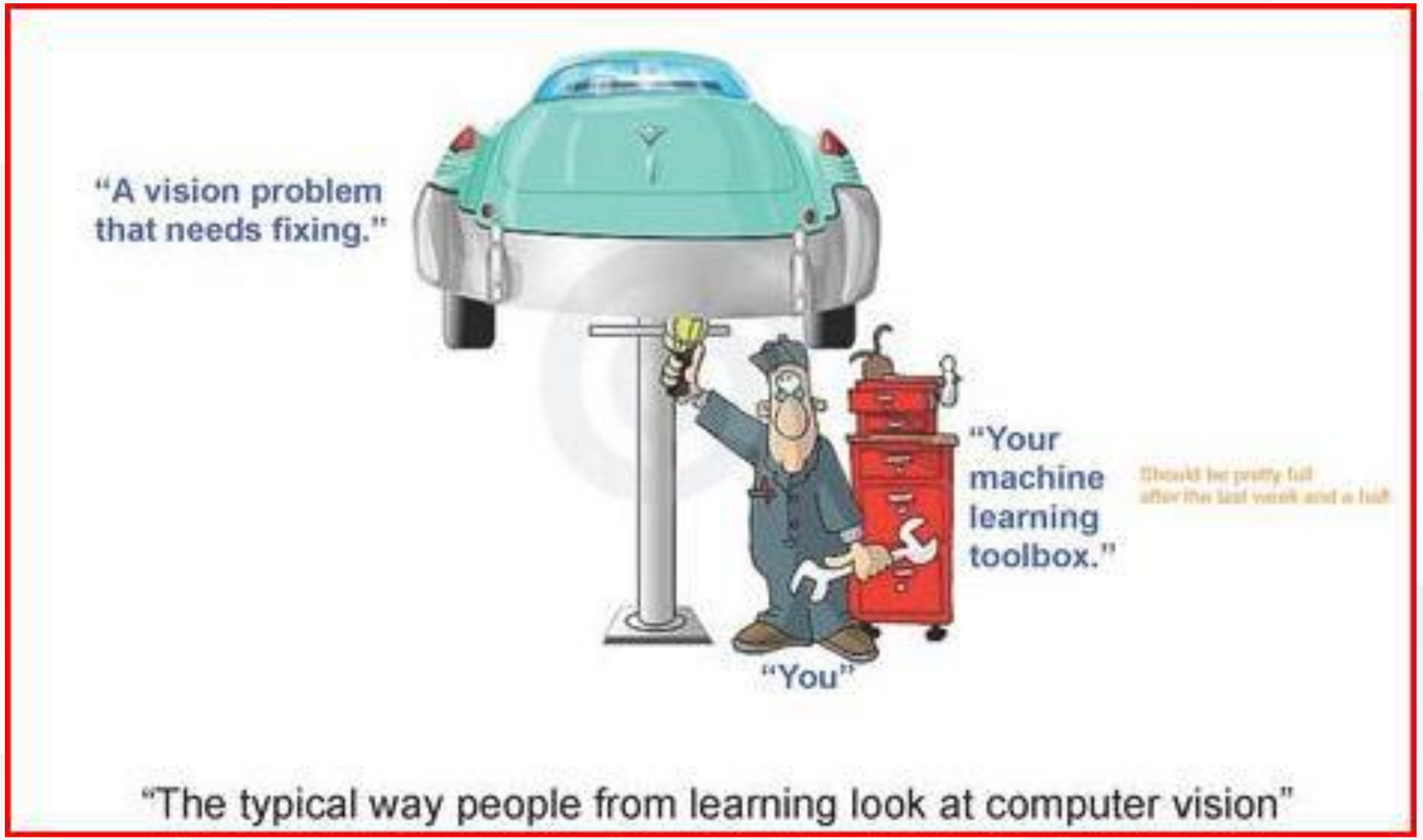
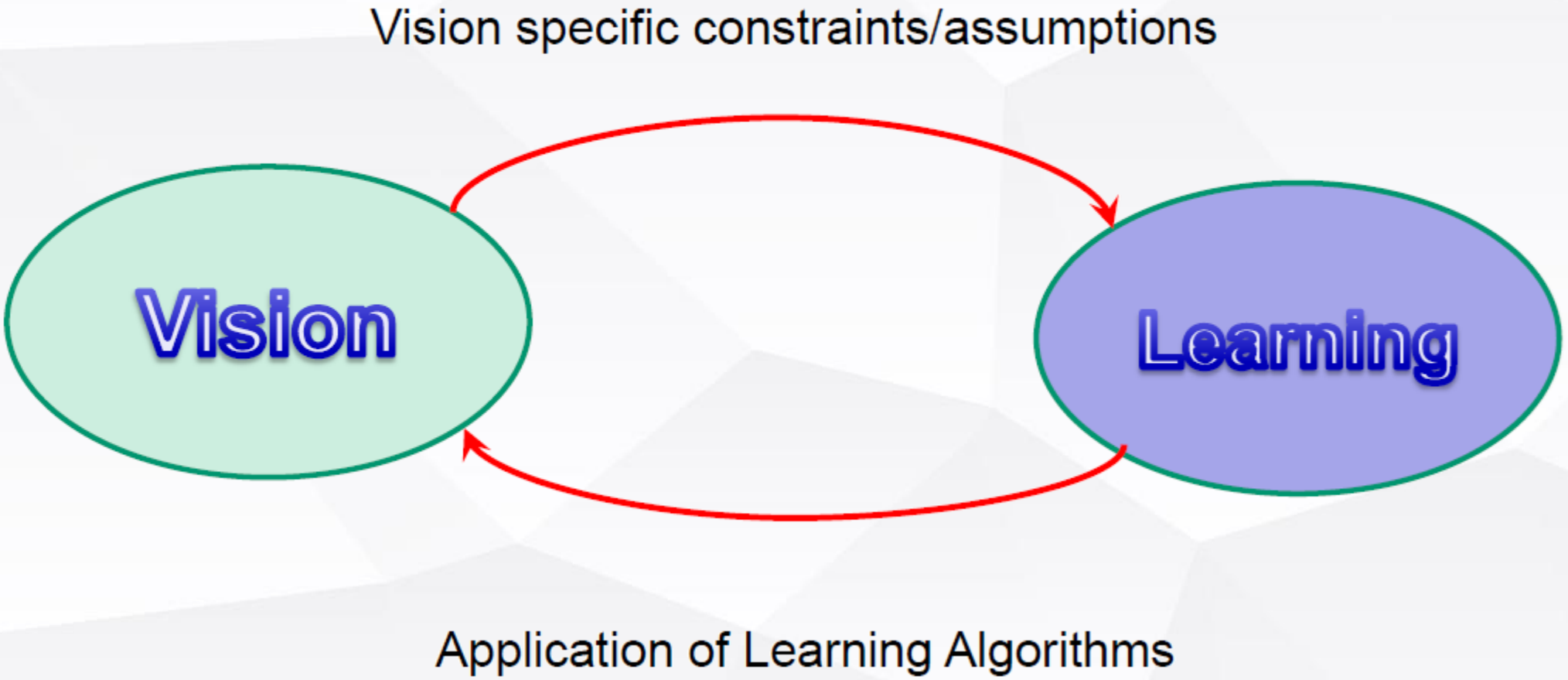
Machine Learning for Computer Vision

俞智斌
yuzhibin@ouc.edu.cn

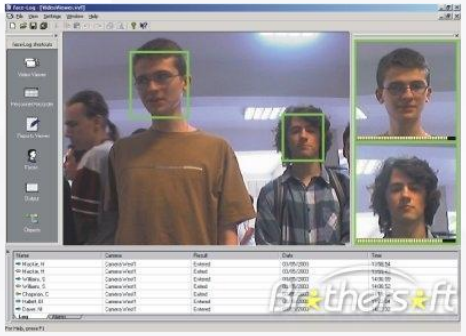
中国海洋大学电子工程系

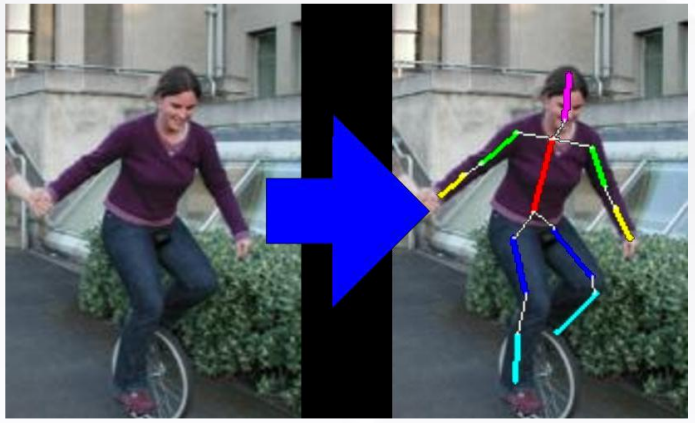

"A vision problem that needs fixing."
"Your machine learning toolbox."
"You"
"The typical way people from learning look at computer vision"


Vision specific constraints/assumptions
Vision
Learning
Application of Learning Algorithms

Machine Learning for Computer vision

- **Why** Machine Learning?
  - We cannot program everything
  - Some tasks are difficult to define algorithmically
    - especially in computer vision
    - … visual sensing has few rules
- Well-defined learning problems?
  - Easy to learning Vs. difficult to learn
  - Varying complexity of visual patterns
- An example: learning to recognize objects…





Machine Learning

- Learning in humans

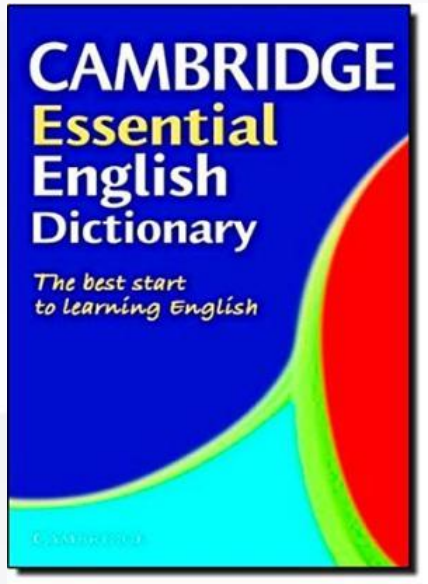
Machine Learning for Computer vision

- Learning in computers

Machine Learning for Computer vision

- Definition of learning(verb)

- the *activity* of **obtaining knowledge**
- knowledge **obtained by study**



English Dictionary, Cambridge University Press

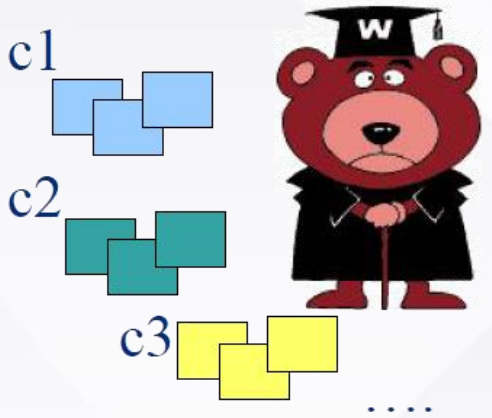
Machine Learning for Computer vision

- Definition of Machine **Learning**
- A set of methods for the automated analysis of structure in data. …. two main strands of work, (i) **unsupervised** learning …. and (ii) **supervised** learning.
- ….similar to ... data mining, but ... focus .. more on autonomous machine performance, …. rather than enabling humans to learn from the data.

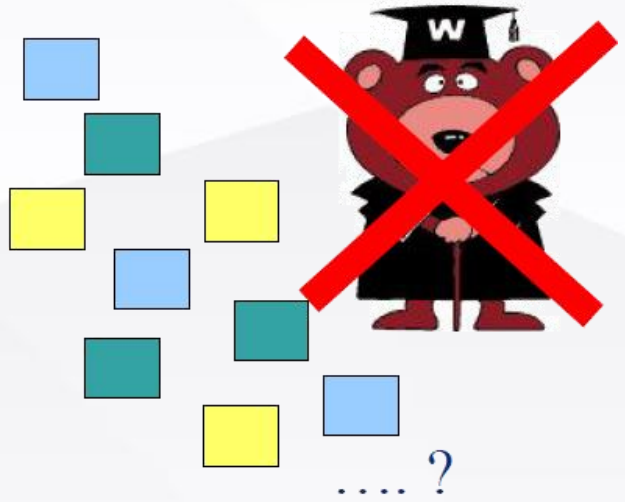
[Dictionary of Image Processing & Computer Vision, Fisher et al., 2014]

Machine Learning for Computer vision

- Supervised
  - **knowledge of output** - learning with the presence of an “expert” / teacher
    - data is **labelled** with a class or value
    - Goal: predict class or value label
      - e.g. Neural Network, Support Vector Machines, Decision Trees, Bayesian Classifiers ....



- Unsupervised
  - **no knowledge of output** class or value
    - data is **unlabelled** or value un-known
    - Goal: determine data patterns/groupings
  - Self-guided learning algorithm
    - (internal self-evaluation against some criteria)
    - e.g. k-means, genetic algorithms, clustering approaches ...



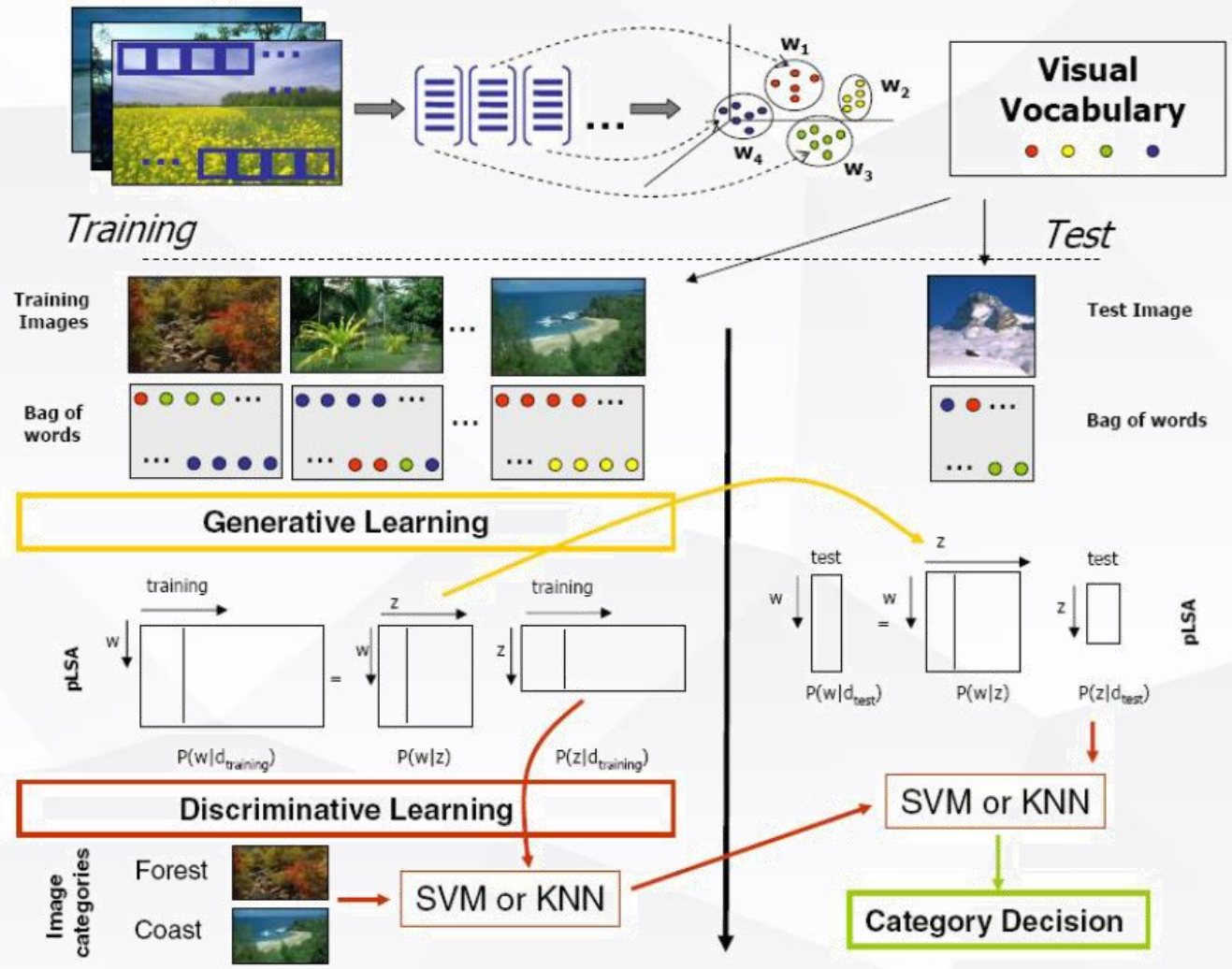
Machine Learning for Computer vision

- Supervised
  - Given a corpora of sample data of various scenes and their associated labels, classify the test data.

