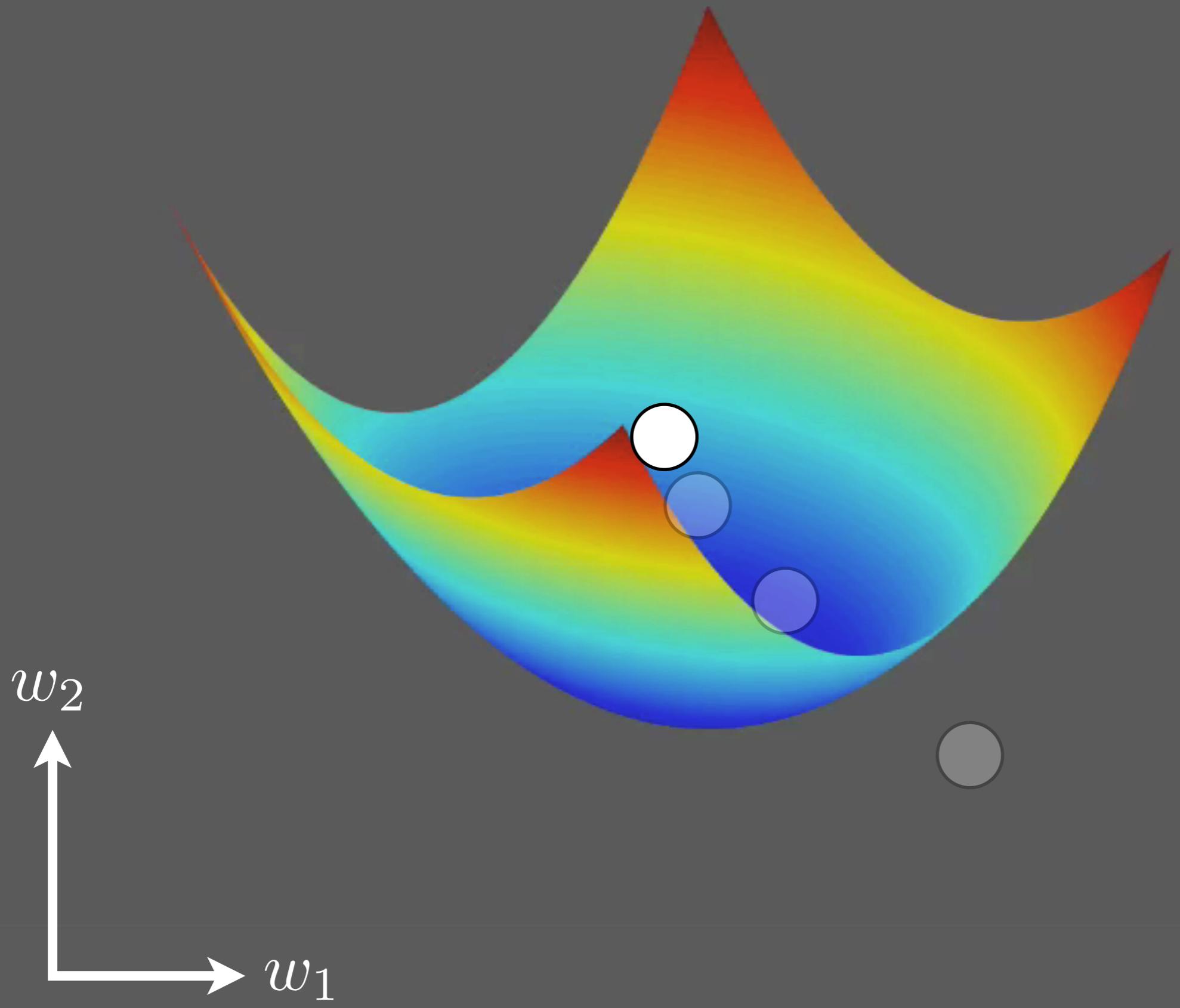


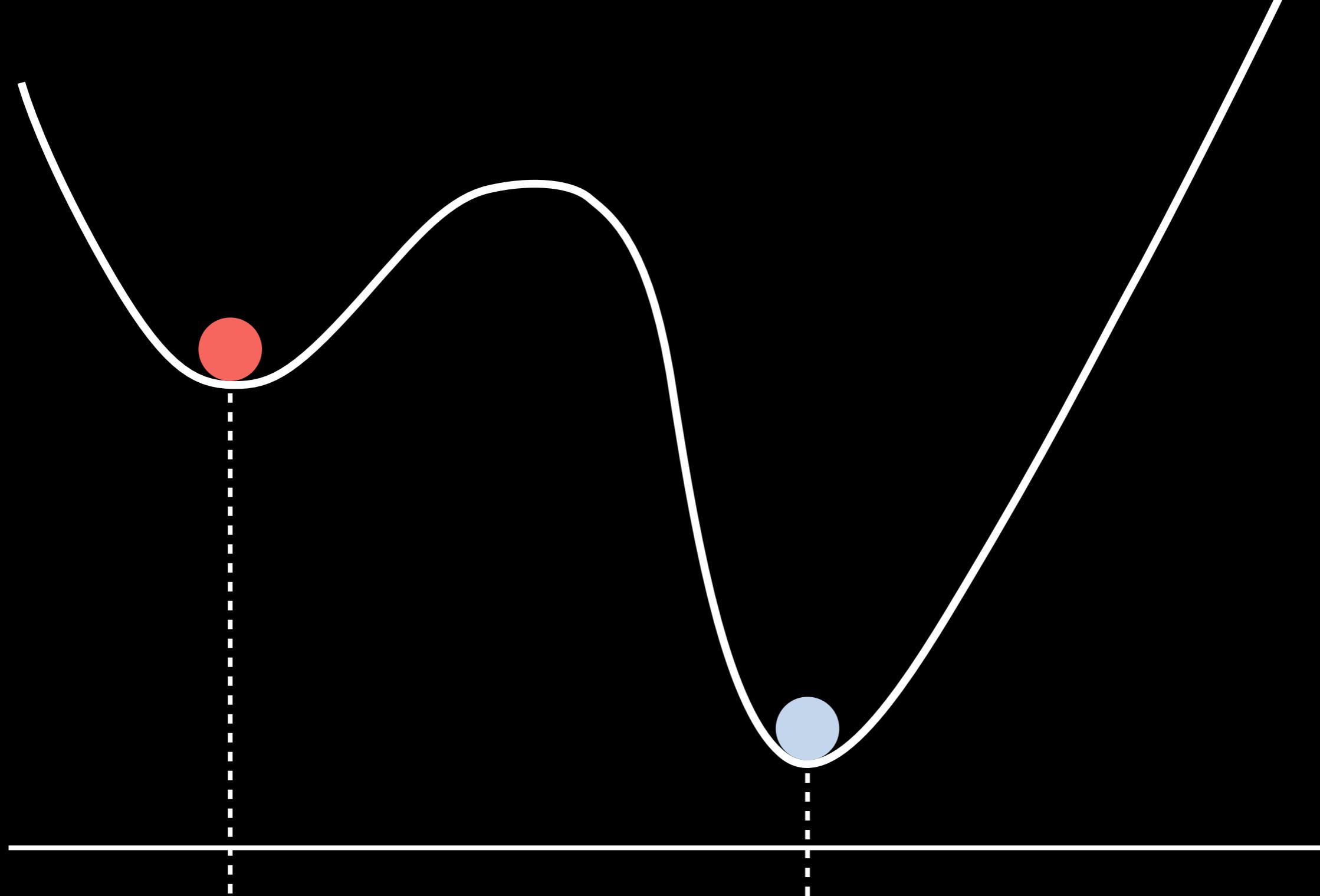
What is the direction of greatest descent?

$$-\nabla L(\mathbf{w}_t)$$

update parameters in negative gradient direction

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla L(\mathbf{w}_t)$$





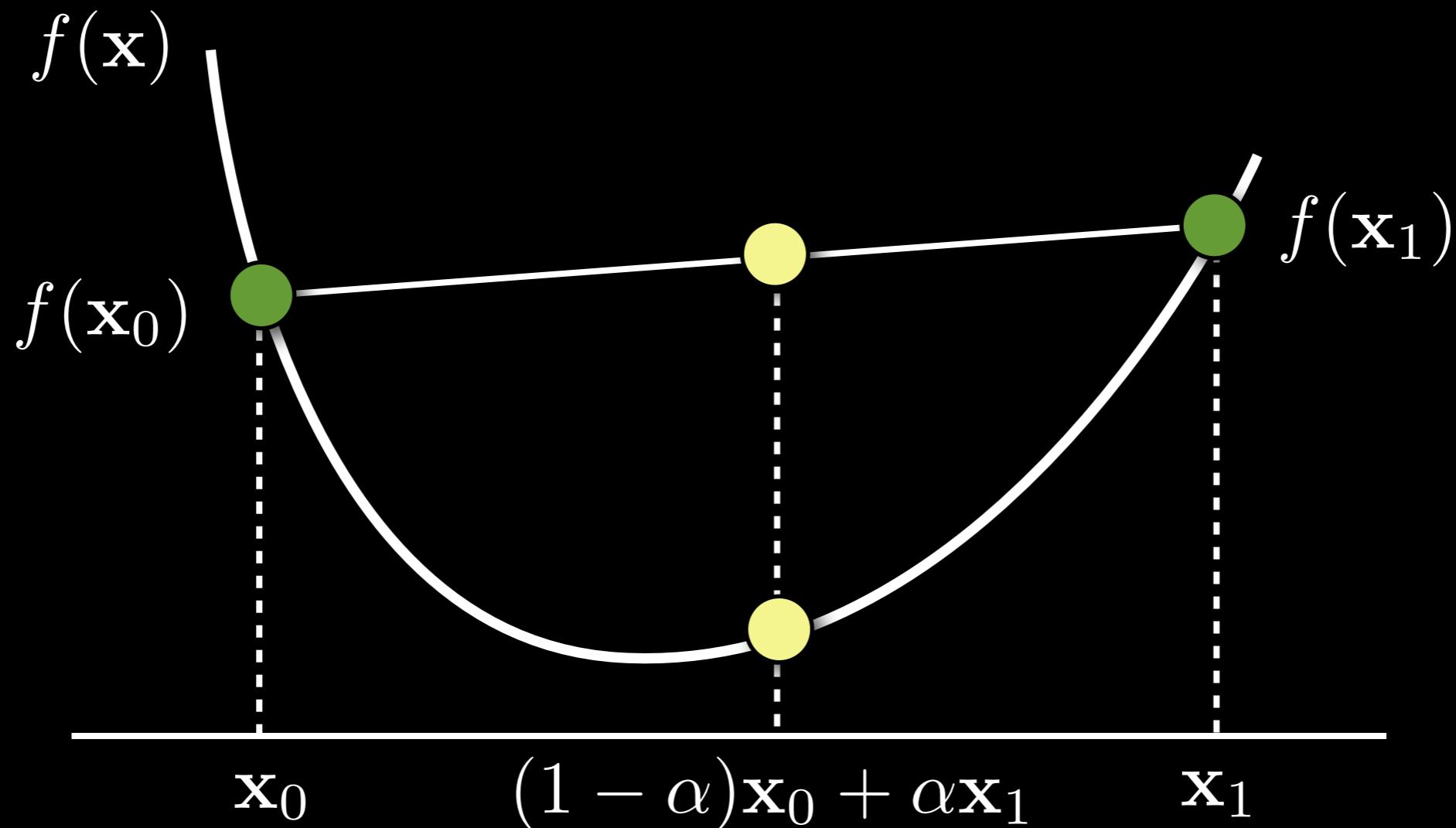
local
minima

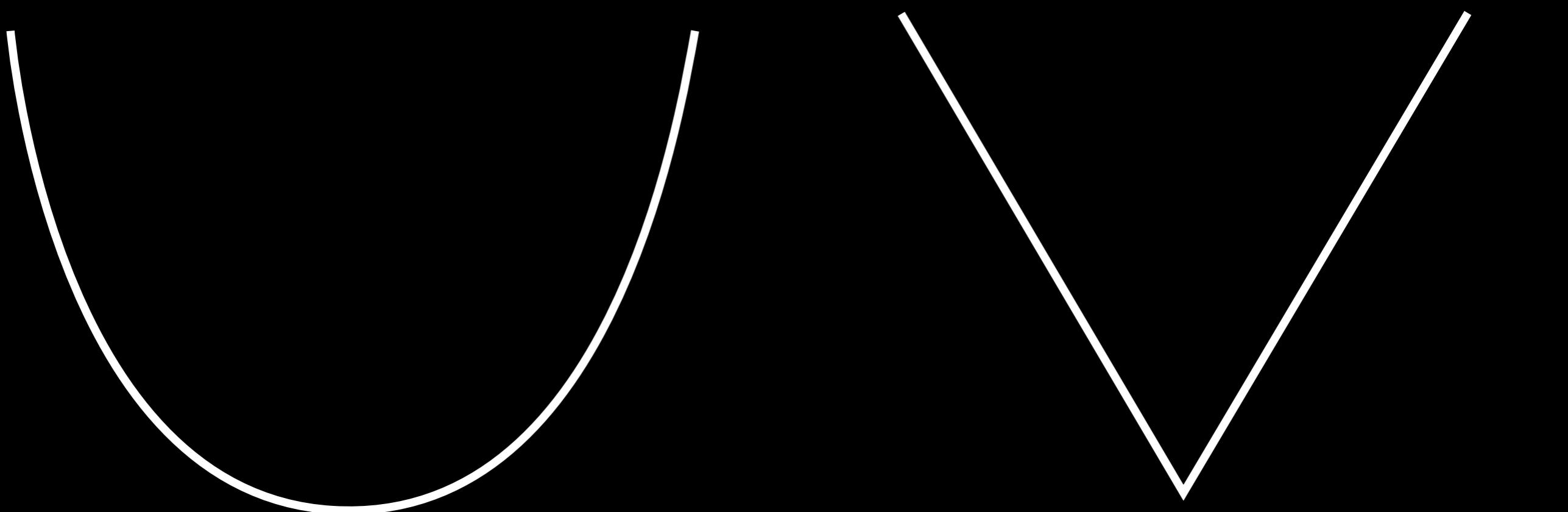
global
minimum

Definition:

A function, $f : D \rightarrow \mathbb{R}$, is said to be **convex** if for any \mathbf{x}_0 and \mathbf{x}_1 in D and $0 \leq \alpha \leq 1$:

$$f((1 - \alpha)\mathbf{x}_0 + \alpha\mathbf{x}_1) \leq (1 - \alpha)f(\mathbf{x}_0) + \alpha f(\mathbf{x}_1).$$