Training data with labels.

Machine Learning for Computer vision

- Supervised



Zissermann, PAMI‘09

Machine Learning for Computer vision

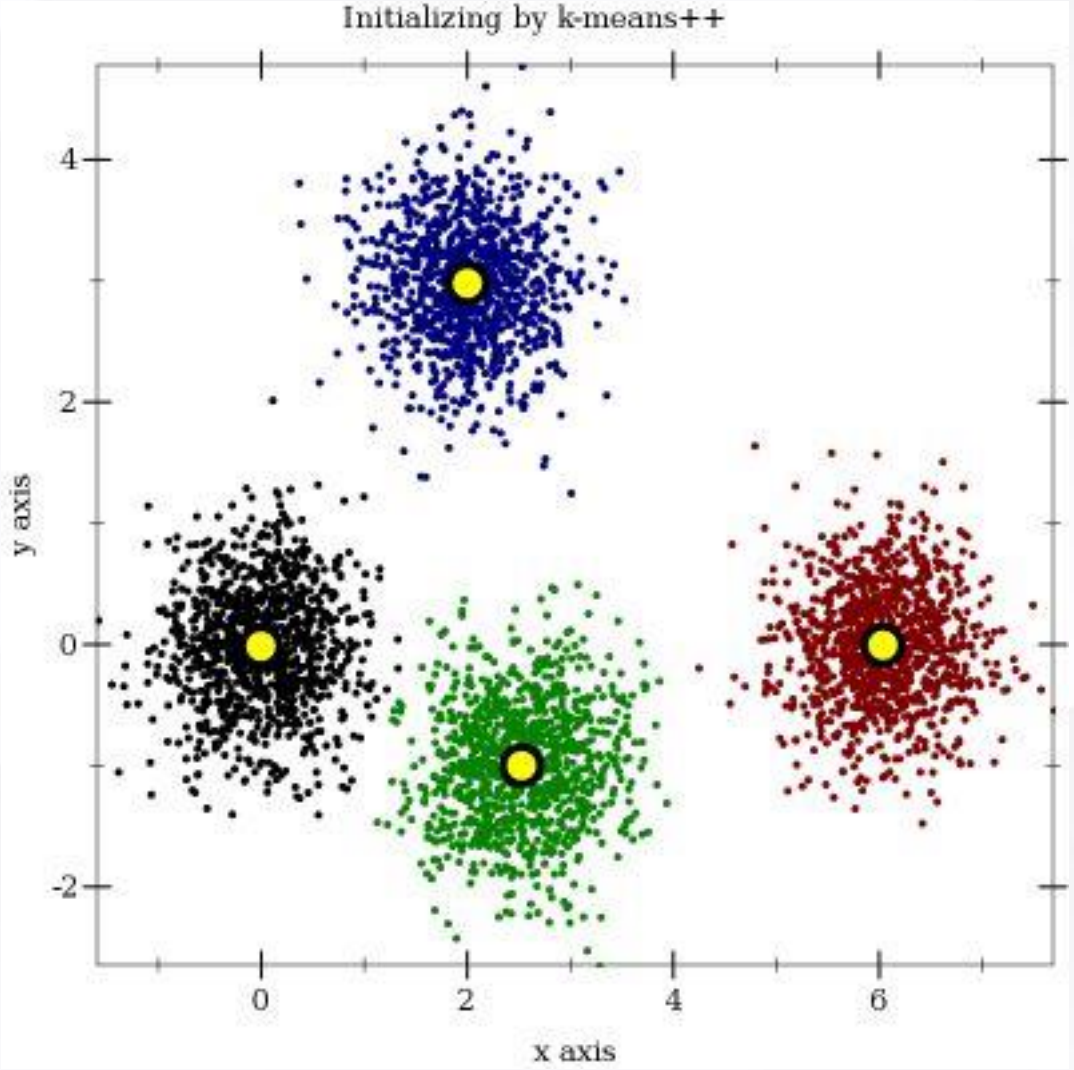
- Supervised problems
  - Unavailability of labeled data for training the classifier
    - Labeling data is boring
    - Experts might not be available (ex: medical imaging).
  - Number of topic categories might not be available (as in the case of scene classification mentioned earlier) or might increase with more data.
  - Solution: Unsupervised Learning.

Machine Learning for Computer vision

- Unsupervised learning
  - Learner is provided only unlabeled data.
  - No feedback is provided from the environment.
  - Aim of the learner is to find patterns in data which is otherwise observed as unstructured noise.
  - Commonly used UL techniques:
    - Clustering (K-Means, Mixture models, etc.).

Machine Learning for Computer vision

- Unsupervised learning

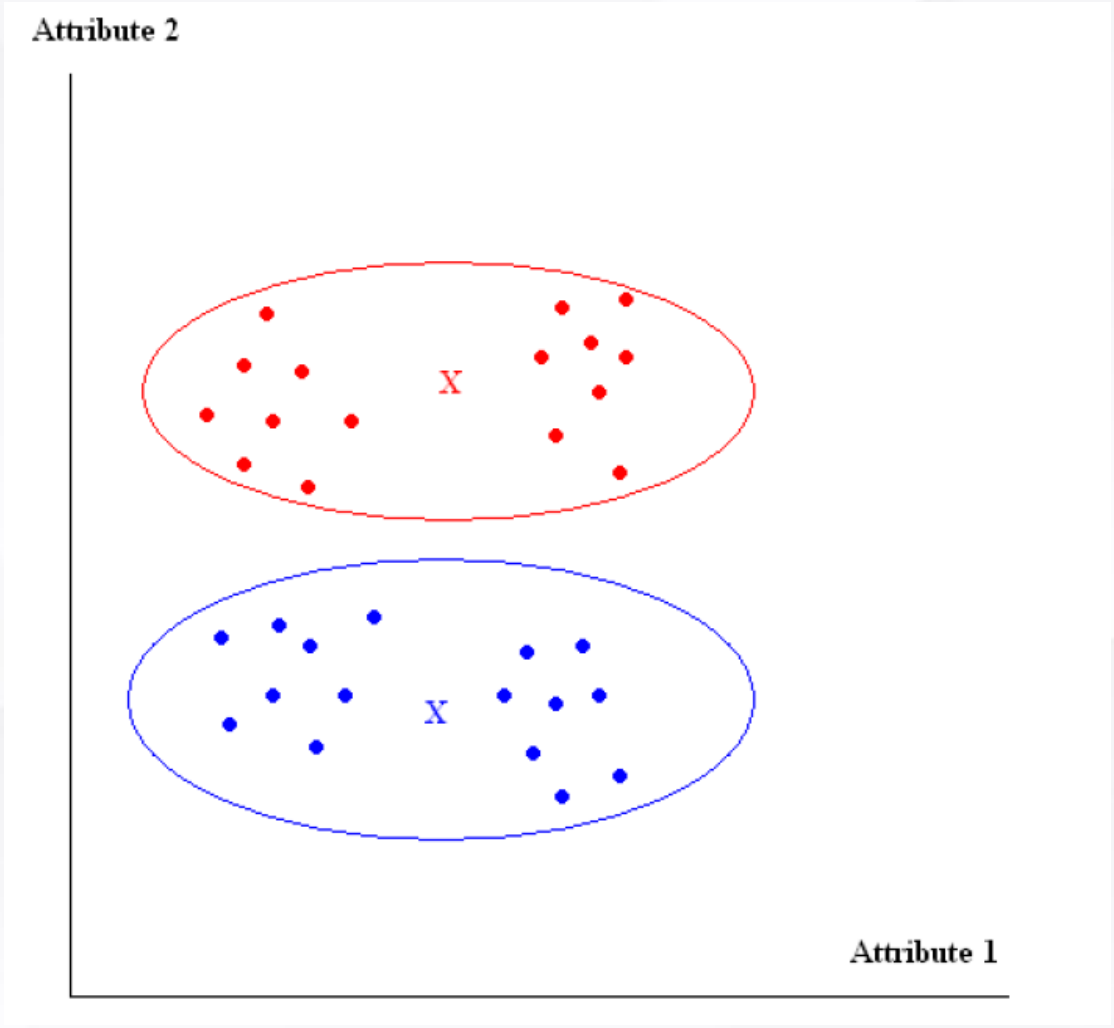


Machine Learning for Computer vision

- Unsupervised problem
  - Clusters generated by unsupervised learners might not adhere with real world clustering.
  - Real world problems are often subjective. Ex: segmentation.
  - Can a little bit of labeled data be used to guide an unsupervised learner?
  - Can the learner incorporate user suggestions and feedback?
  - Solution: Use Semi-Supervised Learning (SSL).

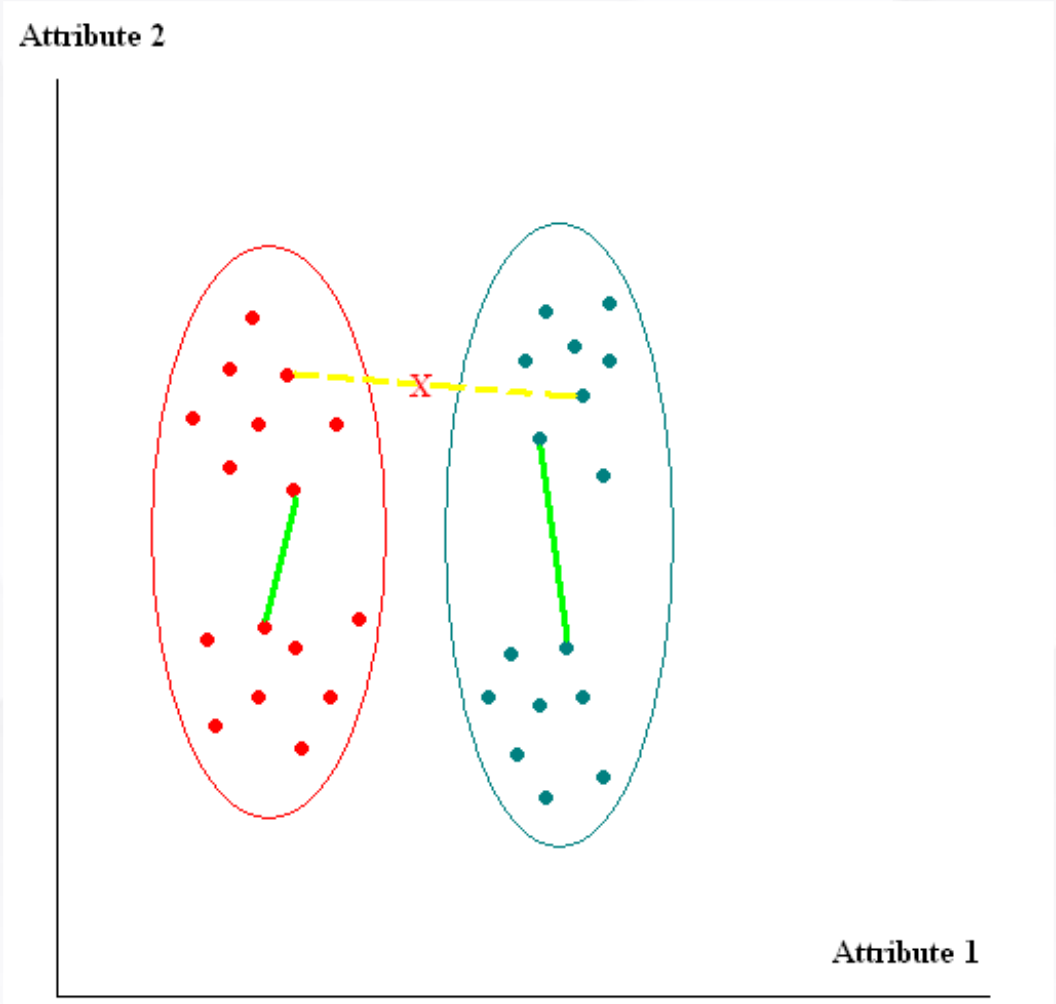
Machine Learning for Computer vision

- Semi-supervised learning
  - Unconstrained clustering



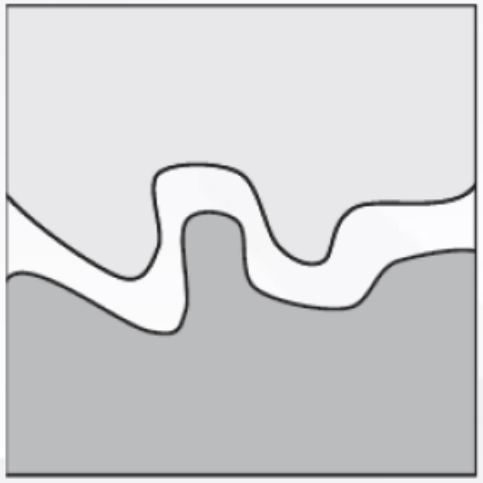
Machine Learning for Computer vision

- Semi-supervised learning
  - Constrained clustering

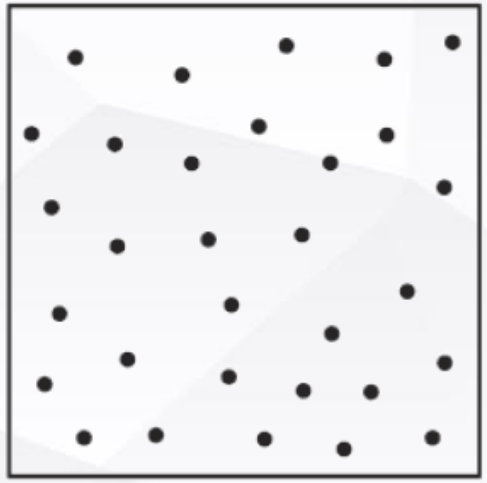


Machine Learning for Computer vision

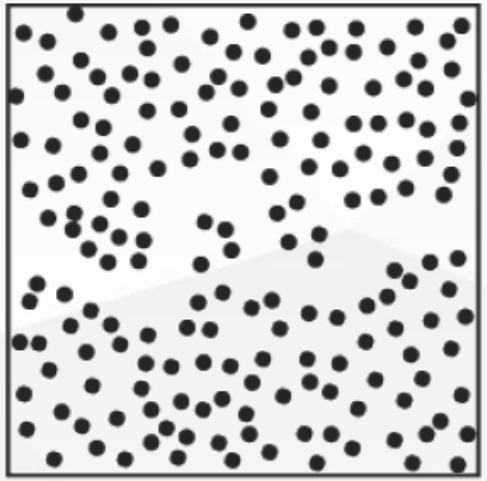
- Semi-supervised learning
  - With lots of unlabeled data the decision boundary becomes apparent.
  - Smoothness assumption:
    - The objective function is locally smooth over subsets of the feature space as depicted by some property of the marginal density.



Original decision boundary



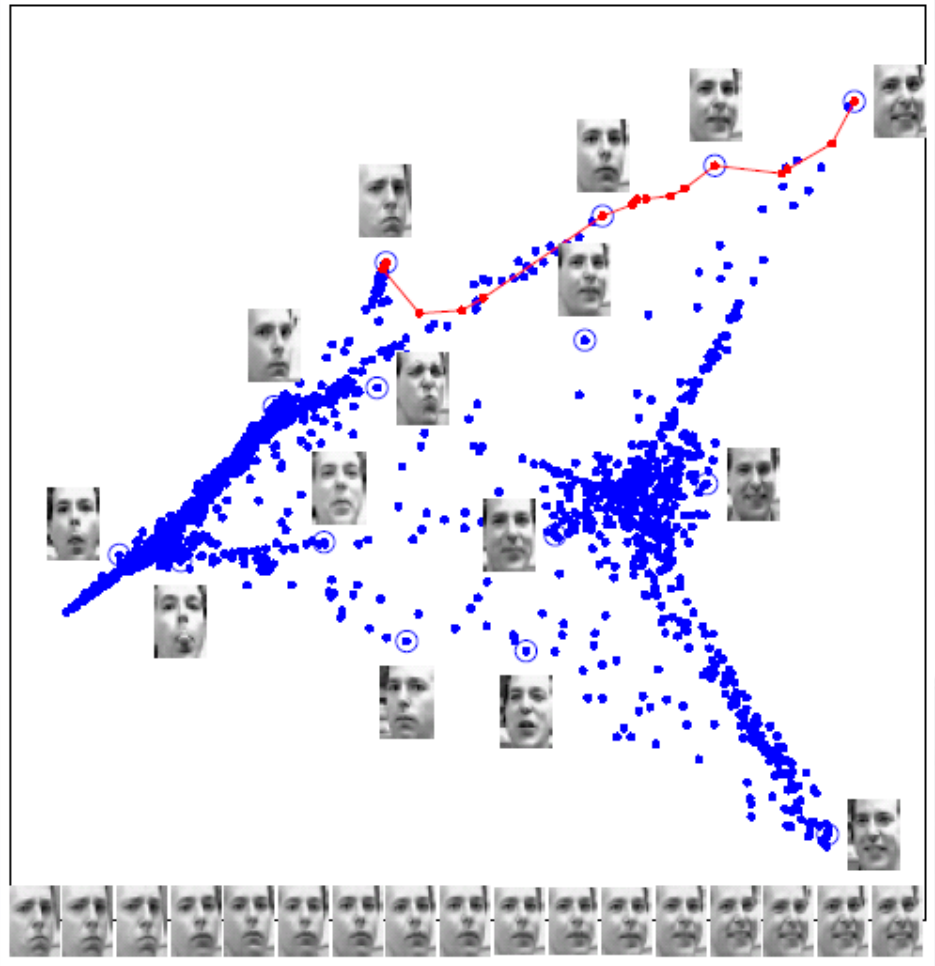
When only labeled data is Given.



With unlabeled data along with labeled data

Machine Learning for Computer vision

- Semi-supervised learning

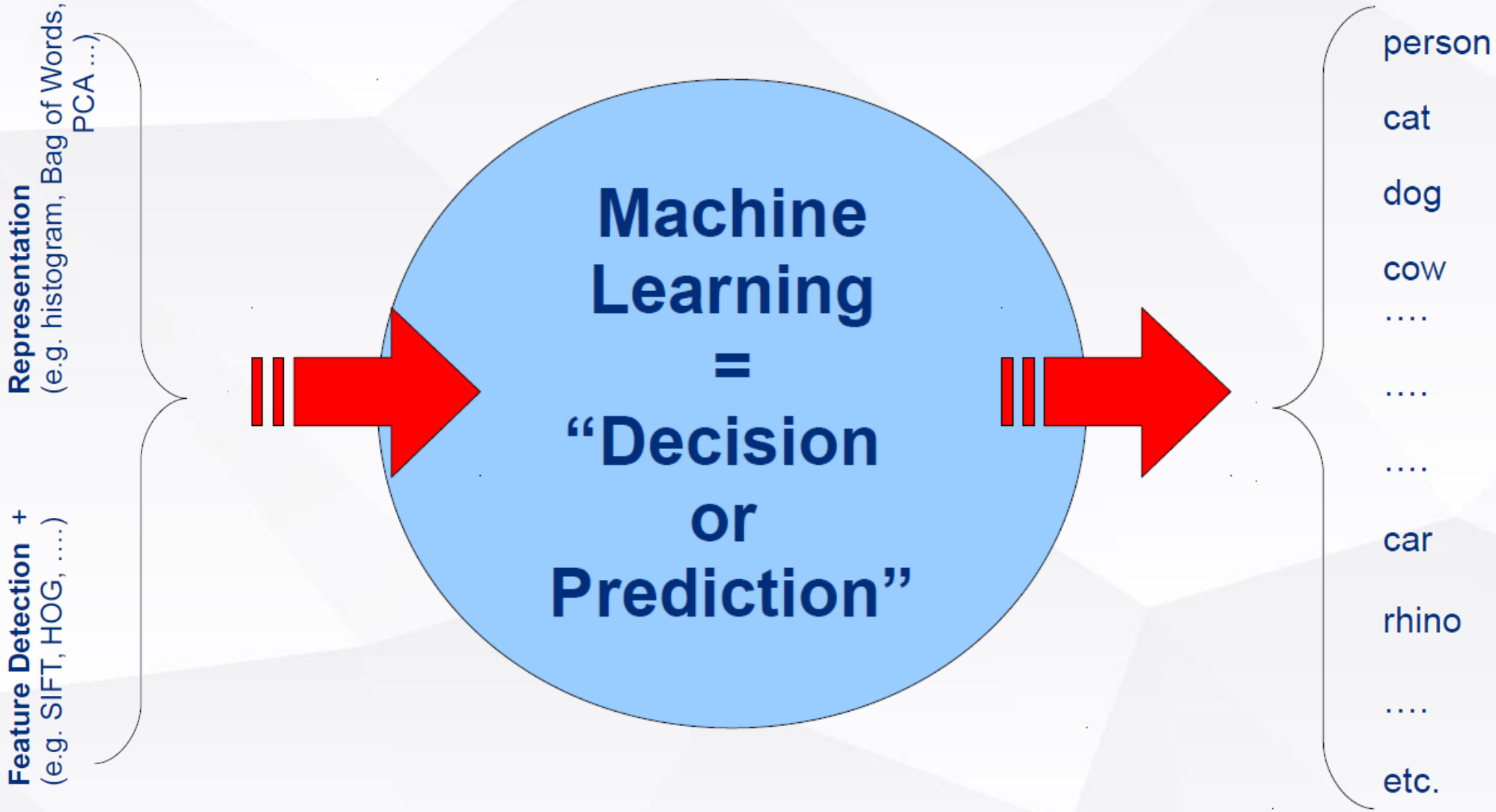


Machine Learning for Computer vision

- Curse of dimension
  - In many applications, we simply vectorizean image or image patch by a raster-scan.
  - 256 x 256 image converts to a 65,536-dimensional vector.
  - Images, therefore, are typically very high-dimensional data
  - Volume, and hence the number of points required to uniformly sample a space increases exponentially with dimension.
  - Affects the convergence of any learning algorithm.
  - In some applications, we know that there are only a few variables, for e.g., face pose and illumination.
  - Data lie on some low-dimensional subspace/manifold in the high-dimensional space.

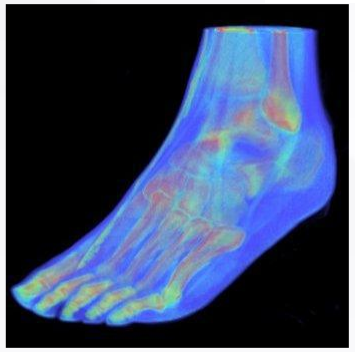
Machine Learning for Computer vision

- … in the big picture

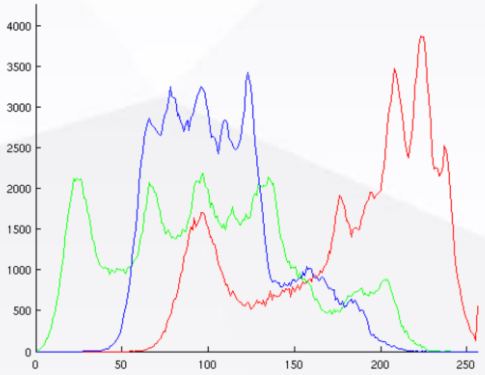


Machine Learning for Computer vision

- Example Input(s) to ML in Computer Vision
  - **“Direct Sample” inputs**
    - Pixels / Voxels / 3D Points
    - sub-samples (key-points, feature-points)



  - **Feature Vectors**
    - shape measures
    - edge distributions
    - colour distributions
    - texture measures / distributions
    - E.g. […. SIFT, SURF, HOG …. etc.]




```
V = { 0.103900,
      120.102,
      30.10101, .... , ...}
```

Machine Learning for Computer vision

- Common ML Tasks in Computer Vision
  - Object **Classification**

    what object ?



  - Object **Detection**

    object or no-object ?

http://pascallin.ecs.soton.ac.uk/challenges/VOC/

{people | vehicle | … intruder}

  - Instance **Recognition** ?

    who (or what) is it ?

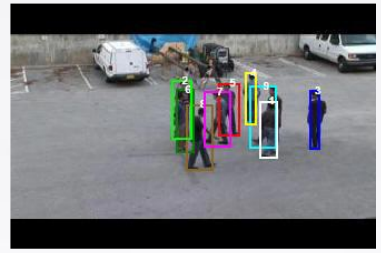

{face | vehicle plate| gait …. → biometrics}

  - **Sub-category** analysis

    which object type ?

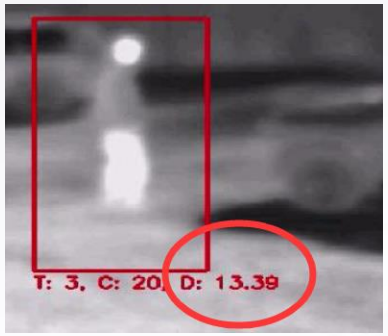
  - Sequence { **Recognition | Classification** } ?

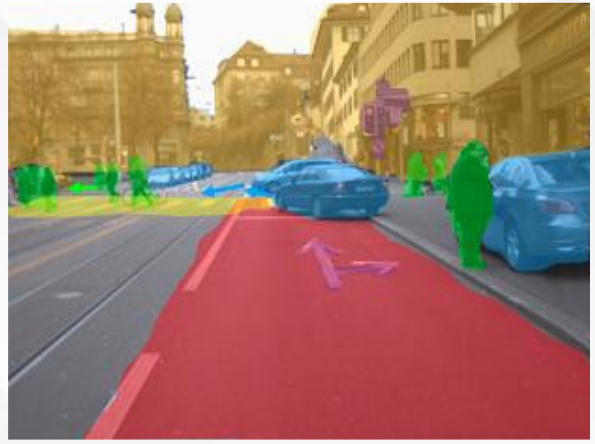
    what is happening / occurring ?

Machine Learning for Computer vision

- Types of ML Problem
- Classification
  - Predict (classify) sample → discrete set of class labels
  - e.g. classes {object 1, object 2 … } for recognition task
  - e.g. classes {object, !object} for detection task
- Regression (traditionally less common in comp. vis.)
  - Predict sample → associated numerical value (variable)
  - e.g. distance to target based on shape features
  - Linear and non-linear attribute to value relationships
- Association or clustering
  - grouping a set of instances by attribute similarity
  - e.g. image segmentation





Machine Learning for Computer vision

- Regression Example – Head Pose Estimation

http://www.youtube.com/embed/UcF_otQSMEc?rel=0

Input: image features (HOG)
Output: { yaw | pitch }
varying illumination + vibration

Machine Learning for Computer vision

- Learning in **general**, from **specific** examples
  - E is **specific** : a specific **example** (e.g. female face)
  - T is **general** : a general **task** (e.g. gender recognition)
  - Program P "**learns**" task T from set of **examples** {E}

- Thus if P improves learning must be of the following form:

"to **learn a general** [ability | behavior | function | rules] **from specific examples**"

Machine Learning for Computer vision

- A simple learning example ....
  - Learn prediction of “Safe conditions to fly ?”
    - based on the weather conditions = attributes
    - classification problem, class = {yes, no}

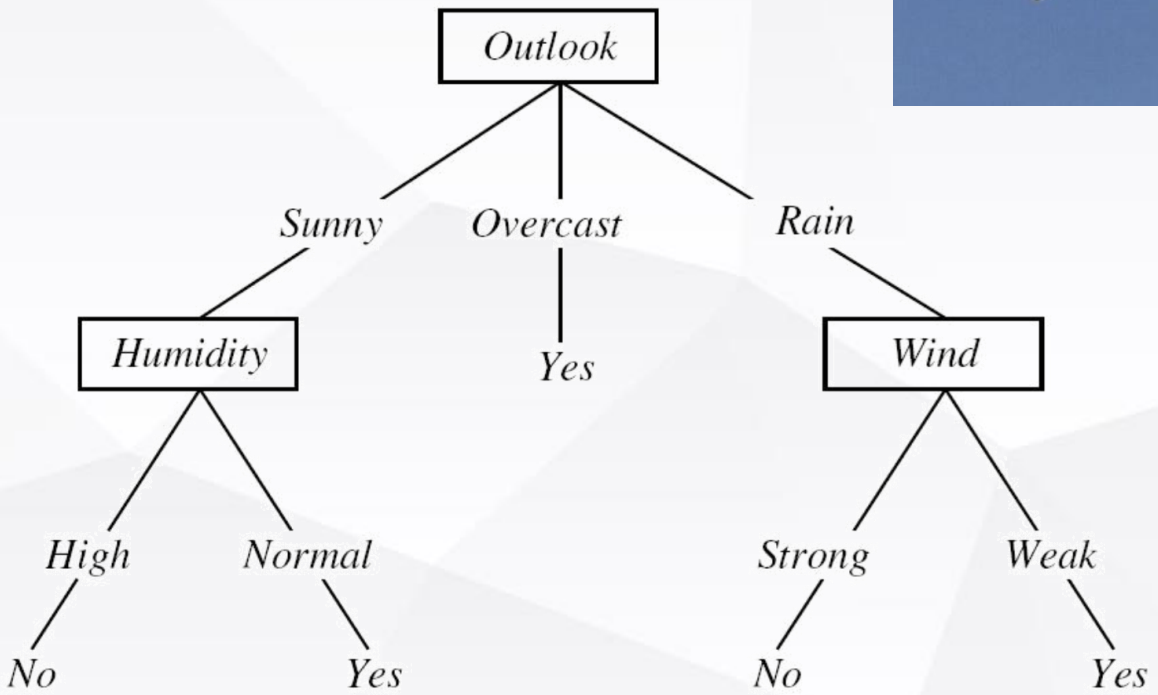
Attributes

Classification

Outlook	Temperature	Humidity	Windy	Fly
Sunny	85	85	False	No
Sunny	80	90	True	No
Overcast	83	86	False	Yes
Rainy	75	80	False	Yes
…	…	…	…	…