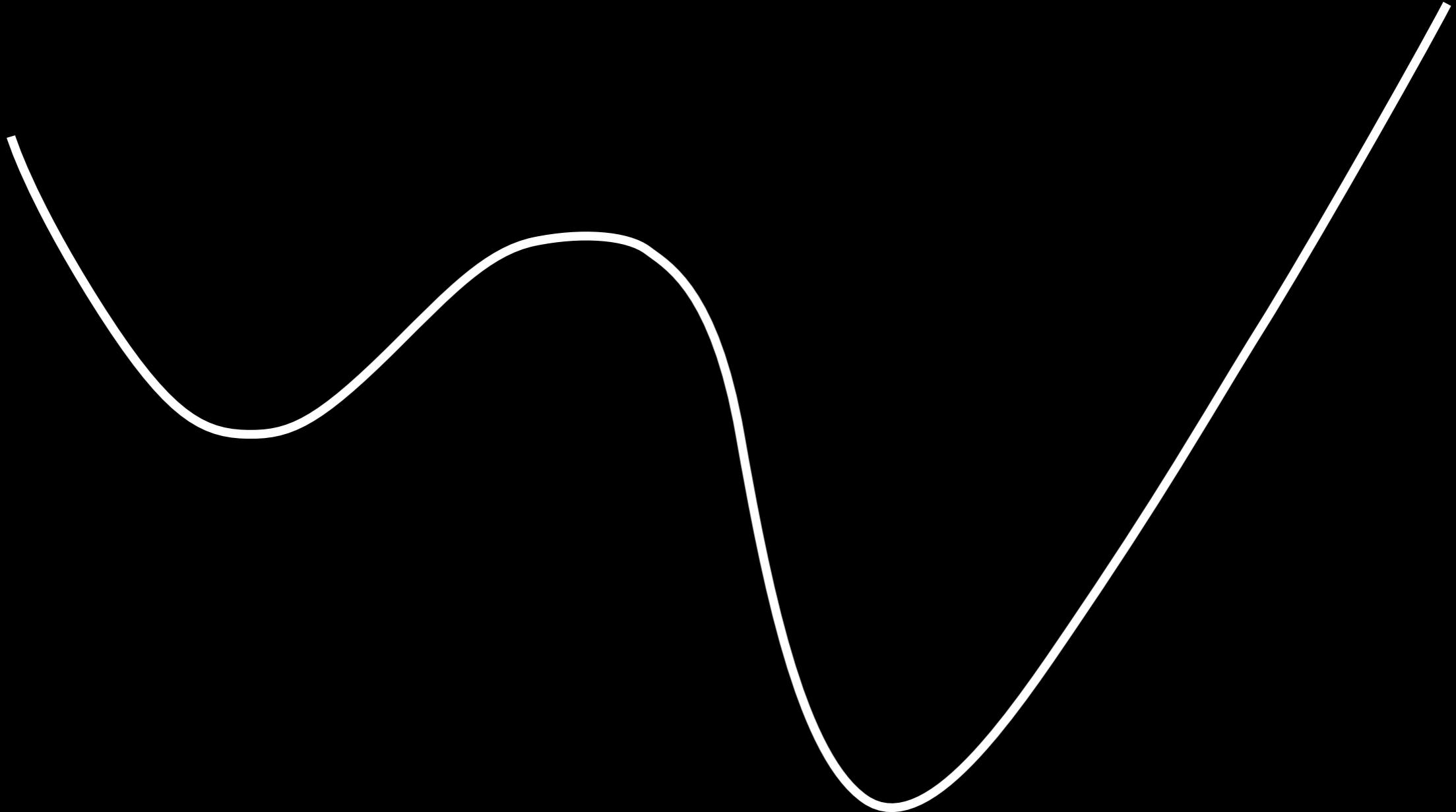




convex functions

$$L = \frac{1}{N} \sum_{i=1}^N \max(0, 1 - y_i [\mathbf{w} \cdot \mathbf{x}_i + b]) + \lambda R(\mathbf{w})$$

SVM cost function is convex



non-convex function

GENERALIZATION

objective of machine learning

Data

NEVER train and test on the same data

Training Data

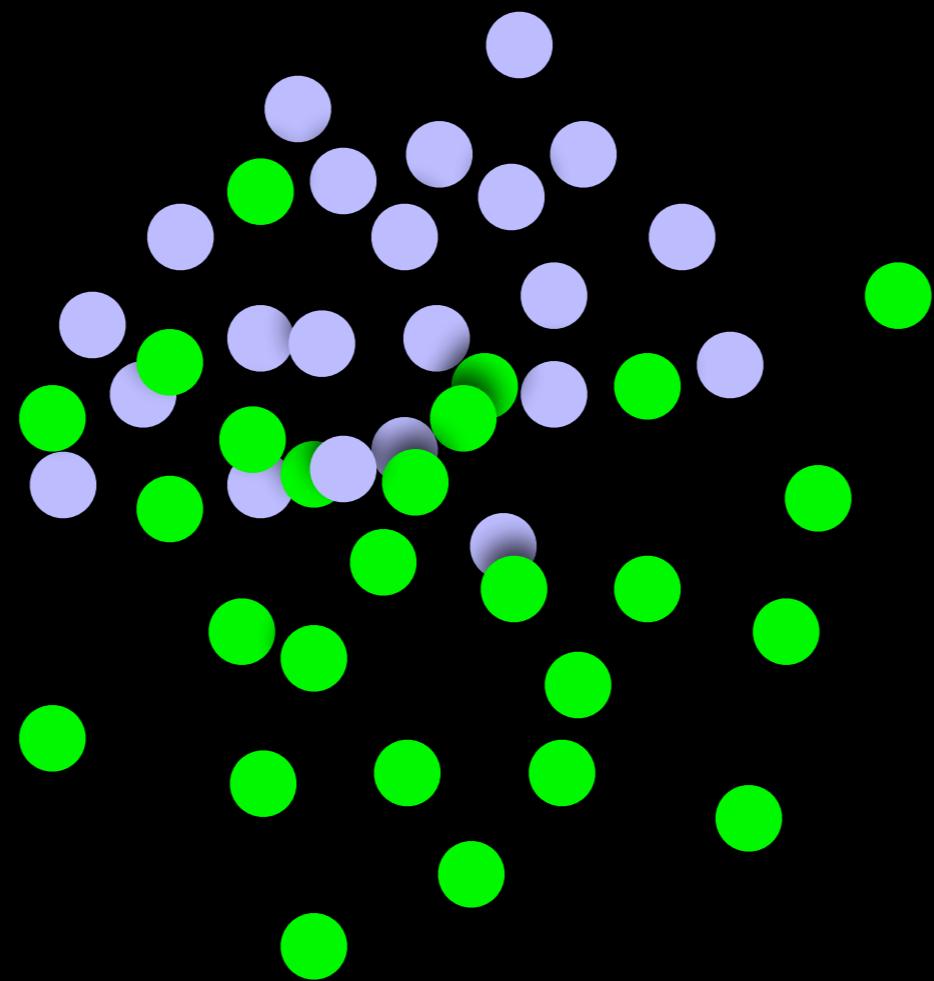
Test

Train on training data and test on withheld test set

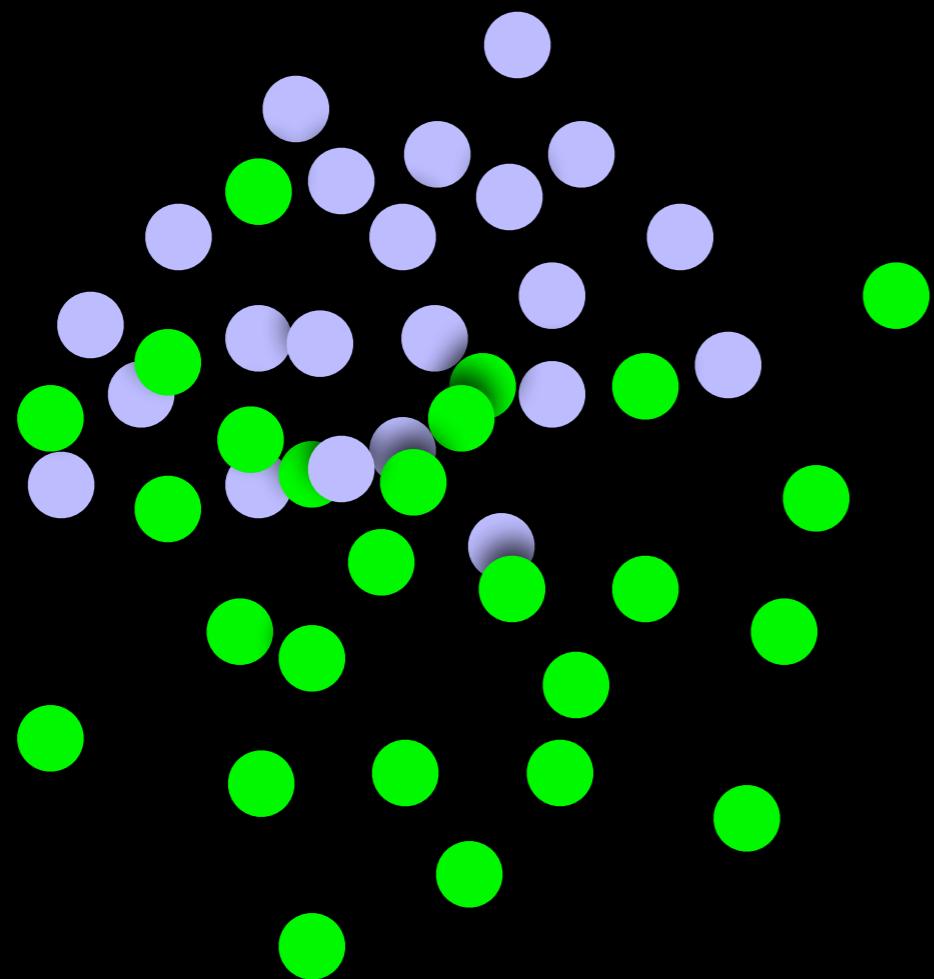
Training Data

Test

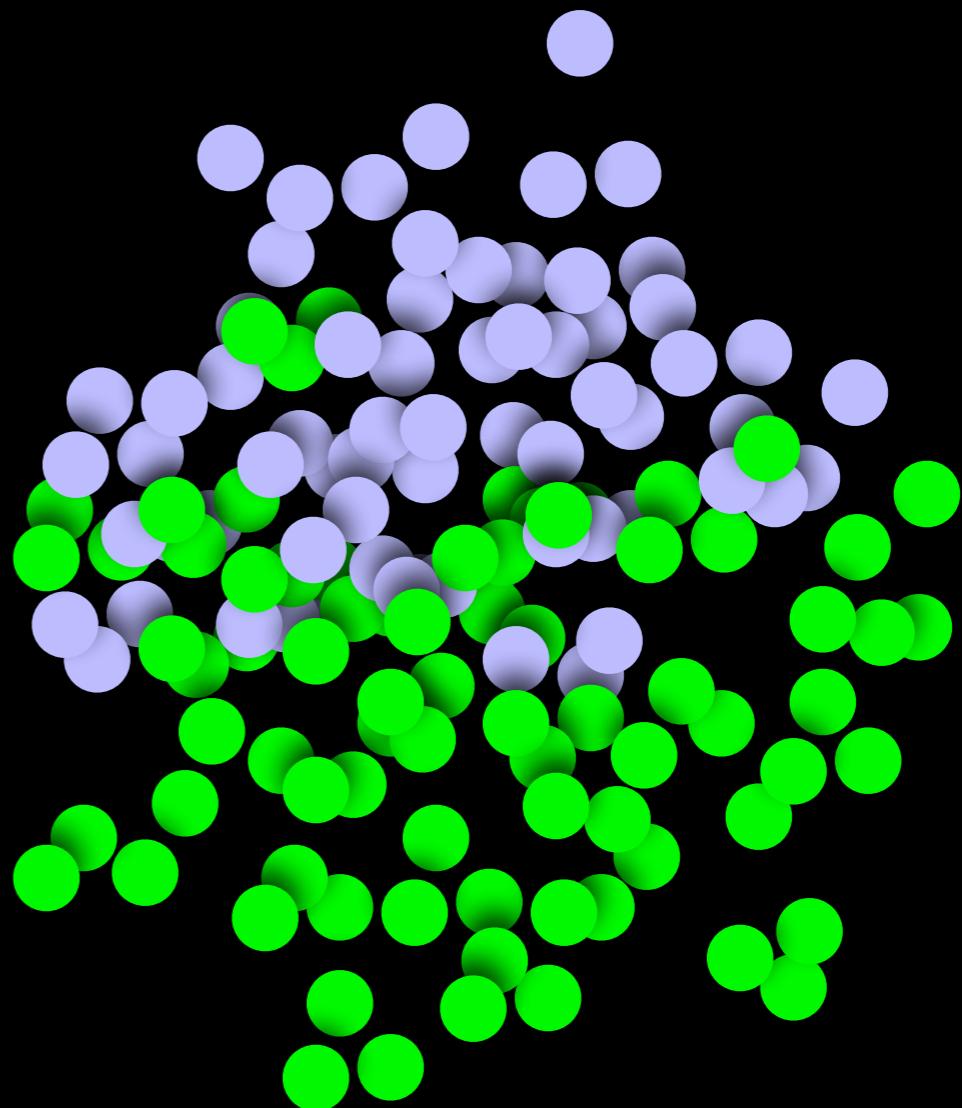
**Assume training and testing data drawn
independently from the same distribution**