Machine Learning for Computer vision

- Decision Trees
  - Attribute based, decision logic construction
  - boolean outcome
  - discrete set of outputs



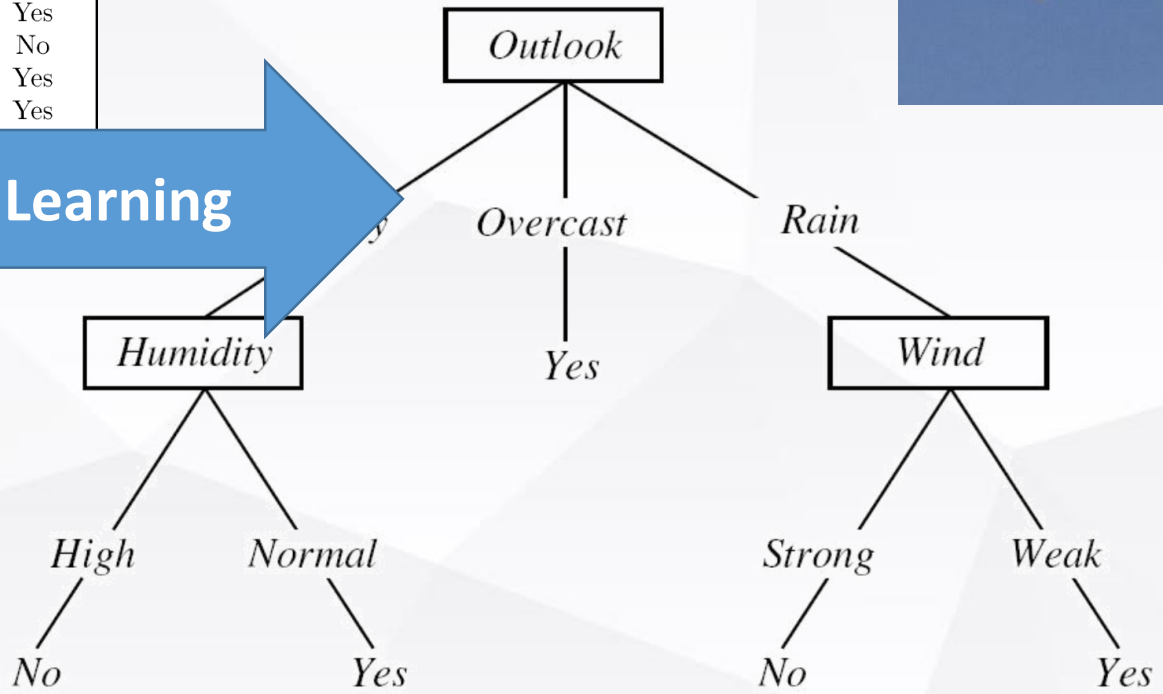
Safe conditions to fly ?

Machine Learning for Computer vision

- Decision Trees
  - Set of Specific Examples …

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	St	
D12	Overcast	Mild	High	St	
D13	Overcast	Hot	Normal	W	
D14	Rain	Mild	High	St	

Learning

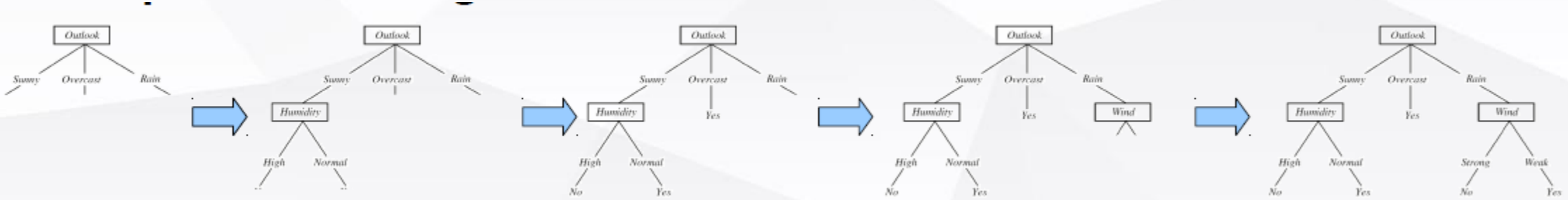


Safe conditions to fly ?

Generalized Rule

Machine Learning for Computer vision

- Growing Decision Trees
  - Construction is carried out top down based on node splits that maximize the reduction in the entropy in each resulting sub-branch of the tree
  - **Key Algorithmic Steps** [Quinlan, '86]
    - 1. **Calculate** the information gain of splitting on each attribute (i.e. reduction in entropy (variance))
    - 2. **Select** attribute with maximum information gain to be a new node
    - 3. **Split** the **training data** based on this attribute
    - 4. Repeat recursively (step 1 → 3) for each sub-node until all



Machine Learning for Computer vision

- Is it really that simple ?

https://vimeo.com/51210467

- Is it really that simple ?

Must (Must, Must!)
avoid over-fitting …..
(i.e. over-learning)

Machine Learning for Computer vision

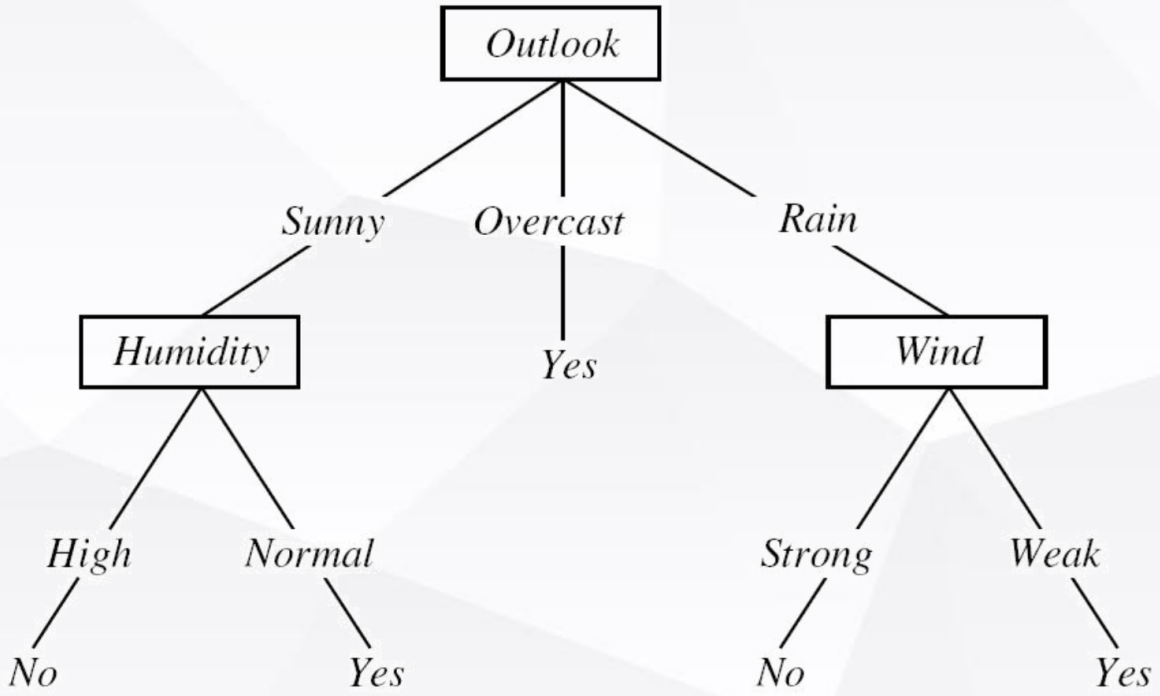
- Follow the principle of Occam's Razor
- Occam's Razor（奥卡姆剃刀原理）
- “entia non sunt multiplicanda praeter necessitatem” (Latin!)
- “All things being equal, the simplest solution tends to be the best one”(English)
- 切勿浪费较多东西去做用较少的东西同样可以做好的事情。(中文）
- Machine Learning : prefer the **simplest** {model | hypothesis | …. tree | network } that fits the data

14th-century English logician William of Ockham

Machine Learning for Computer vision

- Problem of **Overfitting**
  - Consider adding noisy training example:

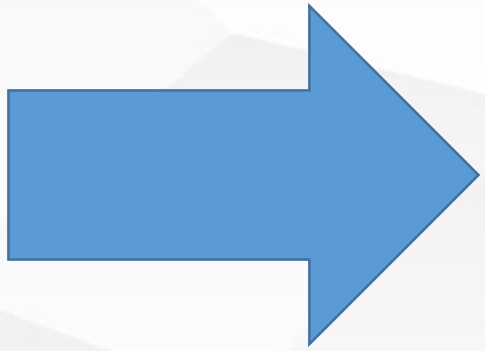
    [Sunny, Hot, Normal, Strong, Fly=Yes](Wrong Label)
  - What training effect would it have on earlier tree?



Machine Learning for Computer vision

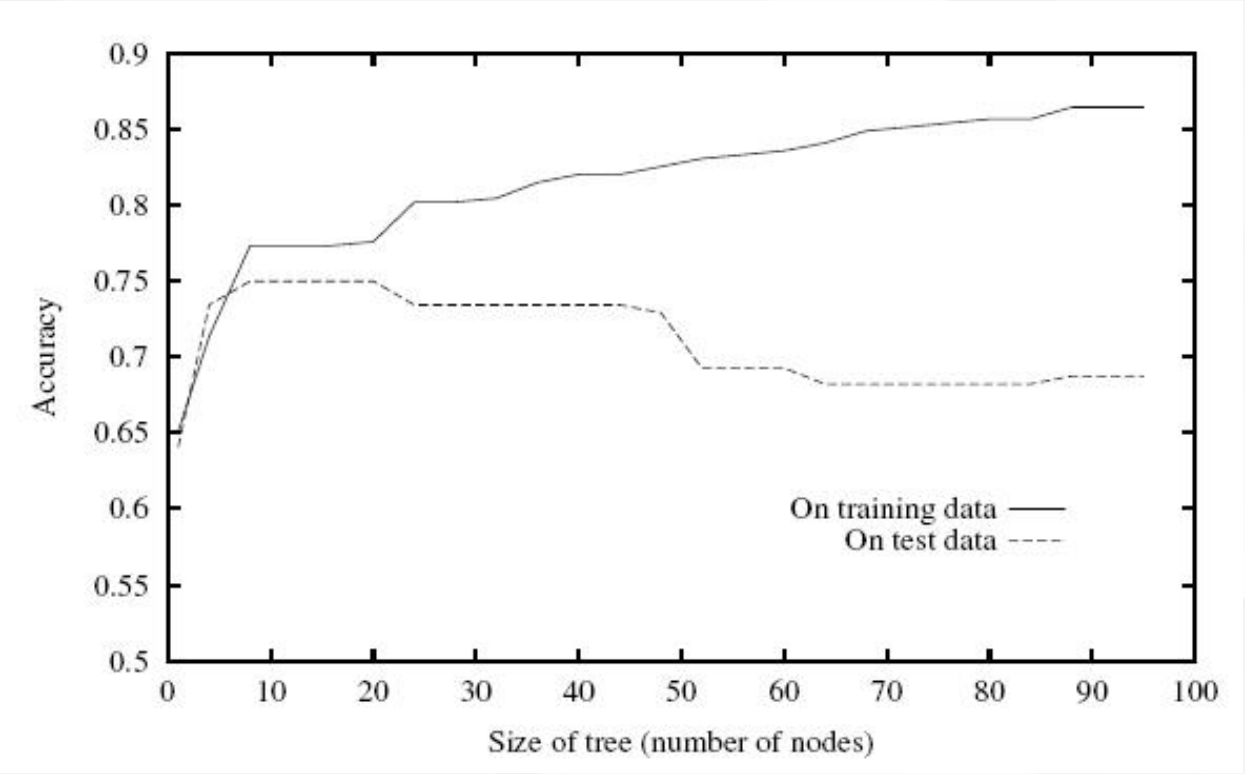
- Problem of **Overfitting**
  - Consider adding noisy training example:

    [Sunny, Hot, Normal, Strong, Fly=Yes](Wrong Label)
  - What training effect would it have on earlier tree?
  - Error in example = **error in tree construction!**



Machine Learning for Computer vision

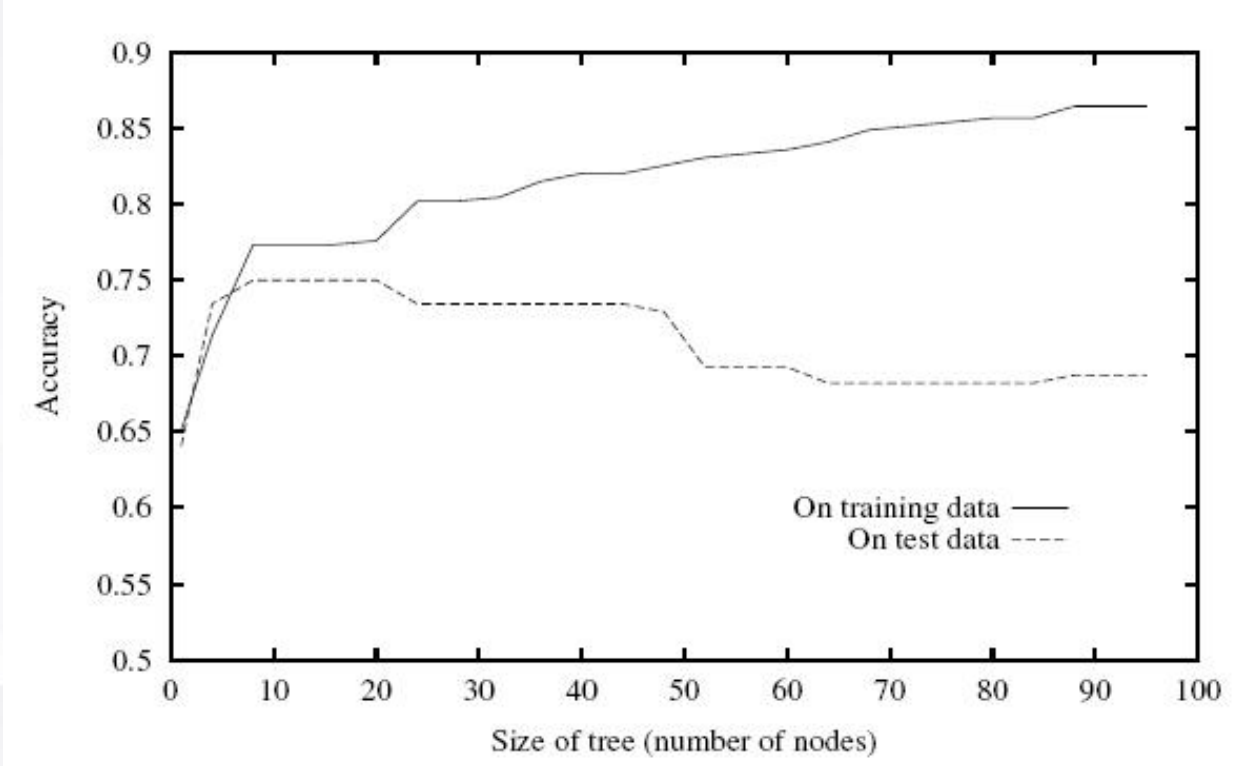
- **Overfitting** in general
  - Performance on the training data (with noise) **improves**
  - Performance on the unseen test data **decreases**