training data



training data

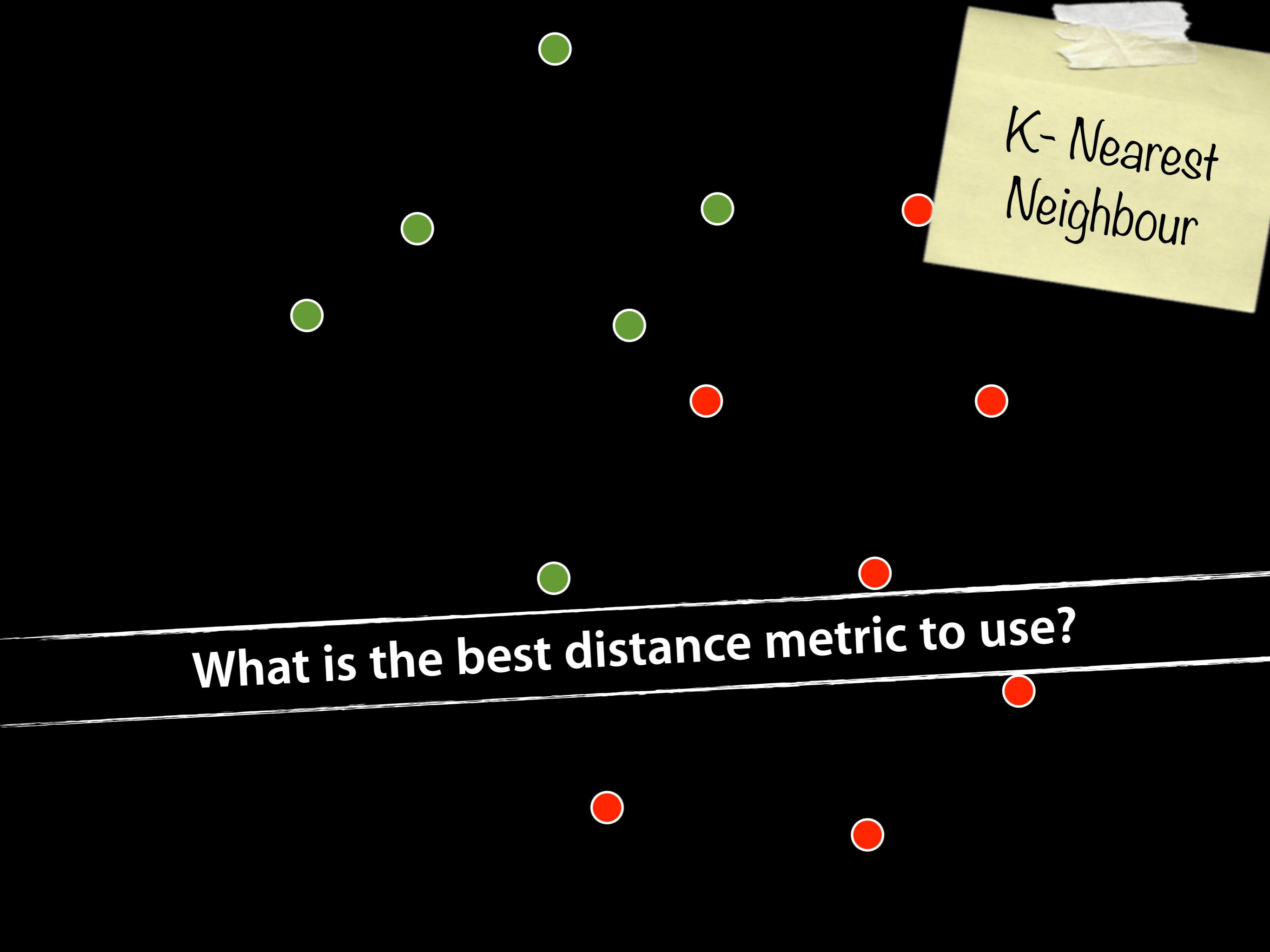


testing data

Training Data

Test

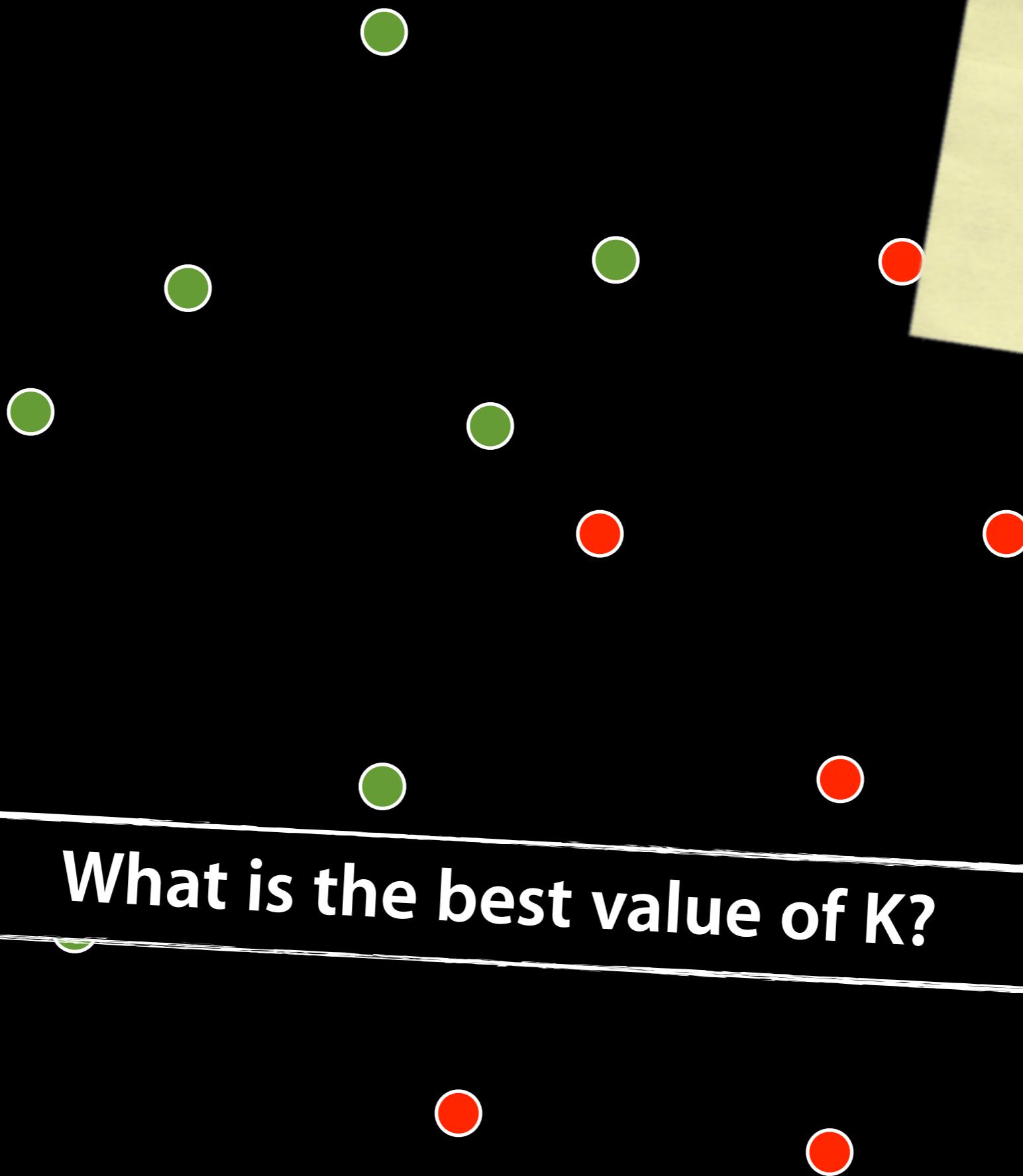
How do we set the hyperparameters?



K-Nearest
Neighbour

What is the best distance metric to use?

K-Nearest
Neighbour



What is the best value of K?

Linear Classifiers

$$L = \frac{1}{N} \sum_{i=1}^N \max(0, 1 - y_i [\mathbf{w} \cdot \mathbf{x}_i + b]) + \lambda R(\mathbf{w})$$

What is the best regularizer?

Linear Classifiers

$$L = \frac{1}{N} \sum_{i=1}^N \max(0, 1 - y_i [\mathbf{w} \cdot \mathbf{x}_i + b]) - \lambda R(\mathbf{w})$$

What is the best regularization parameter?

Training Data

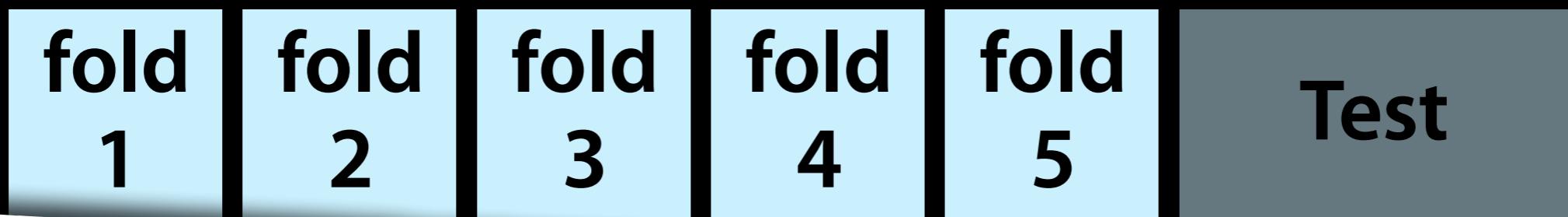


Test

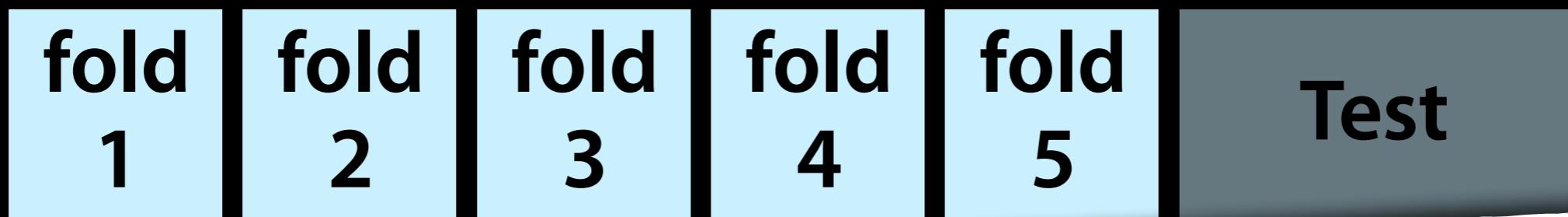
Validation

Training Data

Test

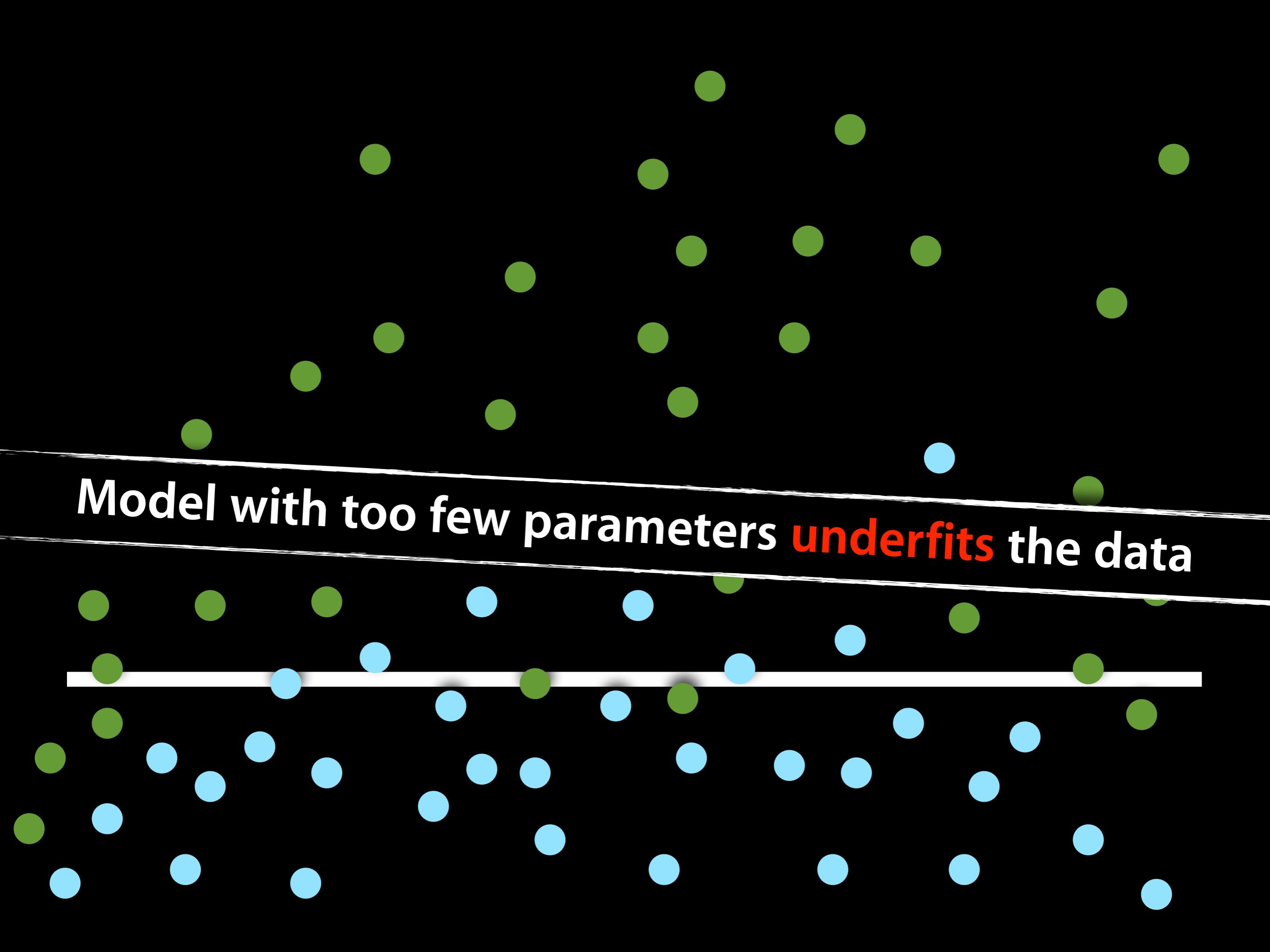


Use a fold as validation data and train on other folds



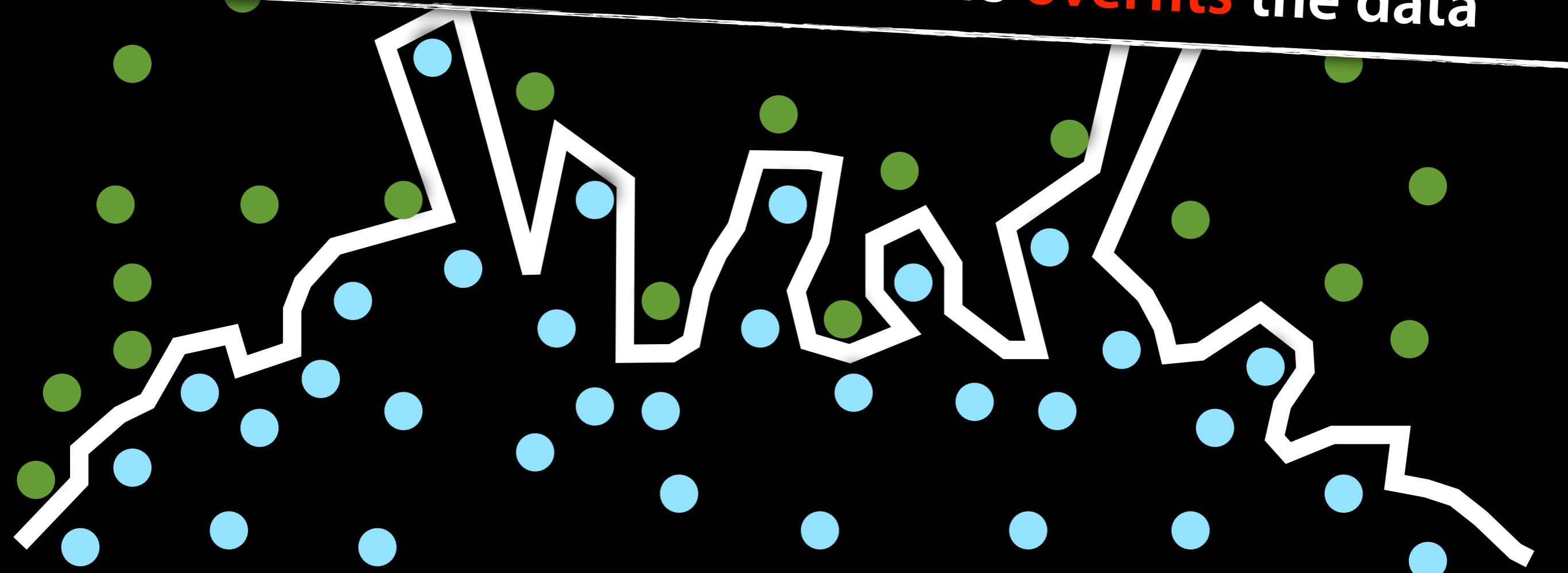
Cross-validation

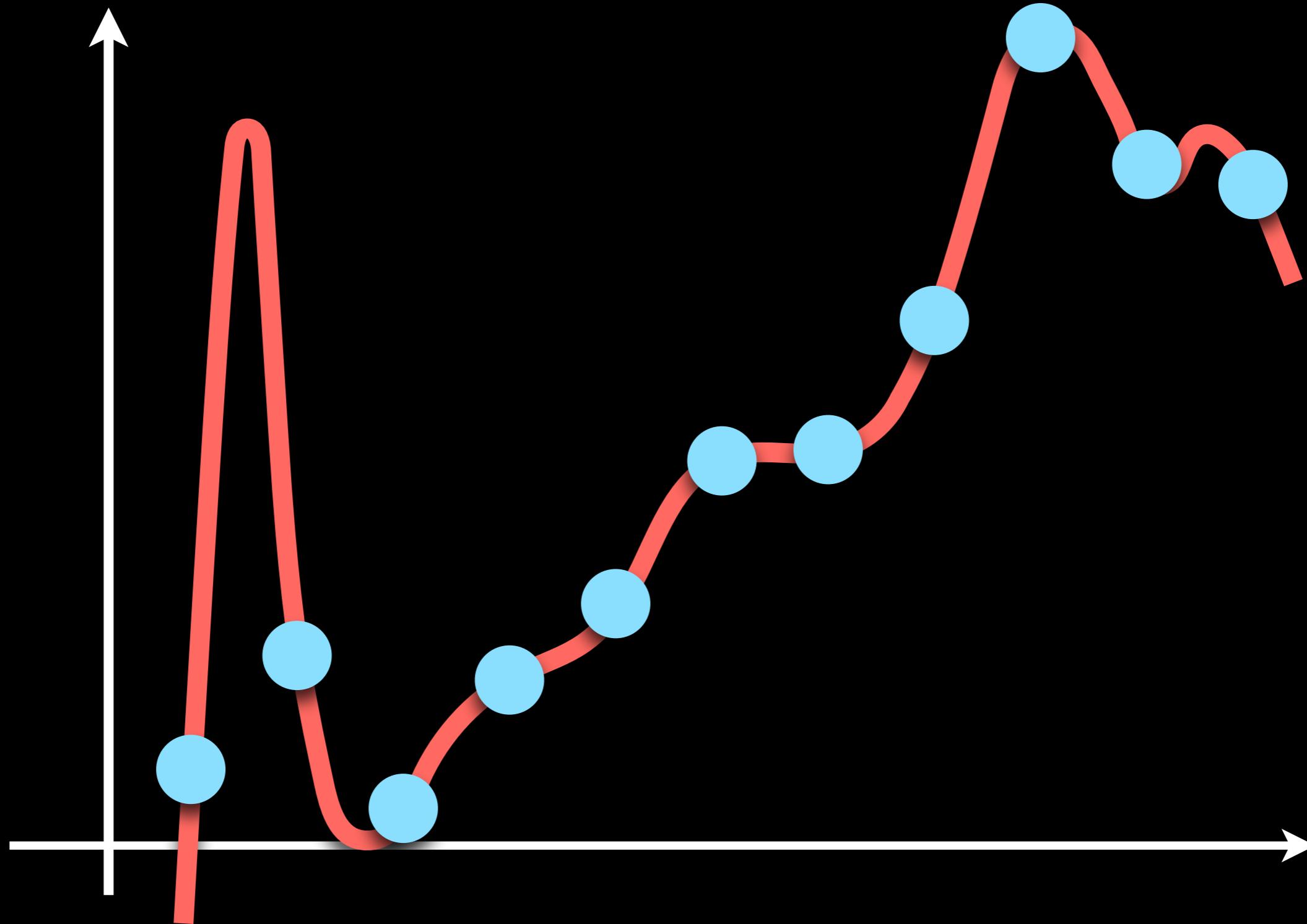
Select each fold in turn as validation data and average results

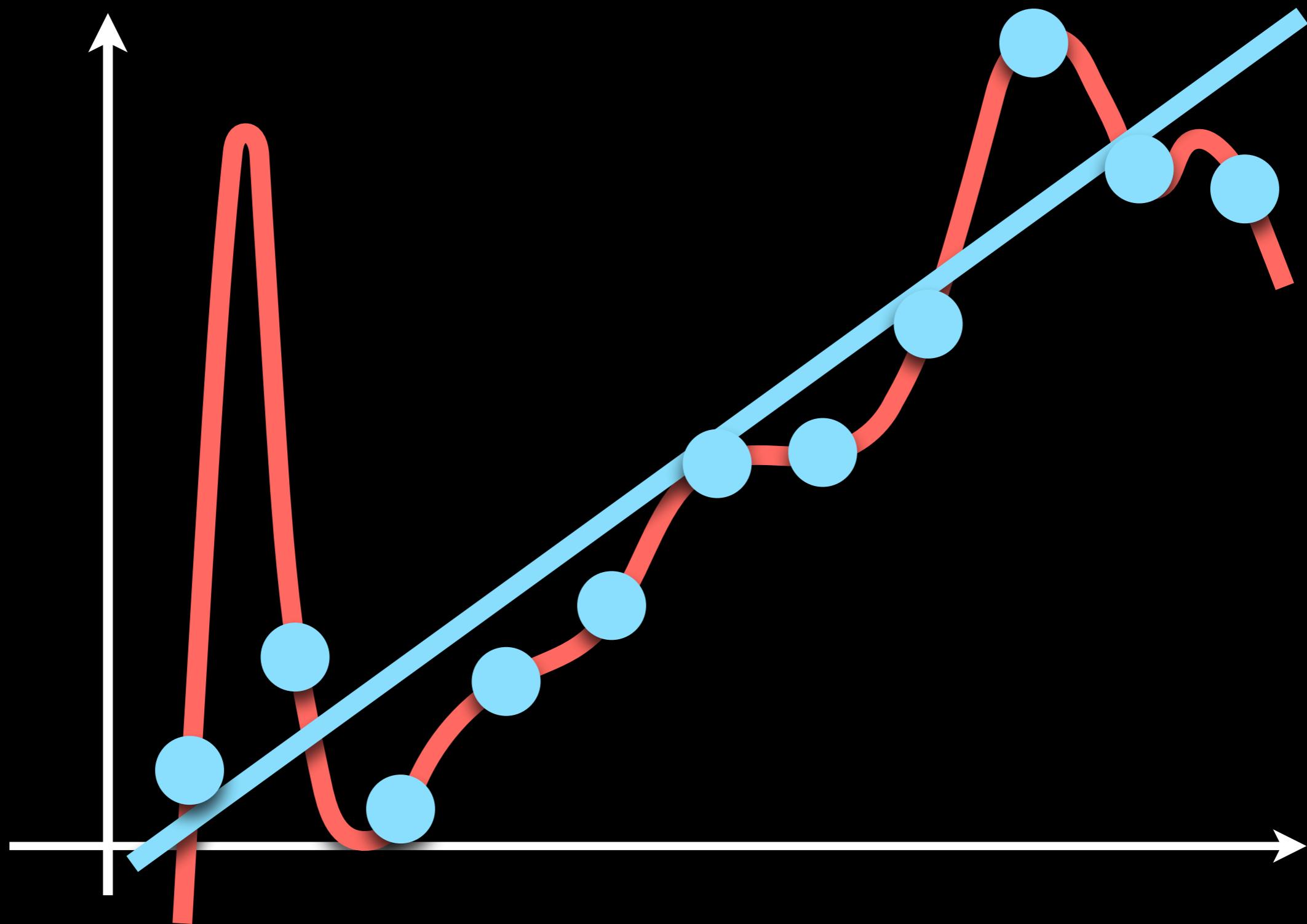


Model with too few parameters *underfits* the data

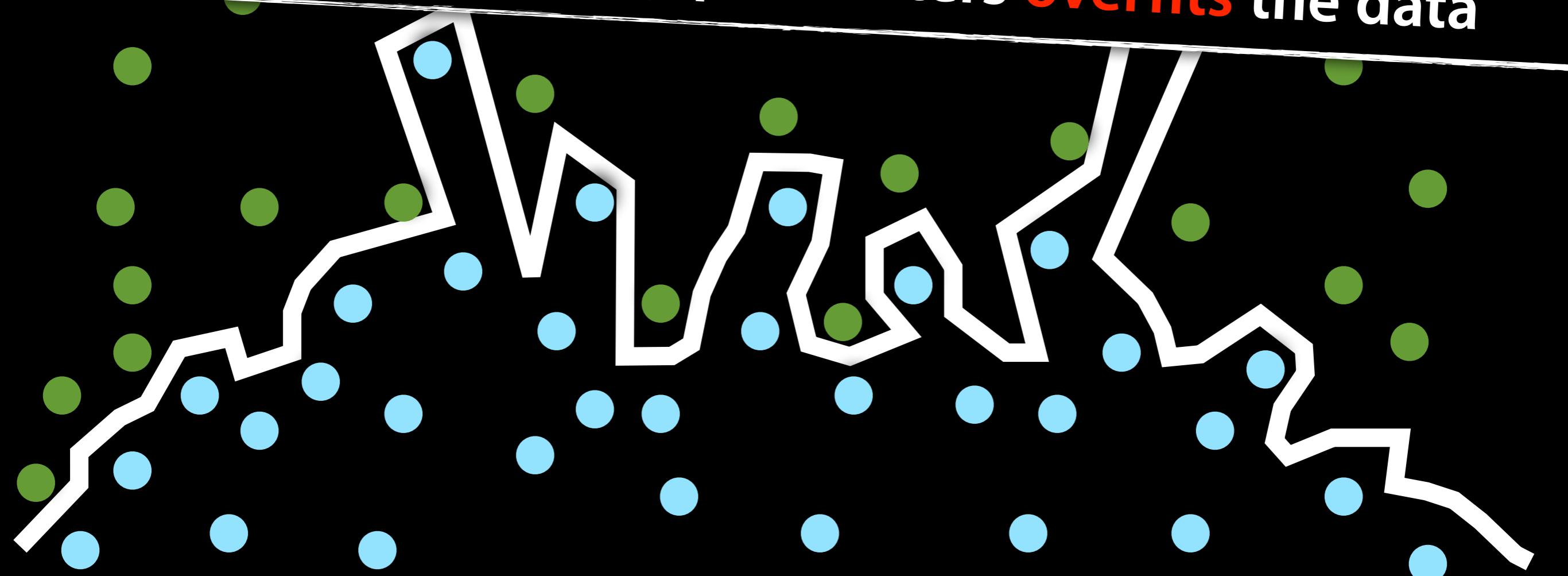
Model with too many parameters **overfits** the data



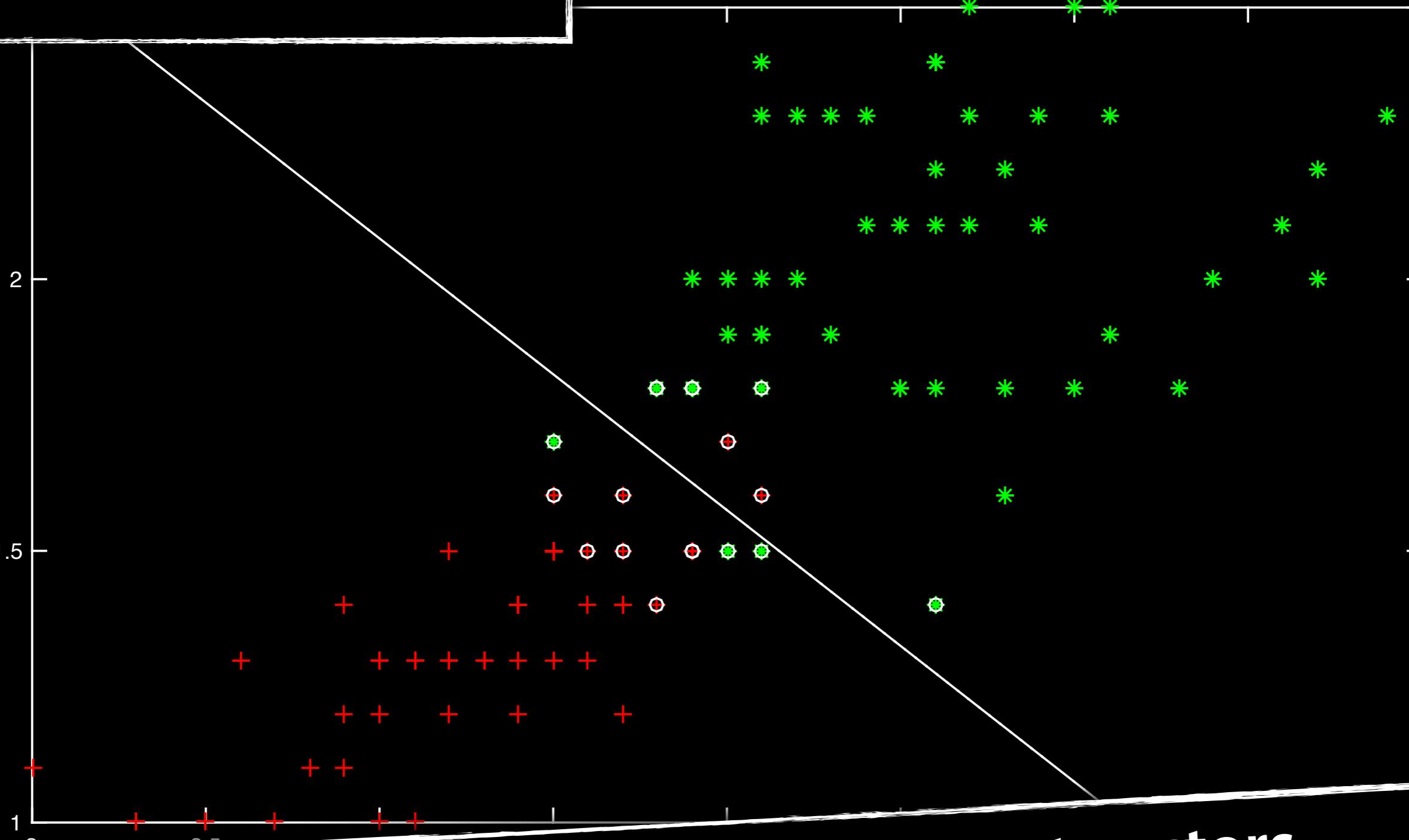




Model with too many parameters **overfits** the data



SVM classification



circled features denote support vectors

```
load fisheriris

% --- training phase
features = meas(51:end, 3:4);
labels = species(51:end);
svmStruct = svmtrain(features, labels, ...
    'ShowPlot', true);

% --- test phase
testDataSet = meas(1:50, 3:4);

predictClass = svmclassify(svmStruct, ...
    testDataSet);
```

Object Detection





Where are all the pedestrians?