- For decision trees: tree complexity increases, learns training data too well! (over-fits)

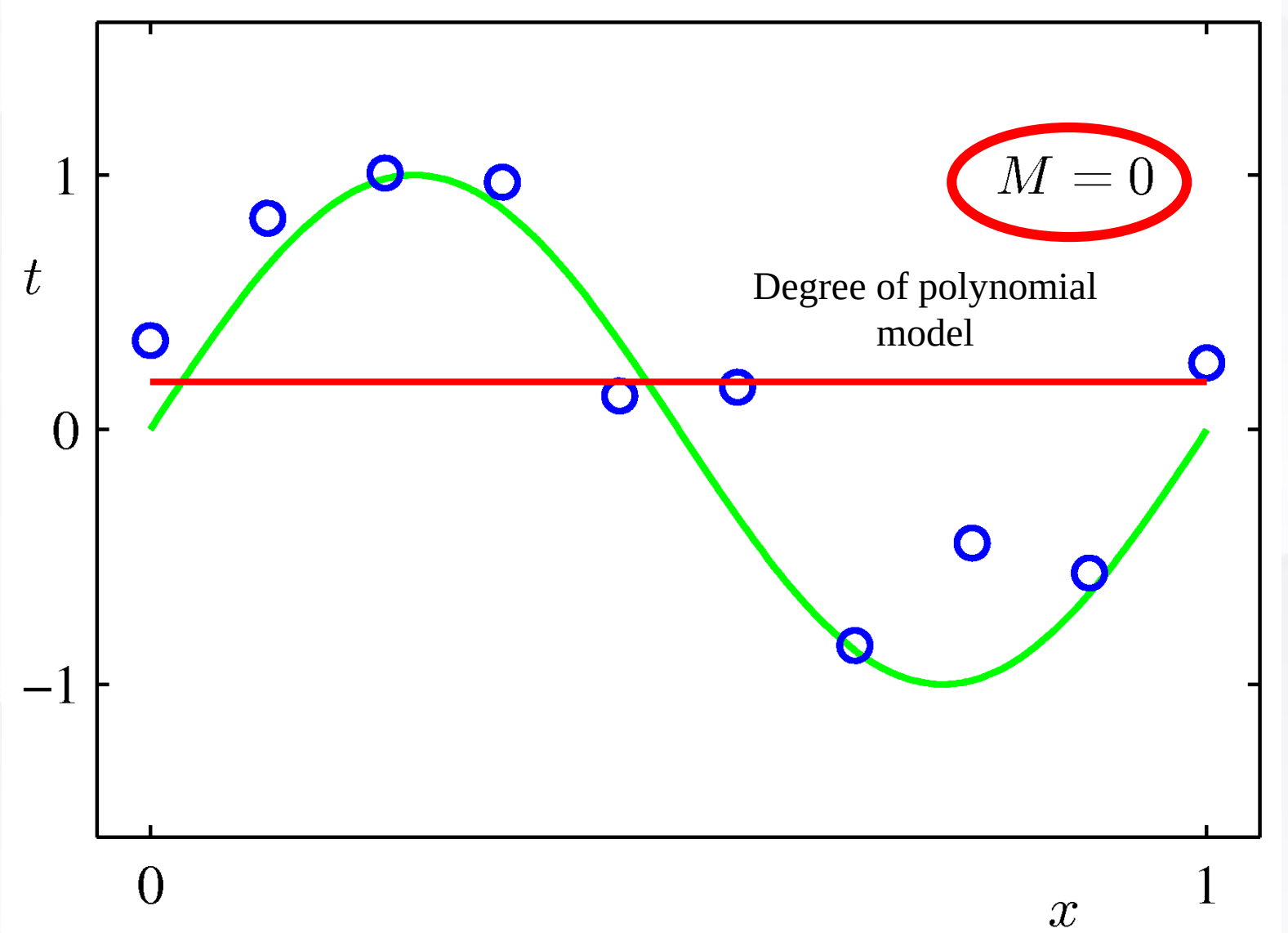
Machine Learning for Computer vision

- **Overfitting** in general
  - Hypothesis is too specific towards training examples
  - Hypothesis not general enough for test data



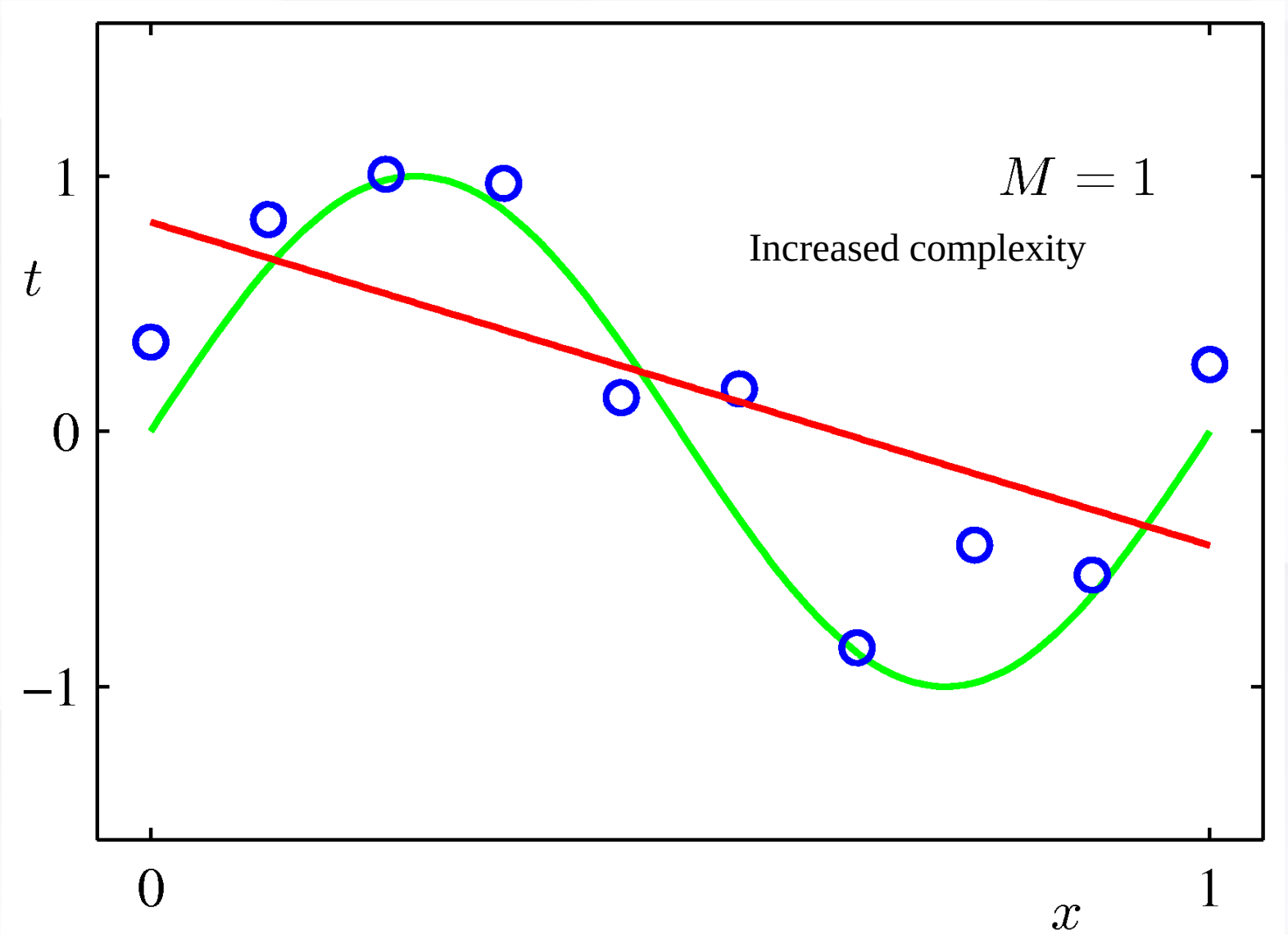
Machine Learning for Computer vision

- A graphical example: function approximation (via regression)



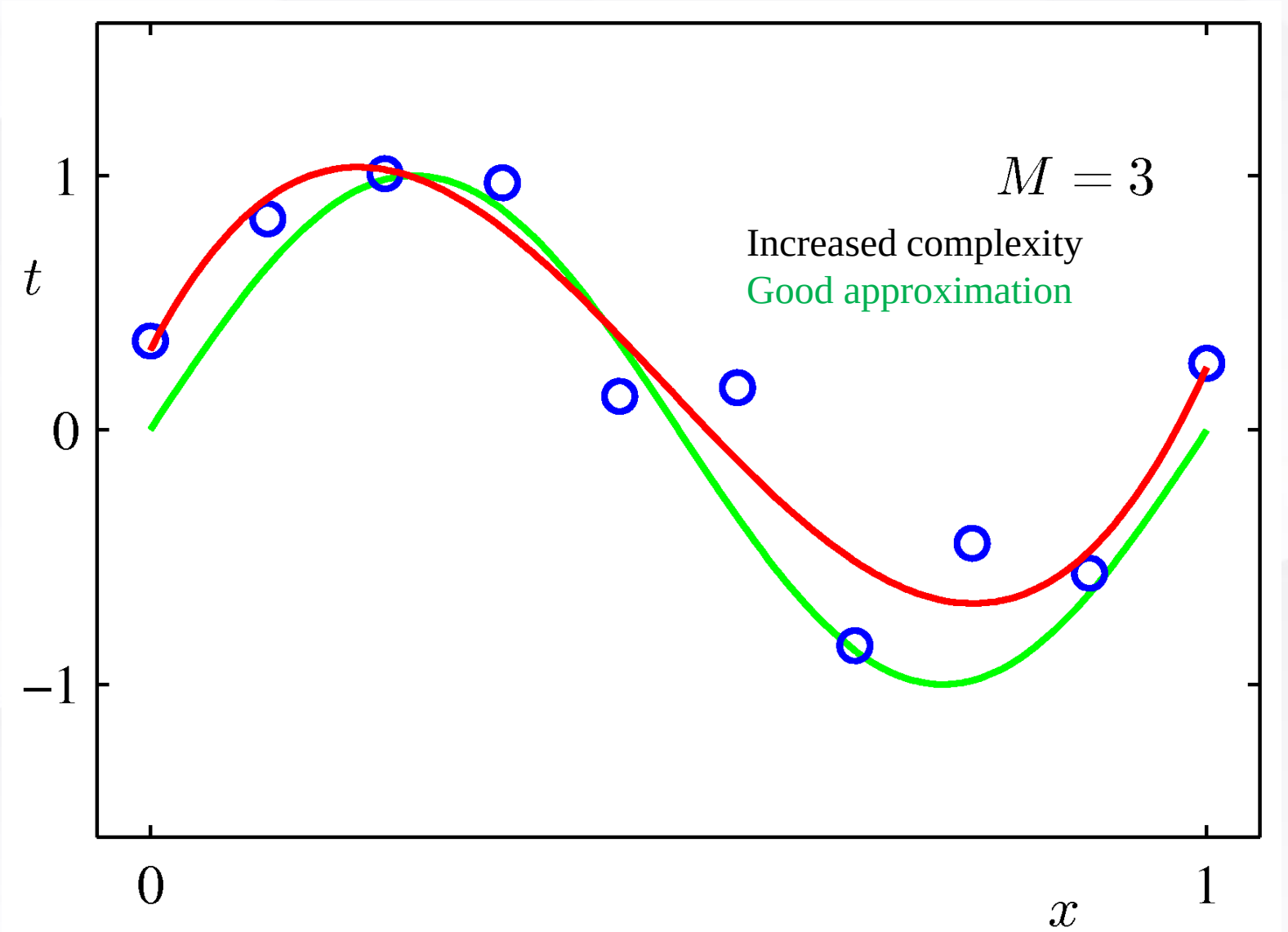
Machine Learning for Computer vision

- A graphical example: function approximation (via regression)



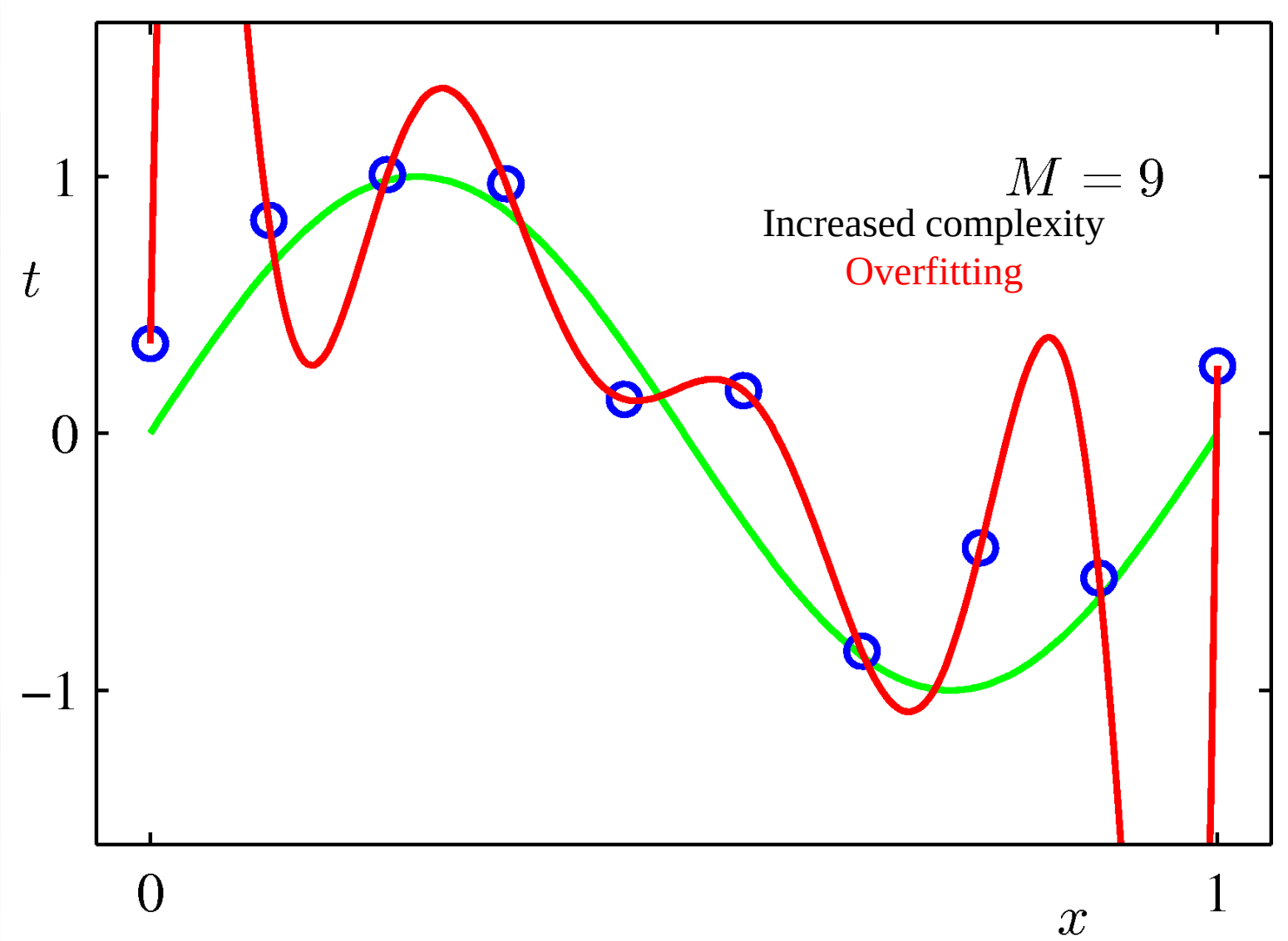
Machine Learning for Computer vision

- A graphical example: function approximation (via regression)



Machine Learning for Computer vision

- A graphical example: function approximation (via regression)



Machine Learning for Computer vision

- A stitch in time ...