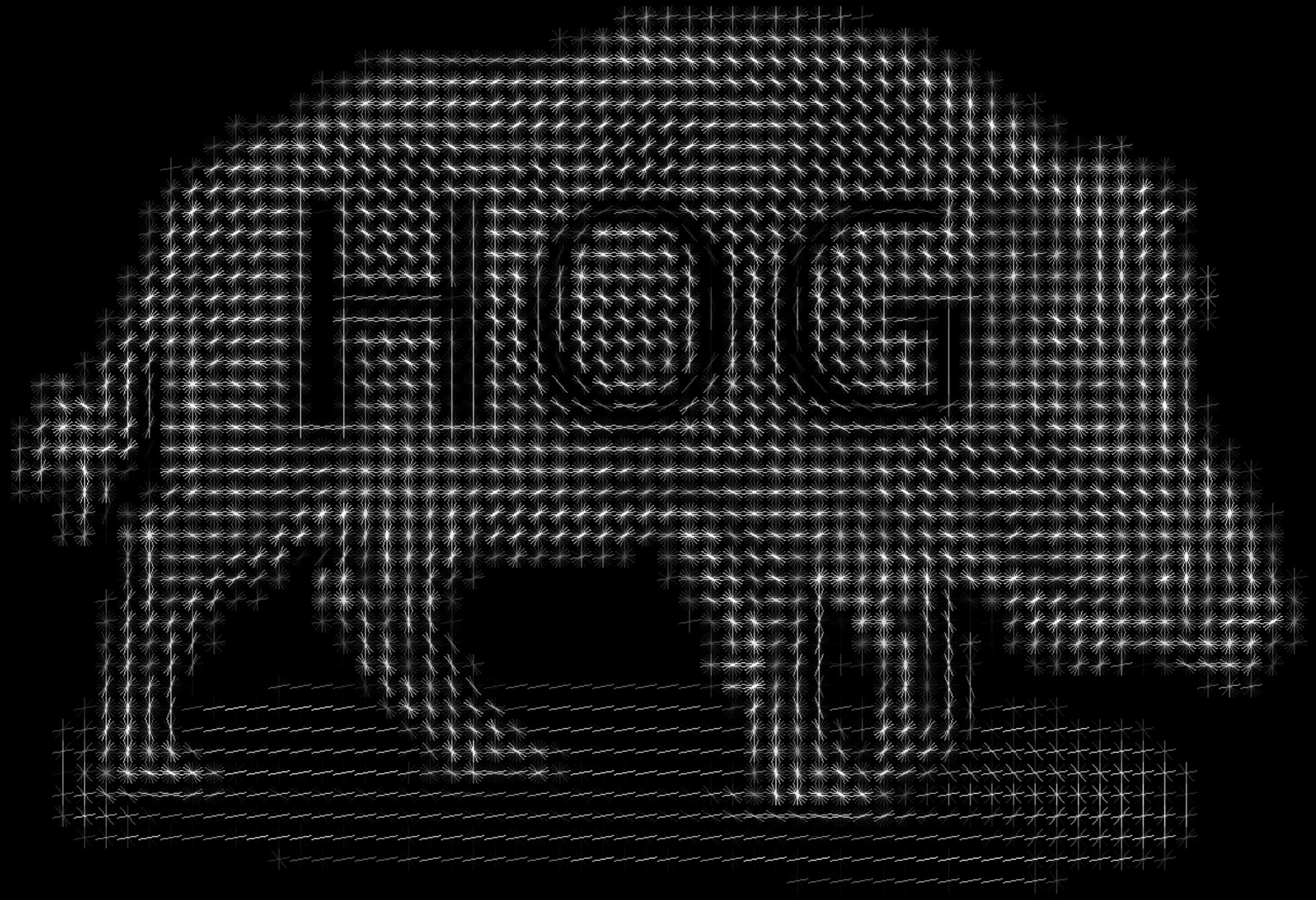
4

major steps

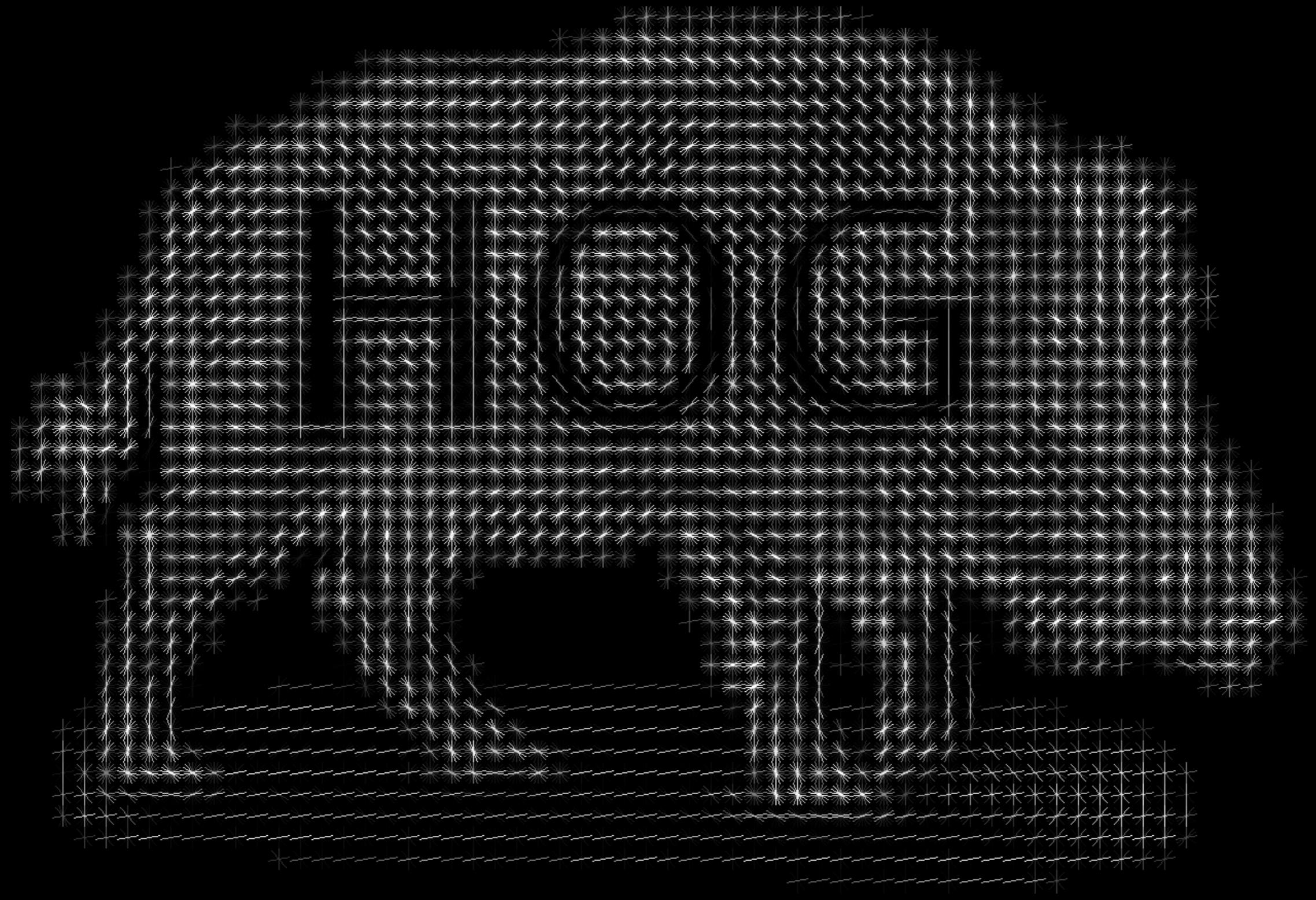
object detection using sliding windows

Step 1

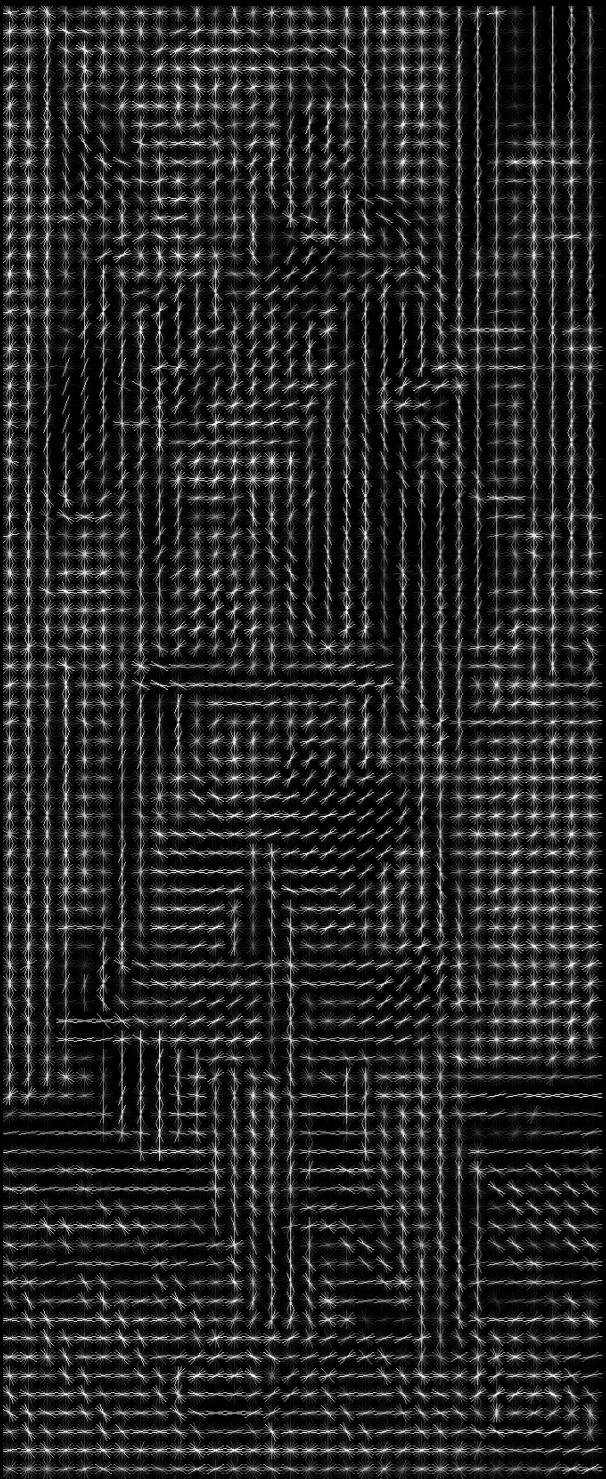
Select feature set



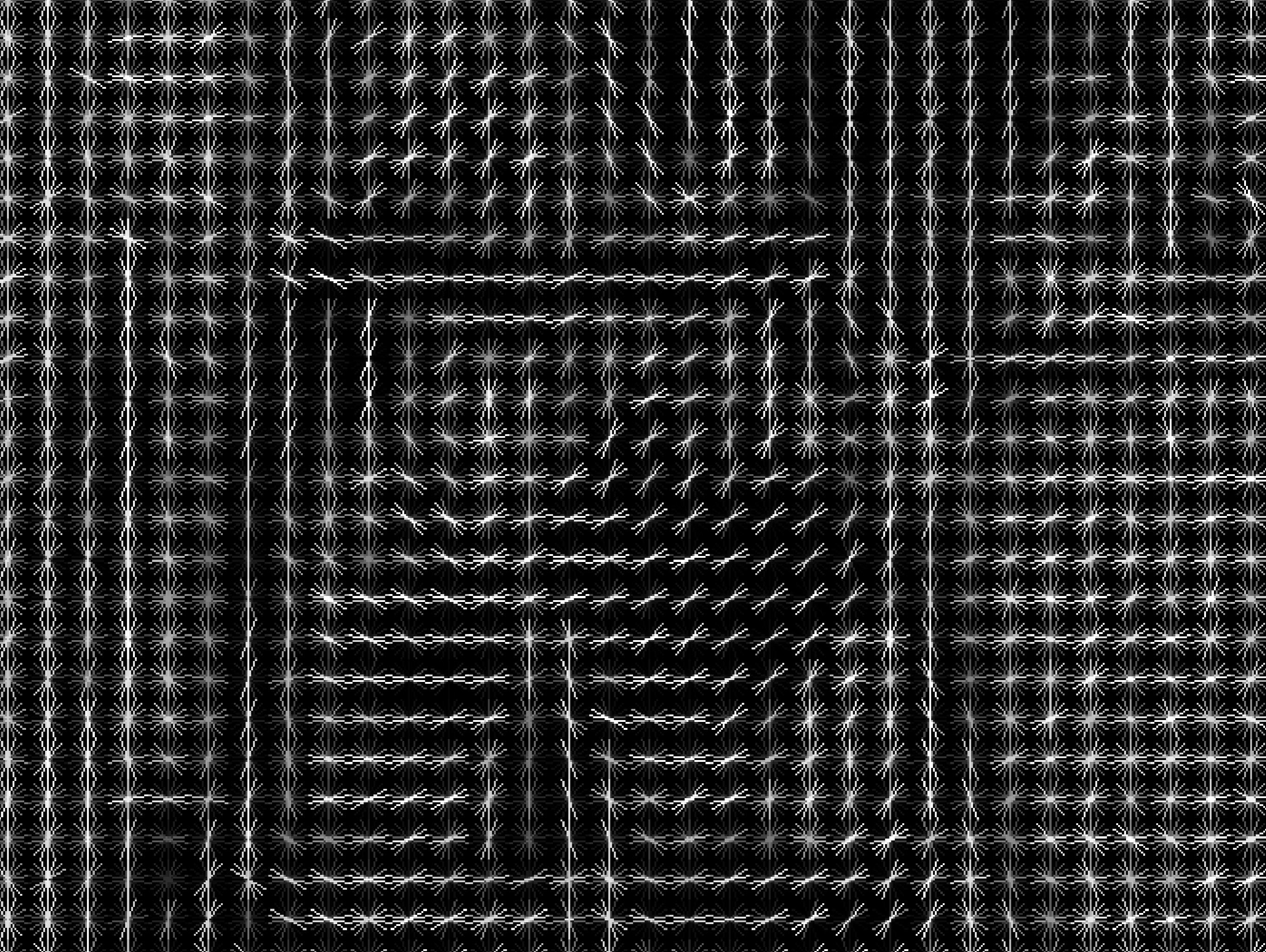
Histogram of Oriented Gradients



Histogram of Oriented Gradients



**compute histograms of gradients
in non-overlapping blocks**



SIFT
Descriptor

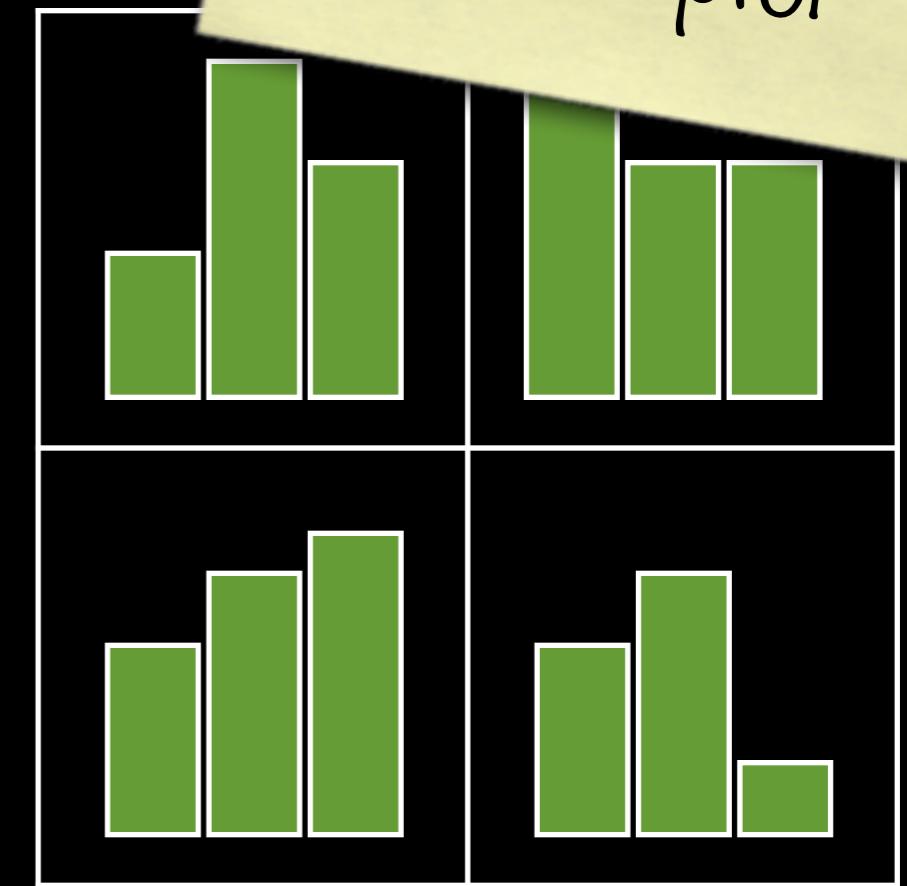
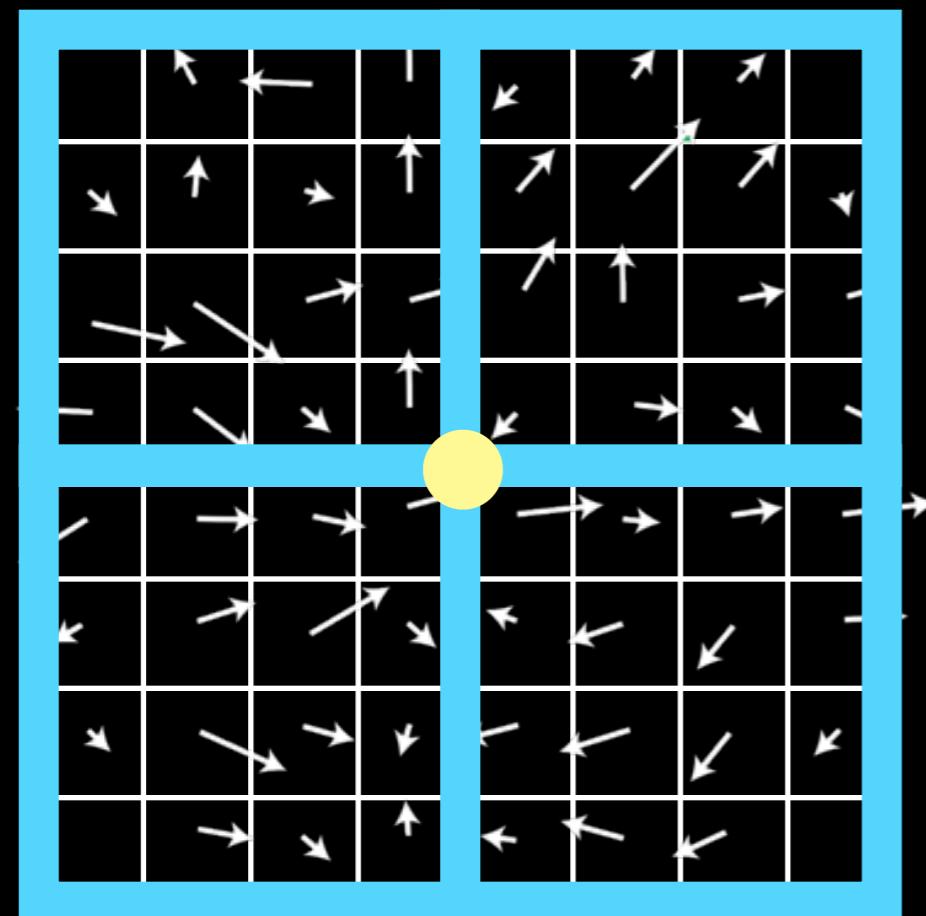
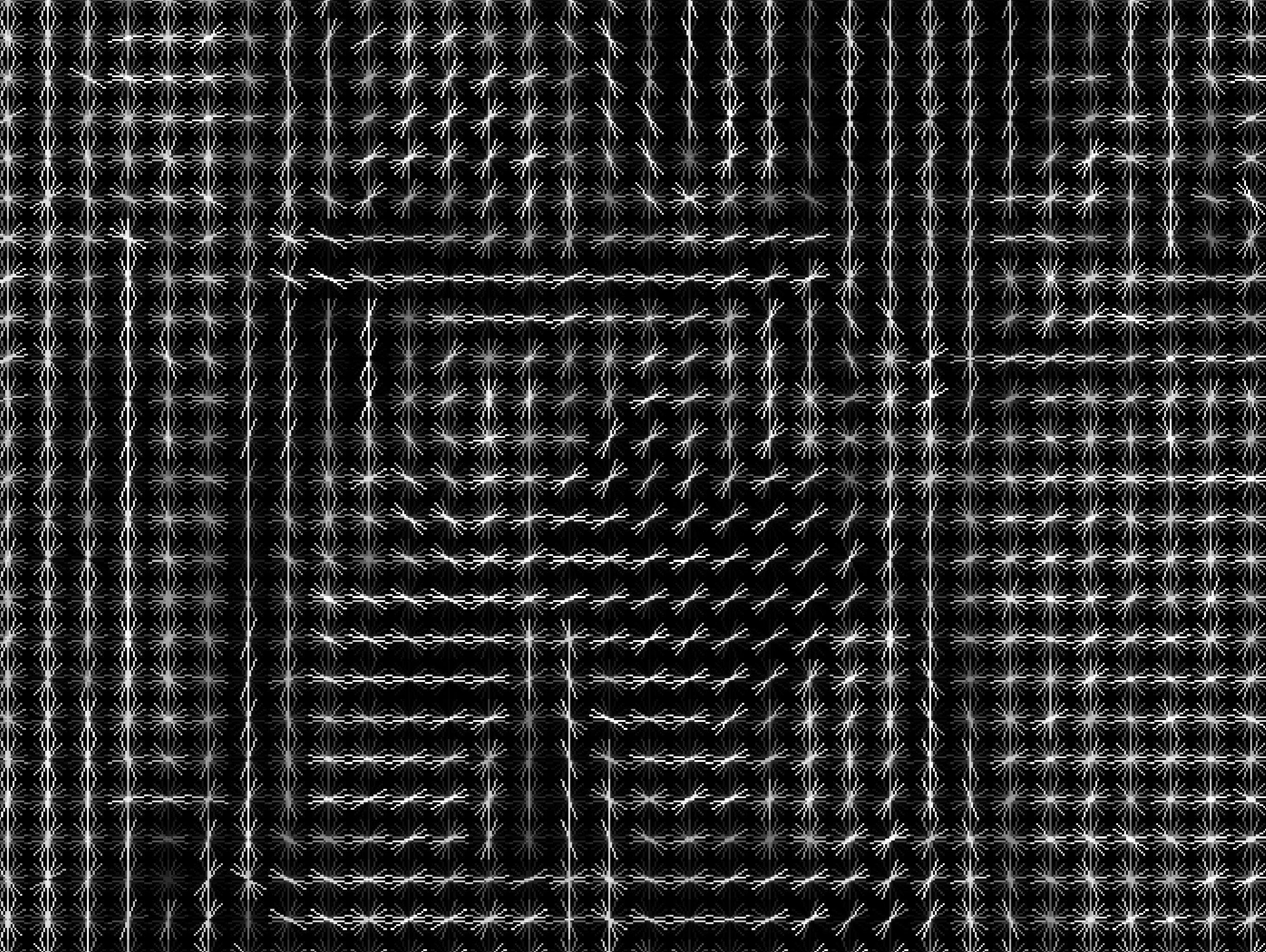
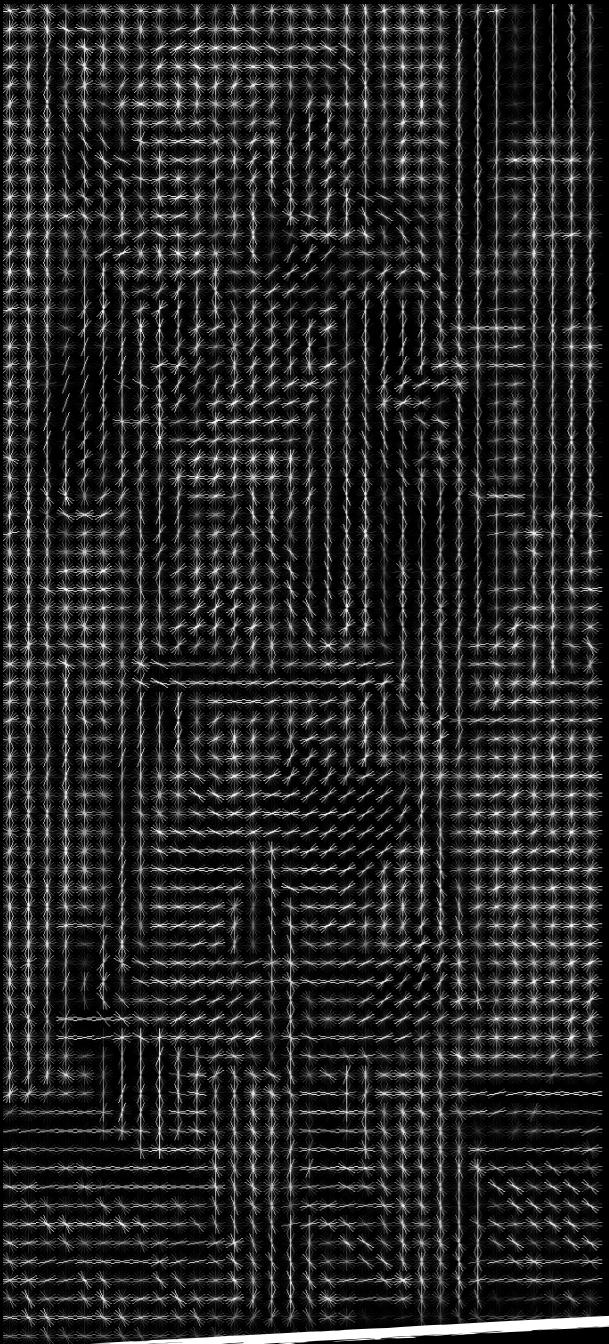