Decision Trees
[Quinlan, '86]
and many others..

Ensemble Classifiers 2010…

Machine Learning for Computer vision

- Extending to Multi-Tree Ensemble Classifiers
  - **Key Concept**: combining multiple classifiers
    - strong classifier: output strongly correlated to correct classification
    - weak classifier: output weakly correlated to correct classification
    - i.e. it makes a lot of miss-classifications (e.g. tree with limited depth)
  - How to combine:
    - **Bagging**:

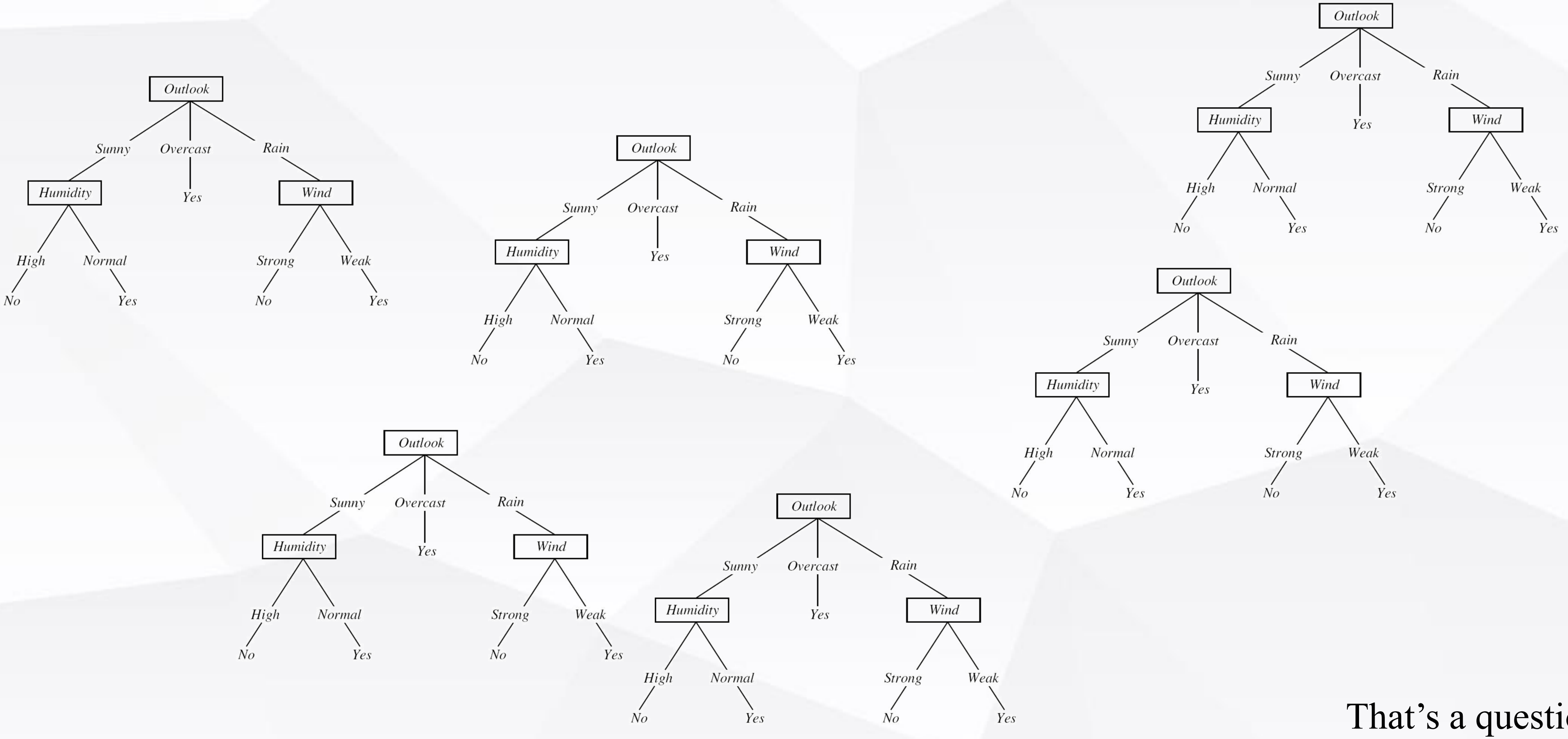
      train N classifiers on random sub-sets of training set; classify using majority vote of all N (and for regression use average of N predictions)

    - **Boosting**:

      As per bagging, but introduce weights for each classifier based on performance over the training set
  - Two **examples**: Boosted Trees + (Random) Decision Forests
    - P. S. Can be used with any classifiers (not just decision trees!)

Machine Learning for Computer vision

- Extending to Multi-Tree Ensemble Classifiers
    - To bag or to boost ….



That’s a question…

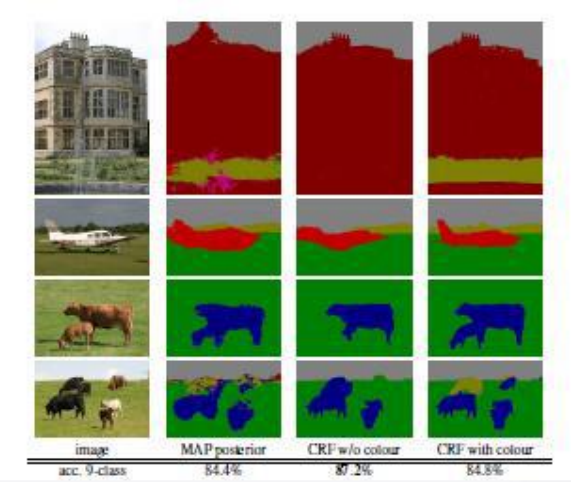
Machine Learning for computer vision

- Extending to Multi-Tree Classifiers
  - Bagging = all equal (simplest approach)
  - Boosting = classifiers weighted by performance
    - poor performers removed (zero or very low) weight
    - $t+1_{th}$ classifier concentrates on the examples t_{th} classifier got **wrong**
  - To bag or boost ? - boosting generally works very well

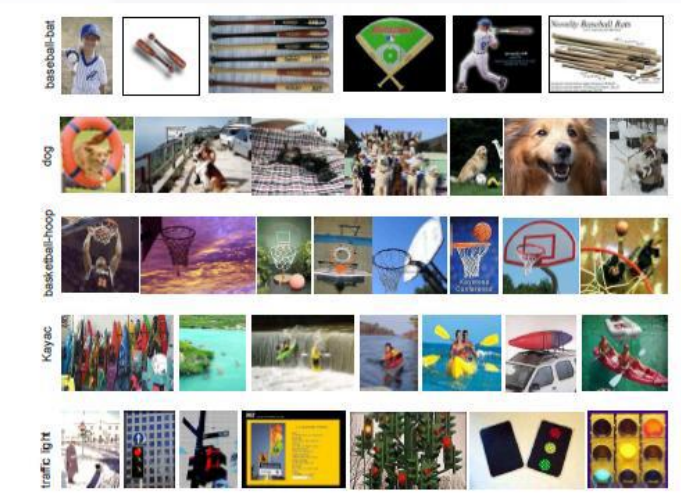
(but what about over-fitting ?)

Machine Learning for computer vision

- Decision Forests (a.k.a. Random Forests/Trees)
  - Bagging using multiple decision trees where each tree in the ensemble classifier ...
    - is trained on a random subsets of the training data
    - computes a node split on a random subset of the attributes
    - close to “state of the art” for object segmentation / classification



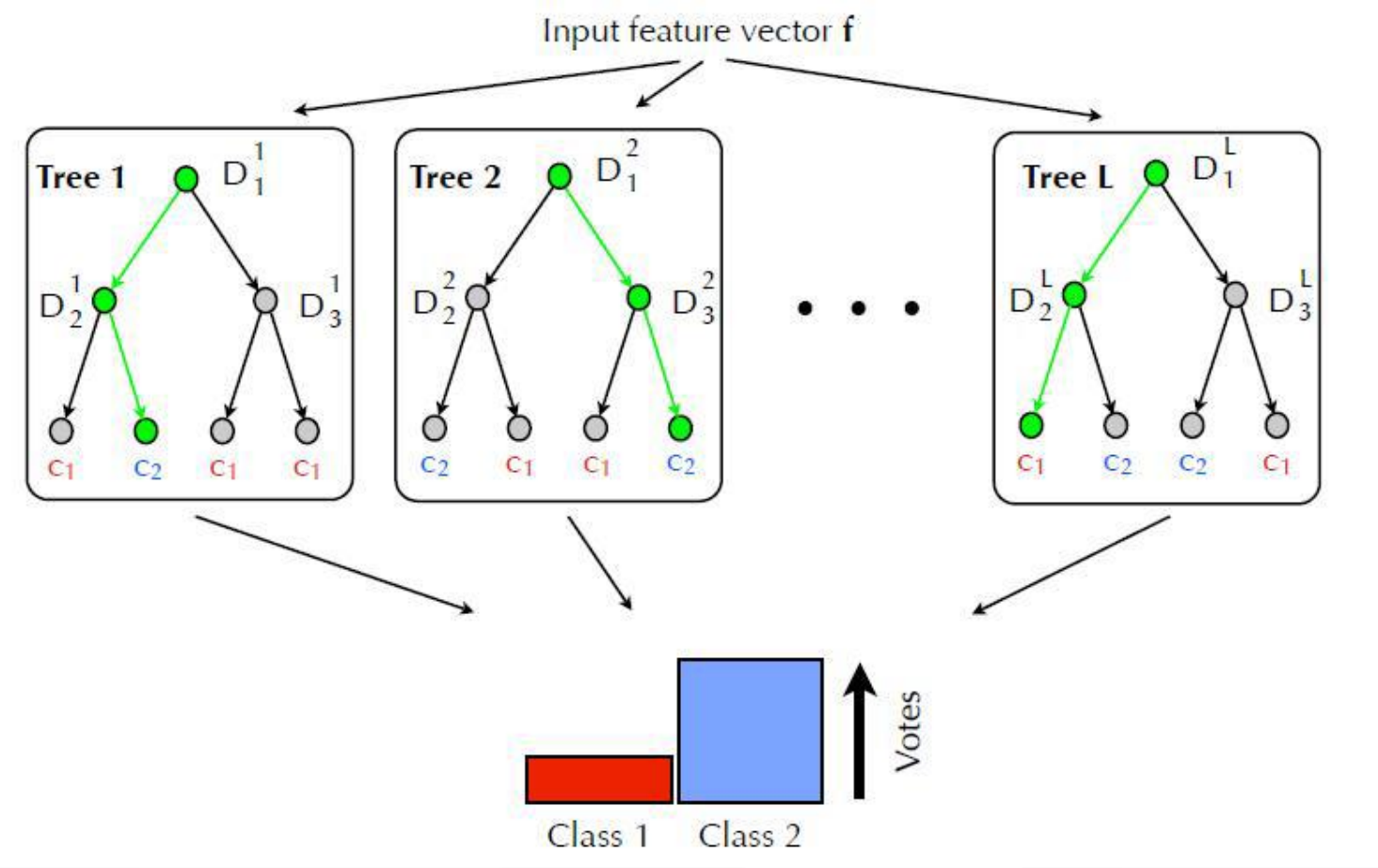
[schroff 2008]



[Bosch 2007]

Machine Learning for computer vision

- Decision Forests (a.k.a. Random Forests/Trees)



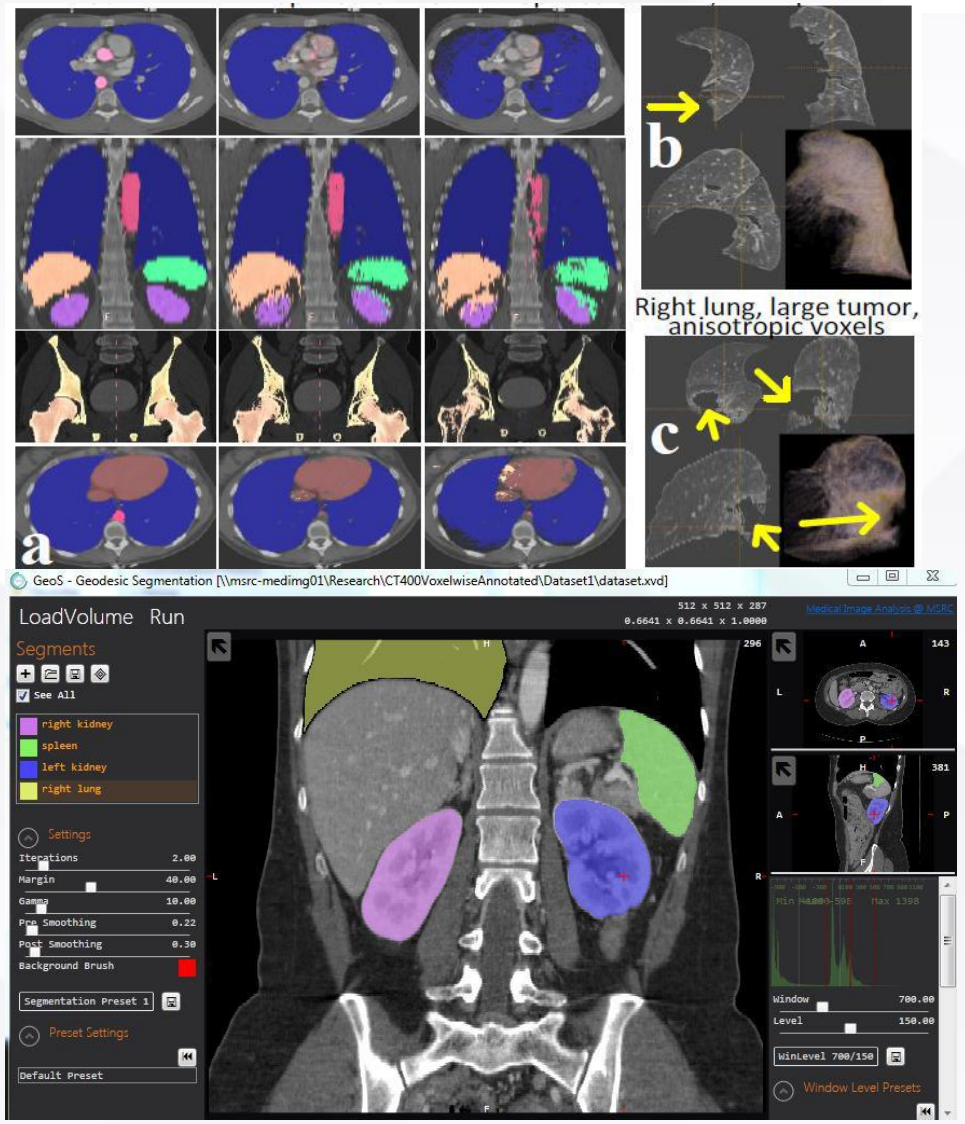
Machine Learning for computer vision

- Decision Forests (a.k.a. Random Forests/Trees)
  - Decision Forest = Multi Decision Tree Ensemble Classifier
    - bagging approach used to return classification
    - [alternatively weighted by number of training items assigned to the final leaf node reached in tree that have the same class as the sample (classification) or statistical value (regression)]
  - **Benefits**: efficient on large data sets with multi attributes and/or missing data, inherent variable importance calc., unbiased test error ("out of bag"), "does not overfit"
  - **Drawbacks**: evaluation can be slow, lots of data for good performance, complexity of storage ...