image gradients

keypoint descriptor

16 cells x 8 bins = 128 dimensional descriptor



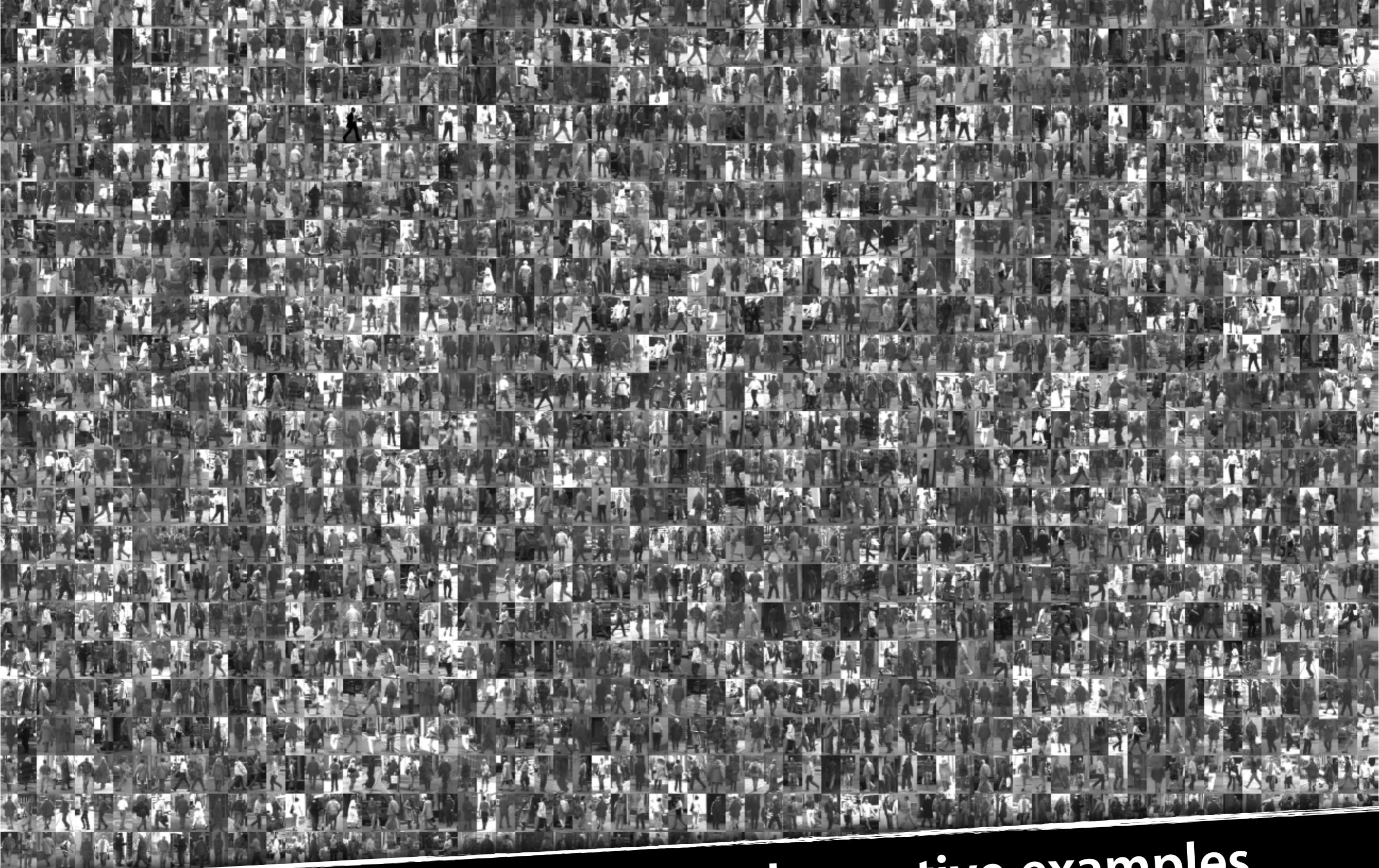


**collect HOGs over all blocks and
form a feature vector**

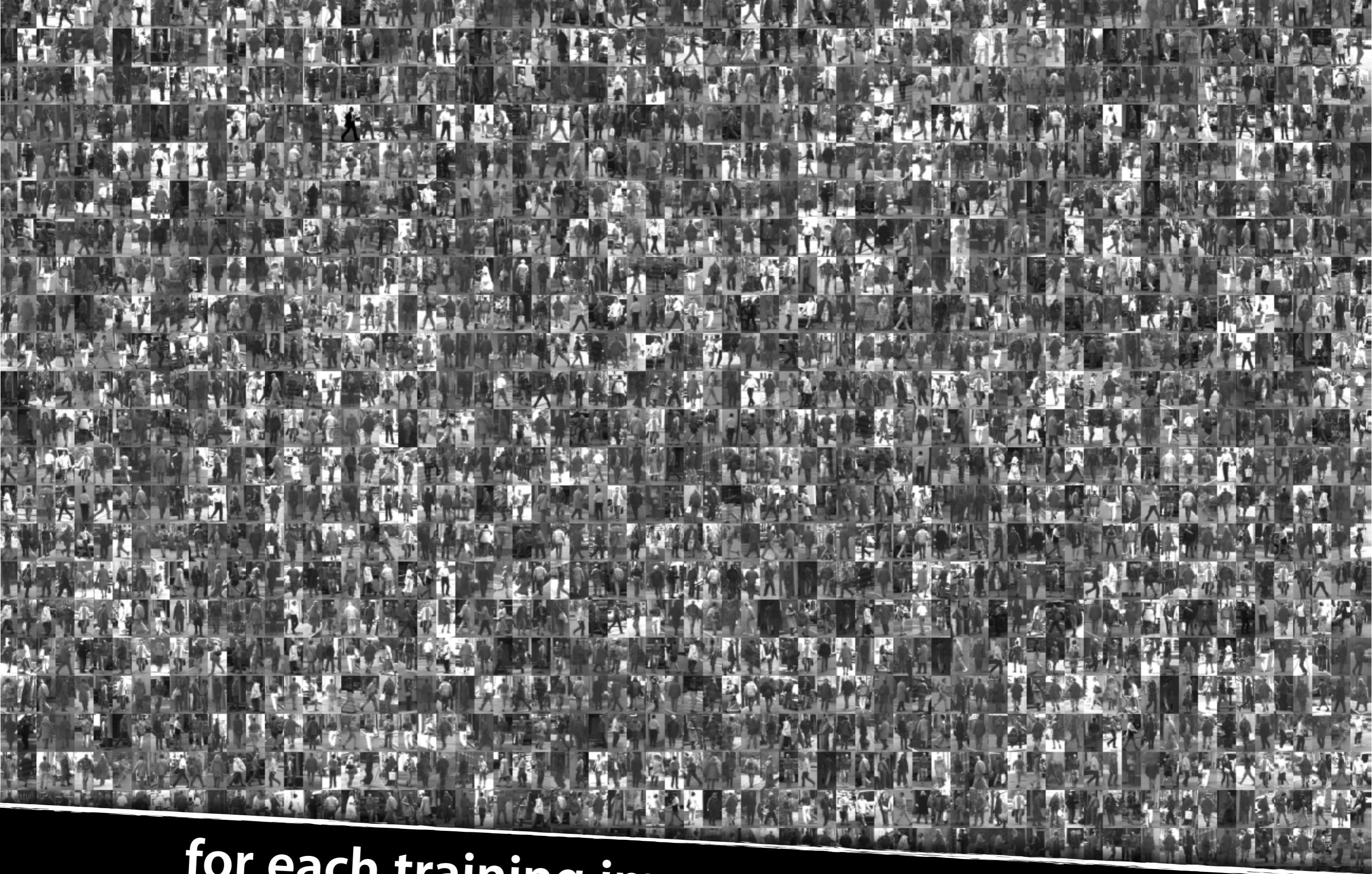
Dalal-Triggs feature vector is 3780 dimensional

Step 2

Build object model



training set of positive and negative examples



for each training image compute features

$$L = \frac{1}{N} \sum_{i=1}^N \max(0, 1 - y_i [\mathbf{w} \cdot \mathbf{x}_i + b]) + \lambda R(\mathbf{w})$$

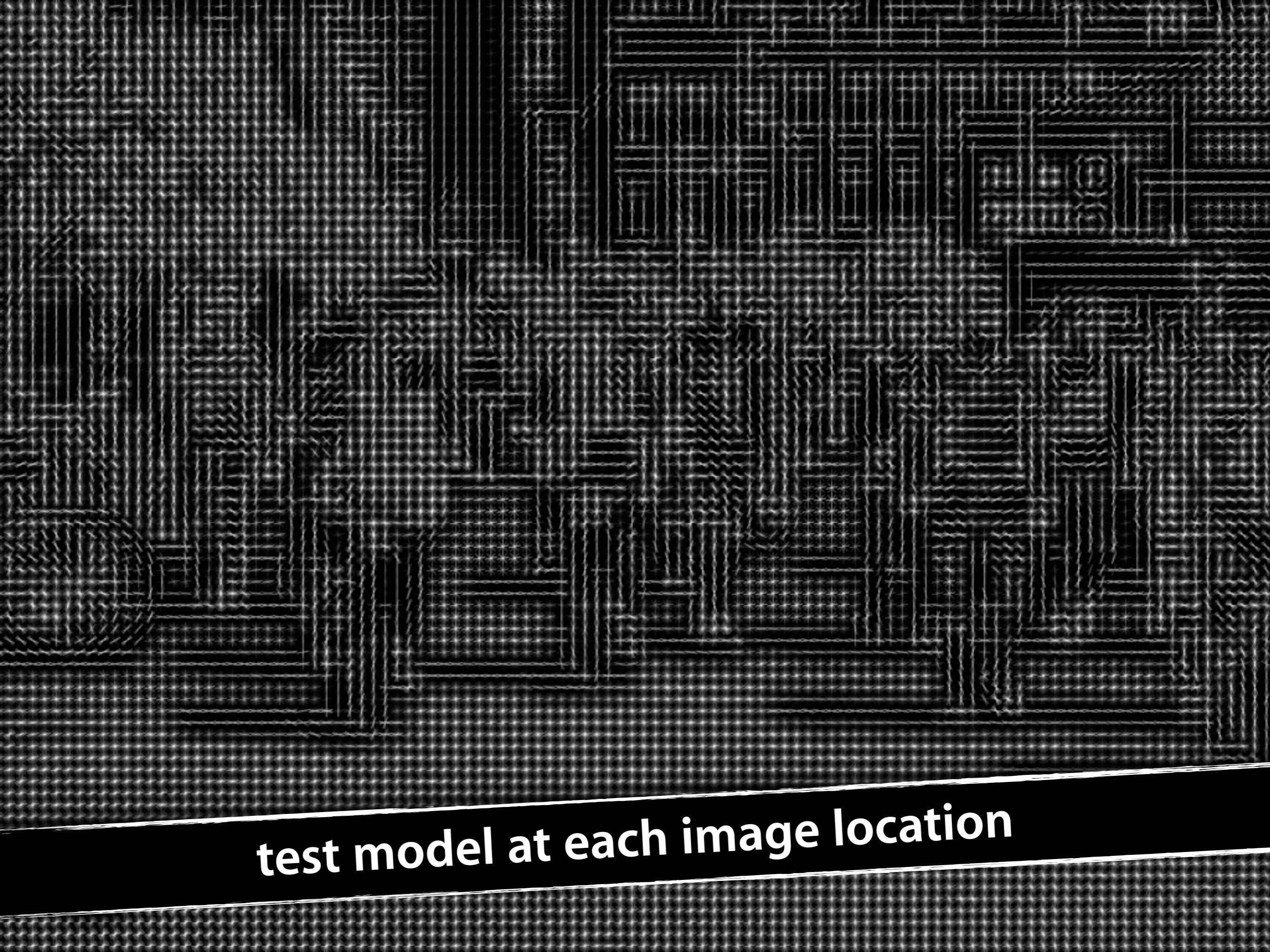
support vector machine (SVM) classifier

Step 3

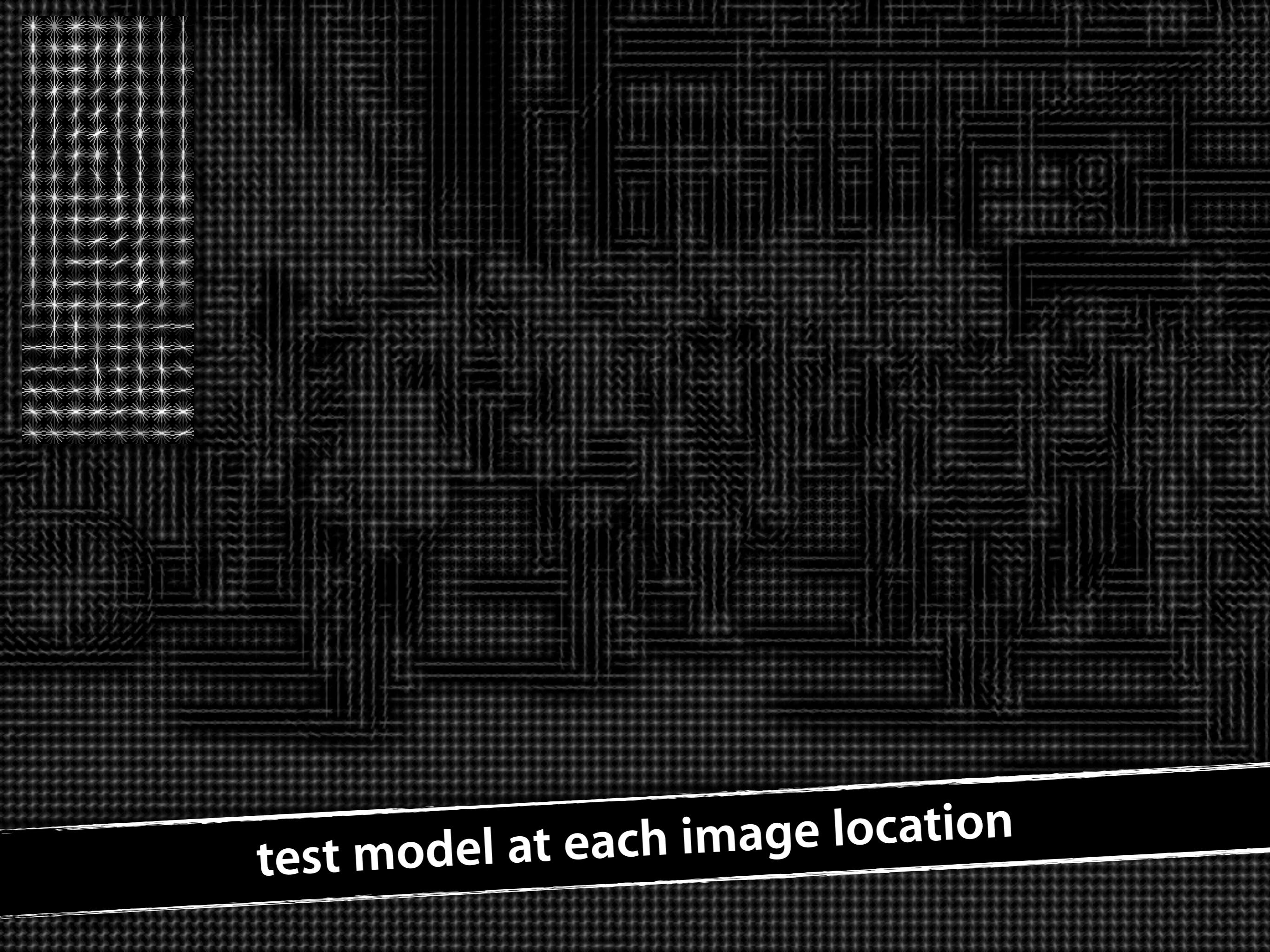
Score image locations

Public
Parking



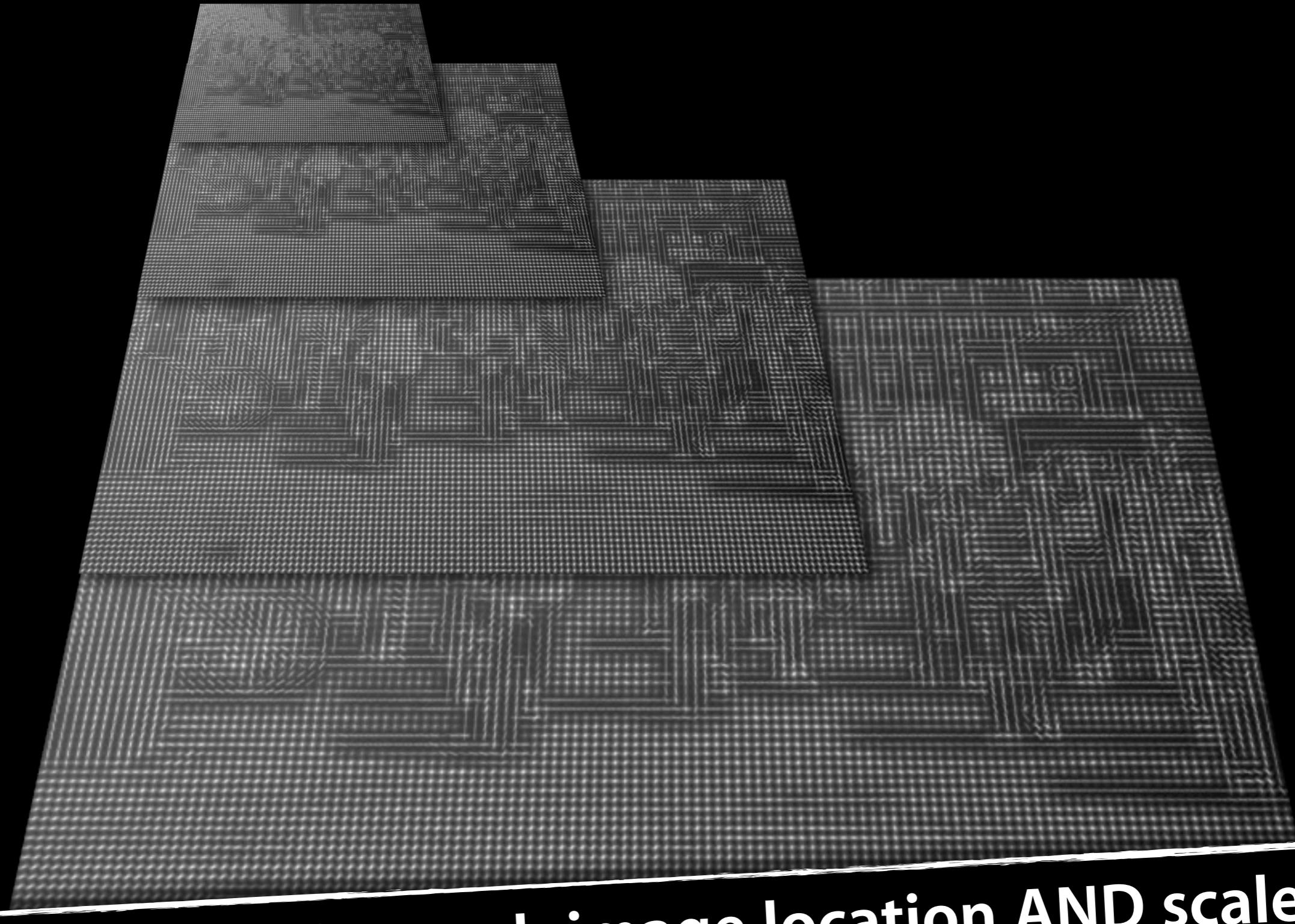


test model at each image location



test model at each image location

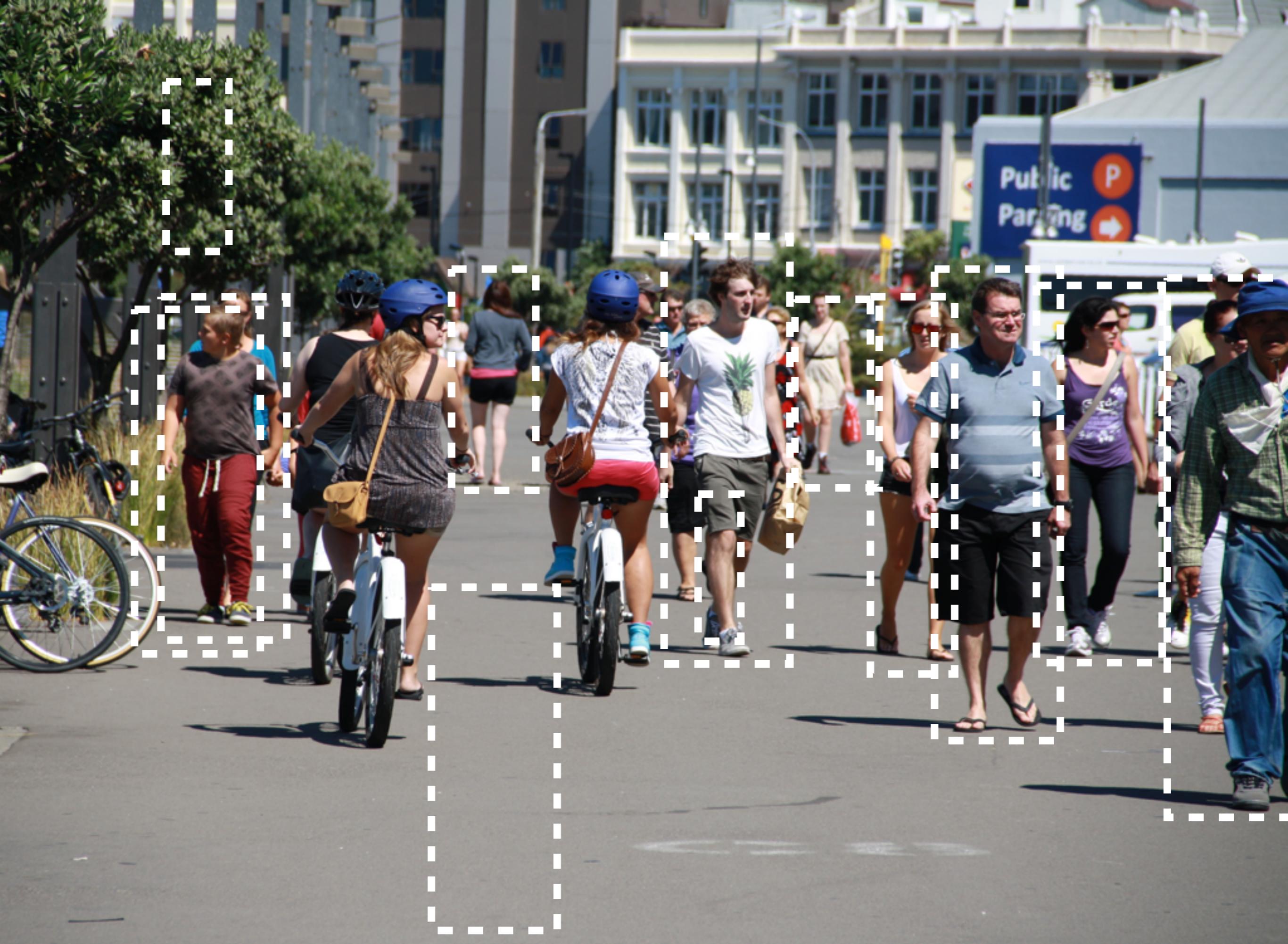
What if the instance size differs from the model?



test model at each image location AND scale

Step 4

Resolve detections

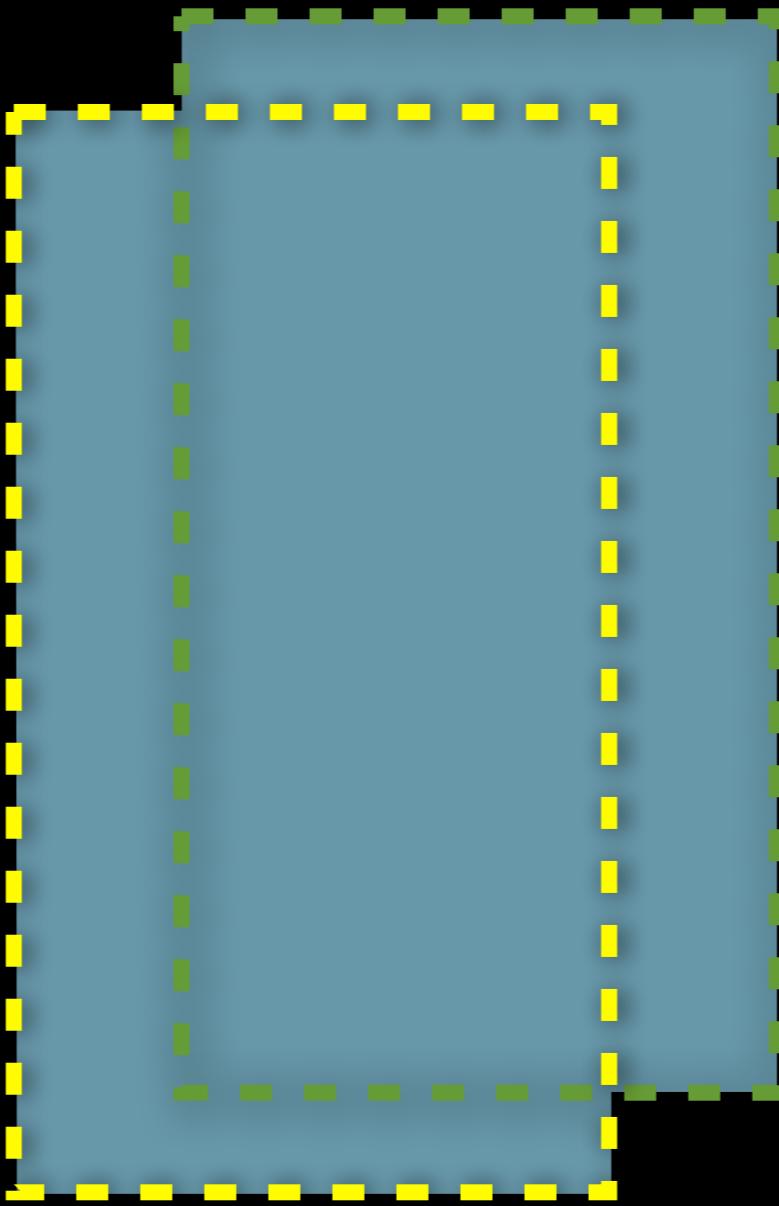




score = 0.8

score = 0.1

eliminate overlapping bounding boxes using
non-maximum suppression



$$\text{overlap} = \frac{\text{area}(\text{box}_1 \cap \text{box}_2)}{\text{area}(\text{box}_1 \cup \text{box}_2)}$$



$$\text{overlap} = \frac{\text{area}(\text{box}_1 \cap \text{box}_2)}{\text{area}(\text{box}_1 \cup \text{box}_2)} > \text{threshold}$$

threshold is typically set to 50%



score = 0.8

eliminate overlapping bounding boxes using
non-maximum suppression

Given a test set, how do we evaluate performance?

ground truth



true positive

$$\frac{\text{area}(\text{box}_1 \cap \text{box}_2)}{\text{area}(\text{box}_1 \cup \text{box}_2)} > .5$$

detection

precision



recall: percentage of true positives detected



recall

precision

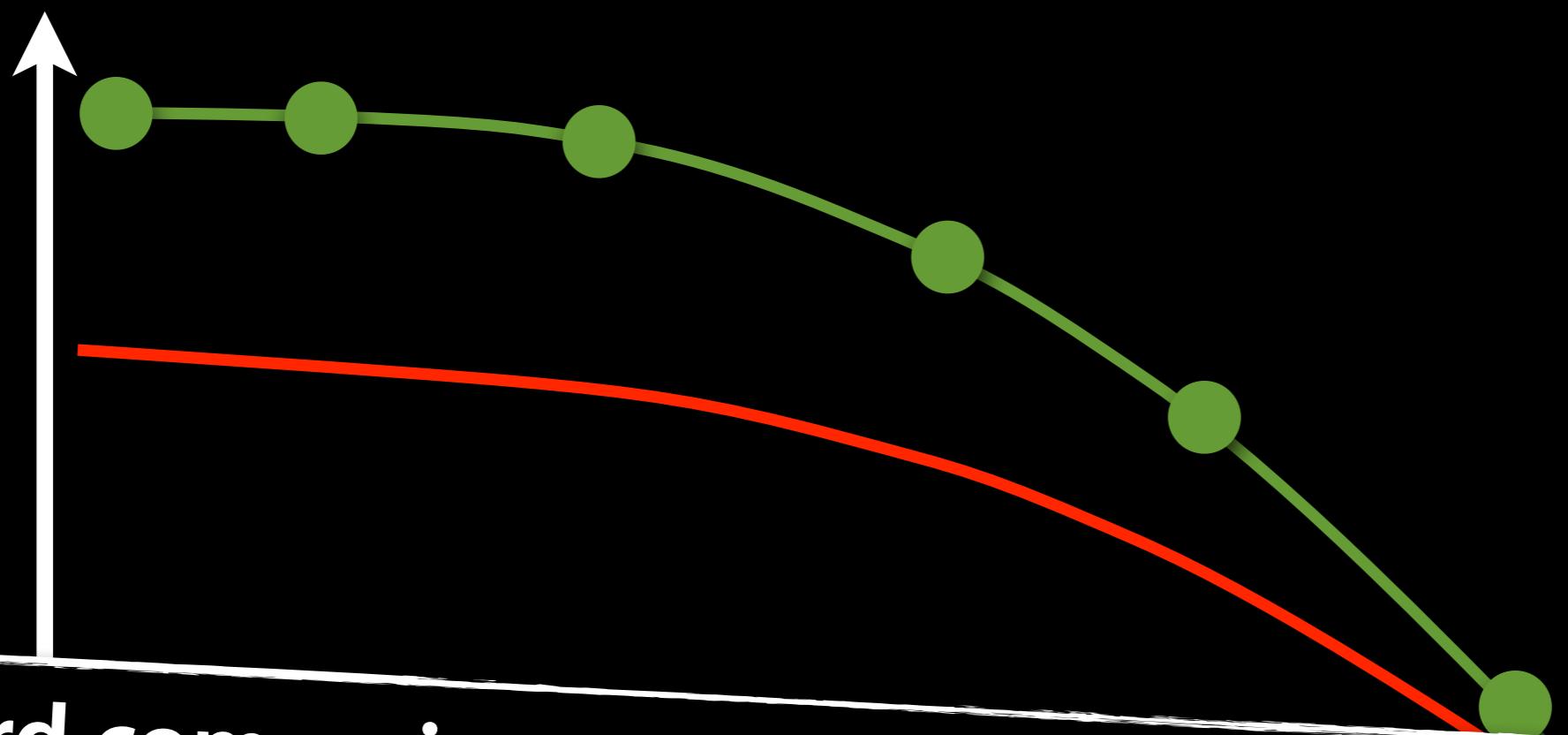


precision: percentage of detections that are correct

recall



precision



standard comparison measure is **Average Precision (AP)**

precision