["Random Forests", Breiman 2001]

Machine Learning for computer vision

- Decision Forests (a.k.a. Random Forests/Trees)

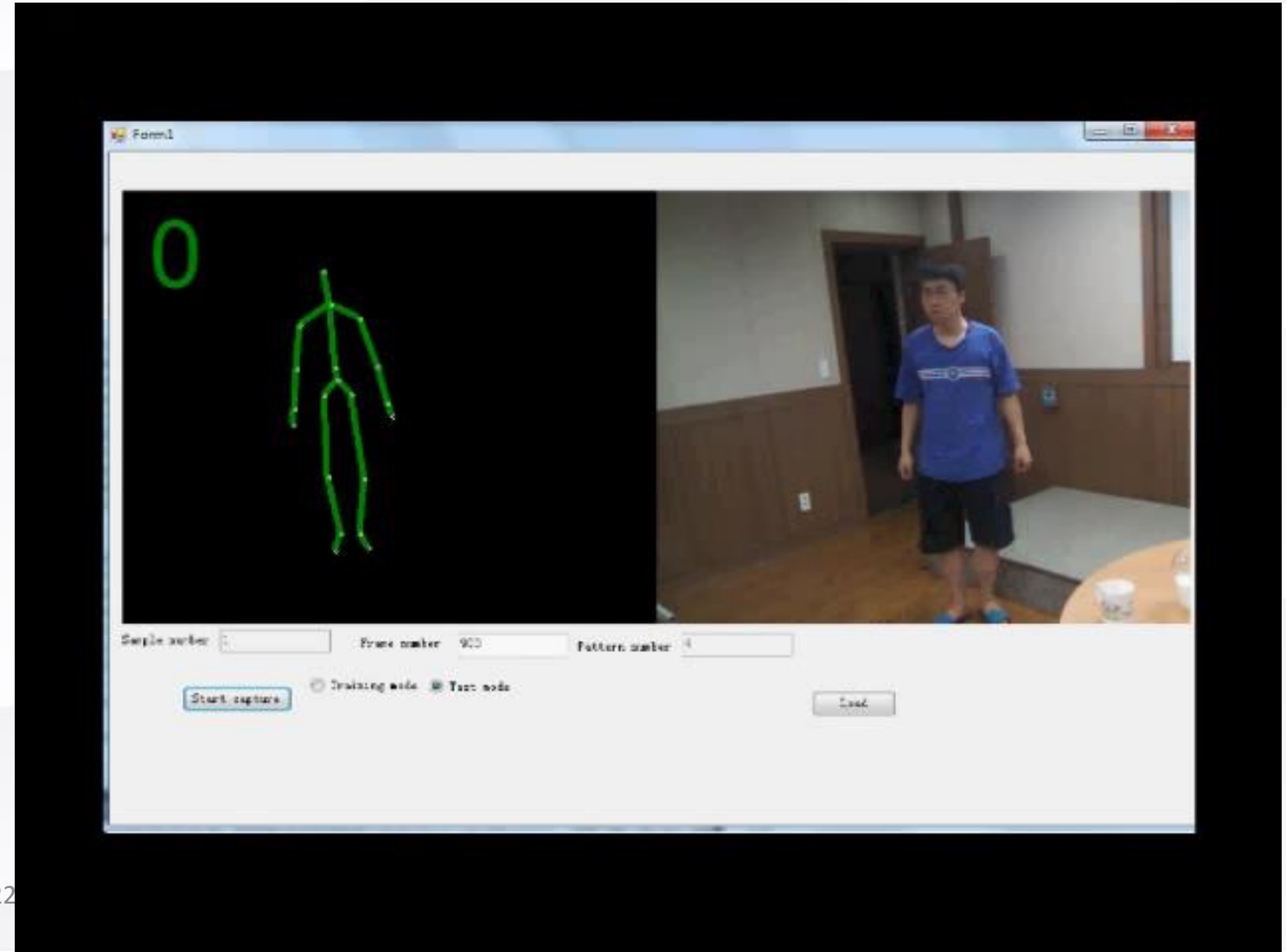


Gall J. and Lempitsky V., Class-Specific Hough Forests for Object **Detection**, IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09), 2009.

Montillo et al.. "Entangled decision forests and their application for semantic segmentation of CT images." In Information Processing in Medical Imaging, pp. 184-196. 2011. http://research.microsoft.com/en-us/projects/decisionforests/

Machine Learning for computer vision

- Microsoft Kinect



Machine Learning for computer vision

- Decision Forests (a.k.a. Random Forests/Trees)
  - Body Pose Estimation in Realtime From Depth Images
  - using Decision Forest Approach

Shotton et al., Real-Time Human Pose Recognition in Parts from a Single Depth Image, CVPR, 2011 http://research.microsoft.com/apps/pubs/default.aspx?id=145347

What if **every weak classifier** was just the presence/absence of **an image feature** ?
(i.e. feature present = {yes, no})

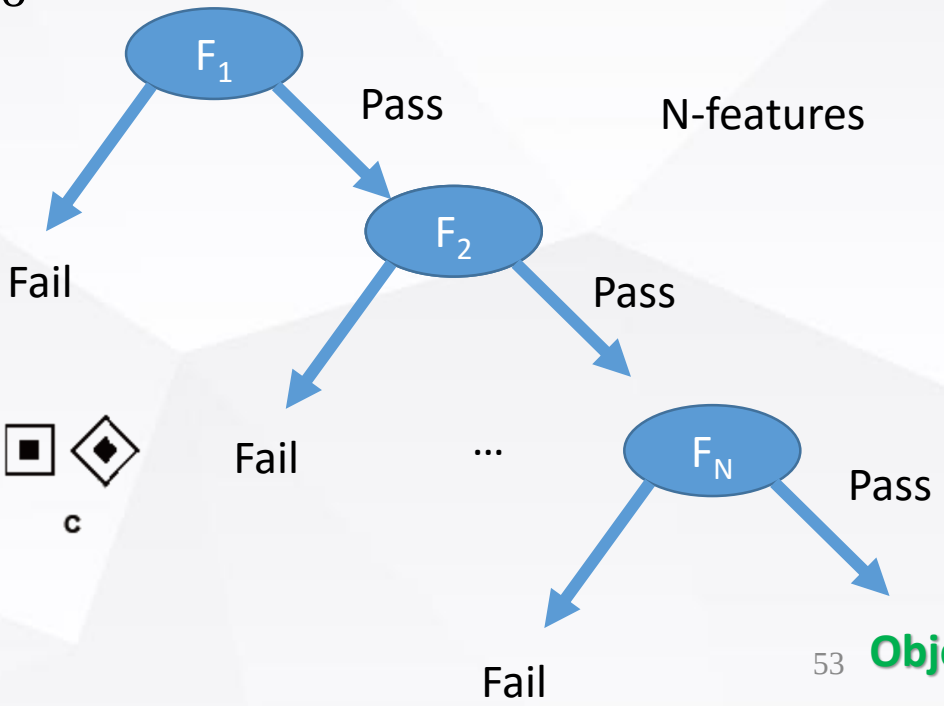
As the **number of features present from a given object**, in a given scene location, **goes up** the **probability of the object not being present goes down!**

This is the concept of **feature cascades**.

Machine Learning for computer vision

- Feature Cascading .....
  - Use boosting to **order image features from most to least discriminative for a given object ....**
    - allow high false positive per feature (i.e. it's a weak classifier!)
  - As feature F_1 to F_N of an object is present → probability of nonoccurrence within the image tends to zero
  - e.g. Extended **Haar features**
    - set of differences between image regions
    - rapid evaluation (and non-occurrence) rejection

[Volia / Jones 2004]



Machine Learning for computer vision

- Feature Cascading .....

[Volia / Jones 2004]

Machine Learning for computer vision

- Feature Cascading .....

[Volia / Jones 2004]

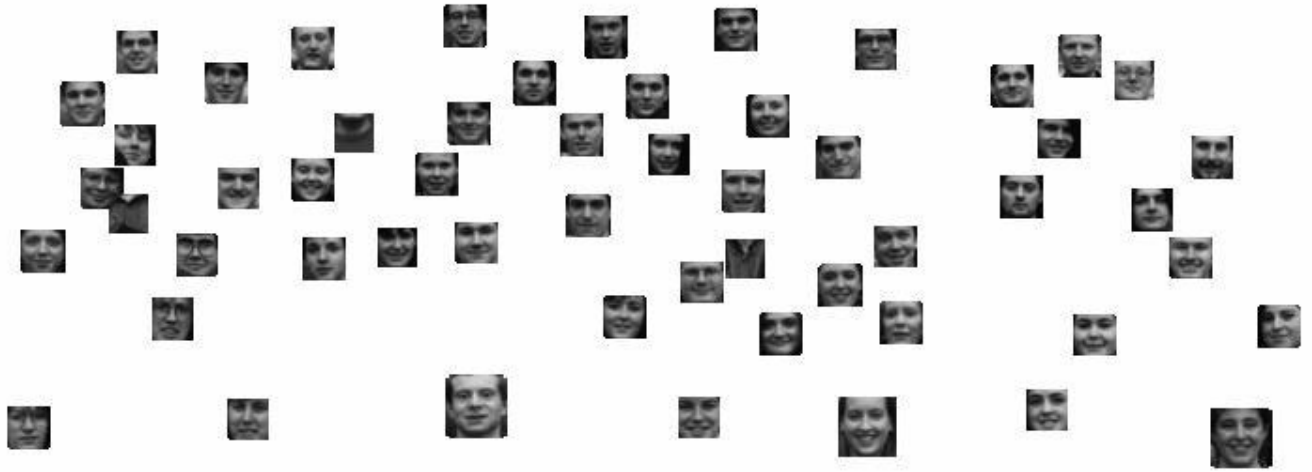
Machine Learning for computer vision

- Feature Cascading .....

[Volia / Jones 2004]

Machine Learning for computer vision

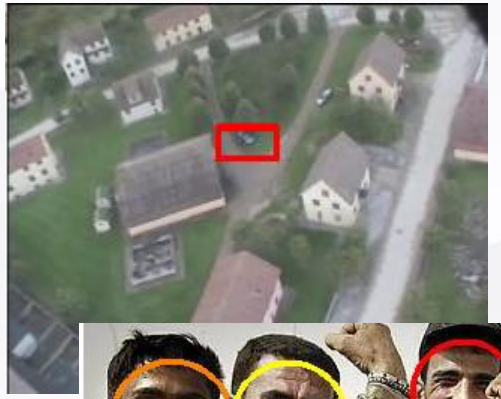
- Feature Cascading .....

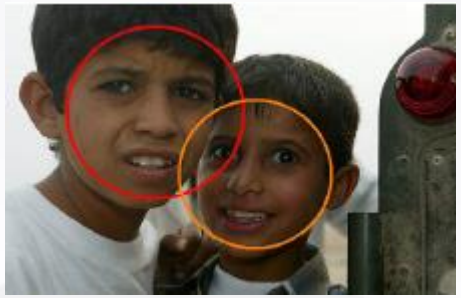


[Volia / Jones 2004]

Machine Learning for computer vision

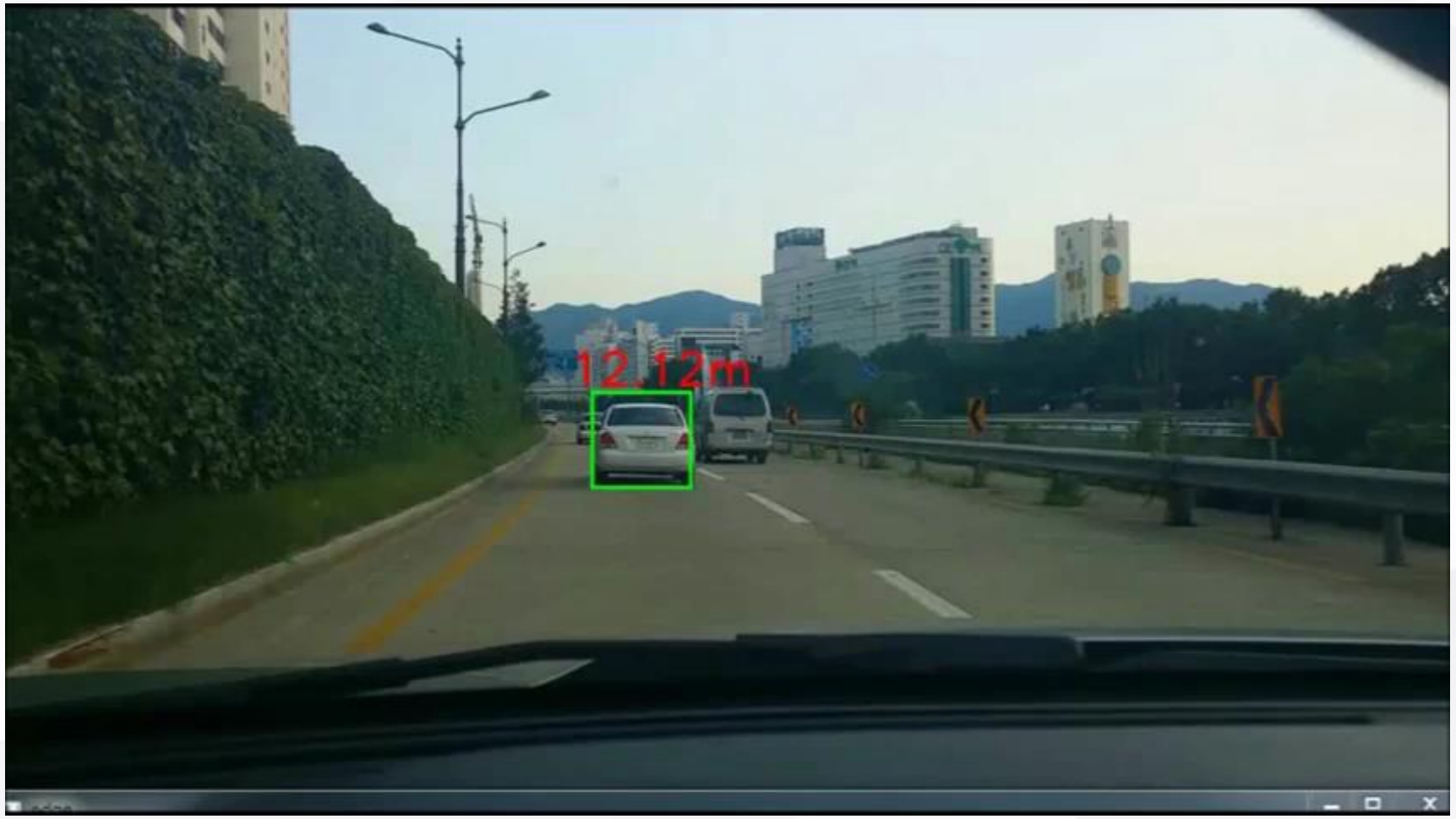
- Haar Feature Cascades
  - Real-time Generalized Object Recognition
  - Benefits
    - Multi-scale evaluation
    - scale invariant
    - Fast, real-time detection
    - “Direct” on image
    - no feature extraction
    - Haar features: contrast/ colour invariant
  - Limitations
    - poor performance on non-rigid objects
    - object rotation





Machine Learning for computer vision

- Haar Feature Cascades

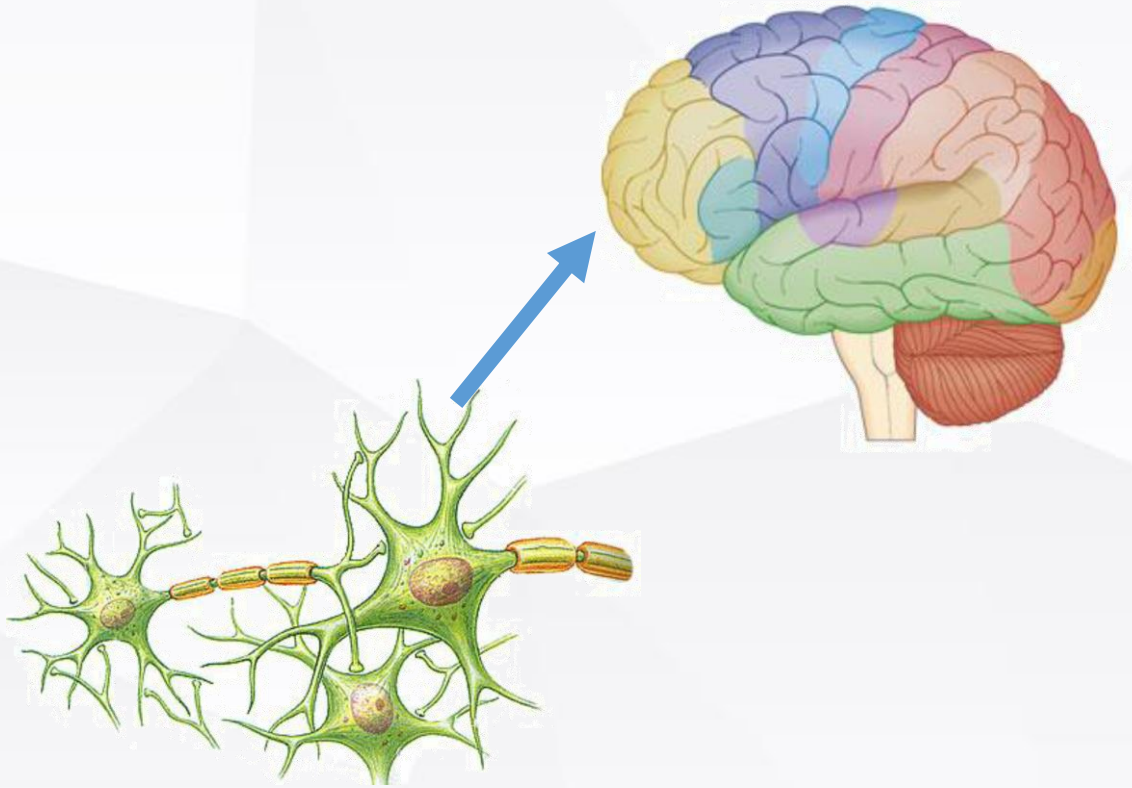


https://www.youtube.com/watch?v=91cp-J-RMf4

Machine Learning for computer vision

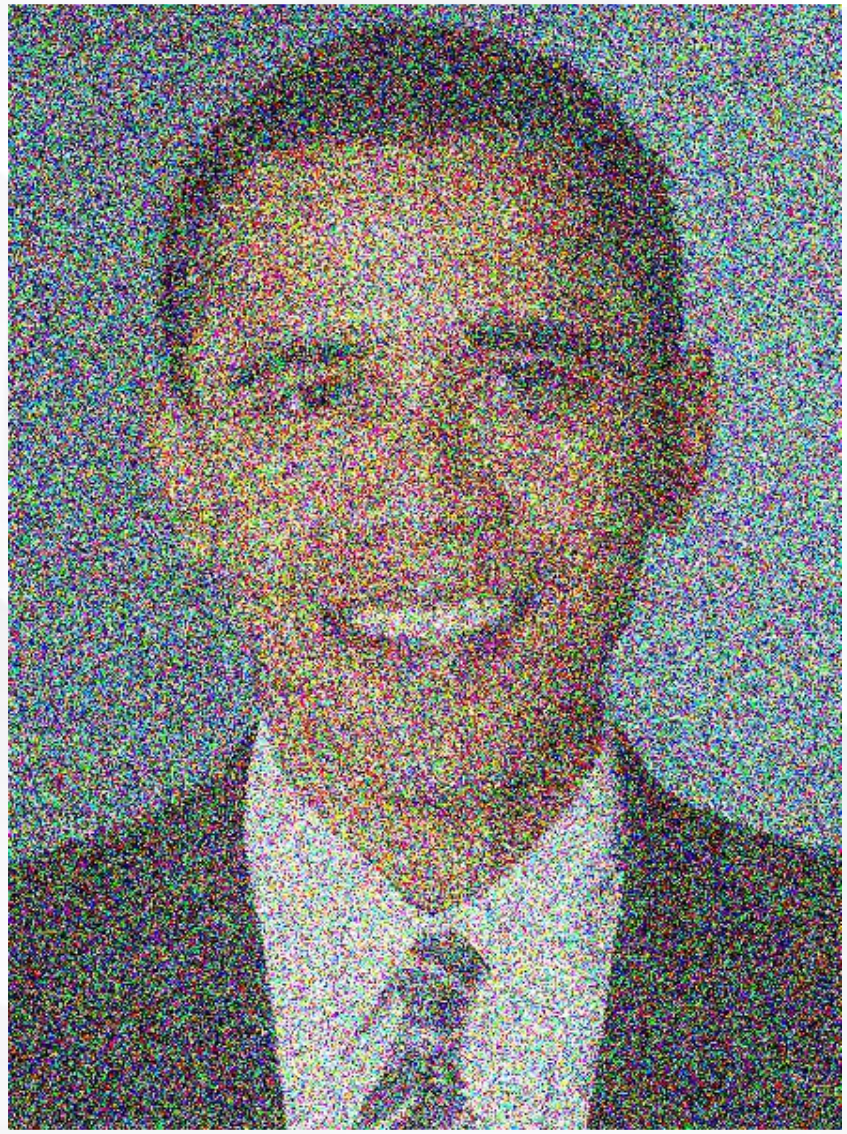
Machine Learning for computer vision

- Real Neural Networks
  - Human brain as a collection of biological neurons and synapses (1011 neurons, 104 synapse connections per neuron)
  - Powerful, adaptive and noise *resilient* pattern recognition
  - Combination of:
    - Memory
      - Memorization
    - Generalization
      - Learning “rules”
      - Learning “patterns”



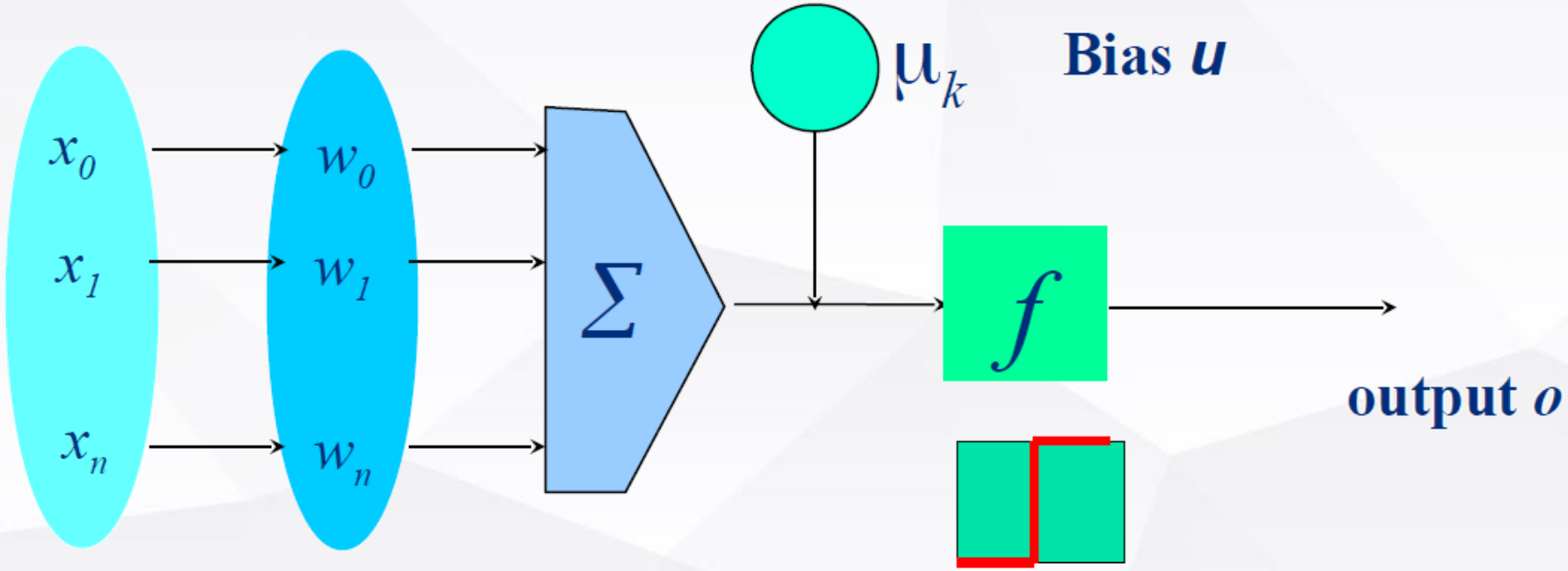
Machine Learning for computer vision

- Neural Networks
  - (biological and computational) are good at *noise resilient* pattern recognition
  - the human brain can cope with *extreme noise* in pattern recognition



Machine Learning for computer vision

- Artificial Neurons (≈ perceptrons)
  - A perceptron: An n-dimensional input vector x is mapped to output variable o by means of the scalar product with weights w, and a nonlinear function mapping f



[Han / Kamber 2006]

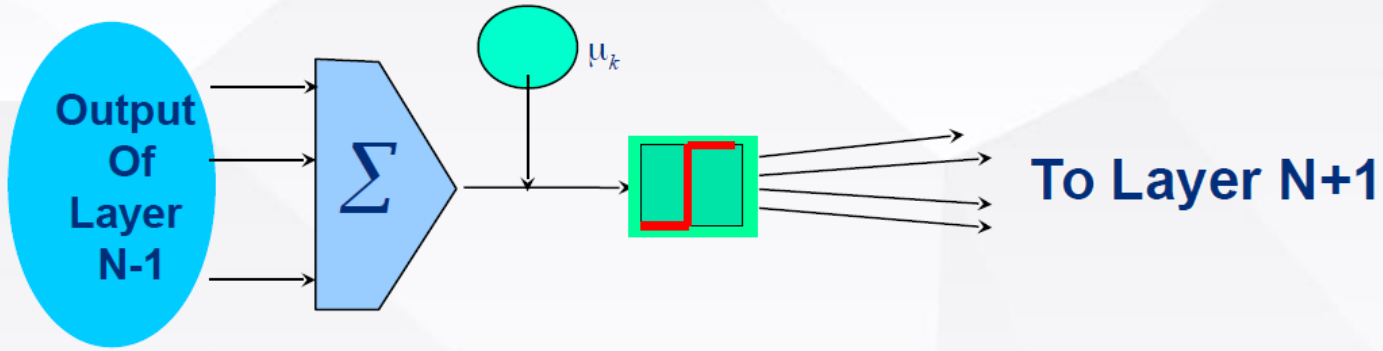
Machine Learning for computer vision

- Artificial Neurons (≈ perceptrons)
- Multiple Layers of Perceptrons
  - Multi Layer Perceptons (MLP)
- N Layers of M Perceptrons
  - Each fully connected (in graph sense to the next)
  - Every node of layer N takes outputs of all M nodes of layer N-1

[Han / Kamber 2006]

Machine Learning for computer vision

- In every node we have ....



- Input to network = (numerical) attribute vector
- describing classification examples
- Output of network = vector representing classification
  - e.g. {1,0}, {0,1}, {1,1}, {0,0} for classes A,B,C,D
  - or alt. {1,0,0}, {0,1,0}, {0,0,1} for classes A, B C

Essentially, **input to output** is
mapped as a **weighted sum**
occurring at **multiple**
(fully connected) **layers** in the network ....

If everything else is constant

...

(i.e. activation function, network topology)

… the weights are only thing
that changes

… so the **weights** are key.

Thus …

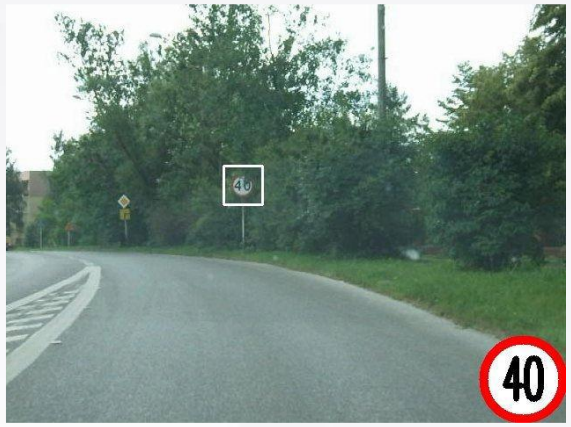
Setting the weights =training the network

- Backpropagation Summary
  - **Modifications** are made in the **"backwards" direction**: from the output layer, through each hidden layer down to the first hidden layer, hence **"Backpropagation"**
  - **Key Algorithmic Steps**
  - **Initialize** weights (to small random values) in the network
  - **Propagate the inputs forward** (by applying activation function) at each node
  - **Backpropagate the error backwards** (by updating weights and biases)
  - **Terminating** condition (when error is very small or enough iterations)

[Han / Kamber 2006]

Machine Learning for computer vision

- Example: speed sign recognition
  - Input:
    - Extracted binary text image
    - Scaled to 20x20 pixels
  - Network:
    - 30 hidden nodes
    - 2 layers
    - backpropogation
  - Output:
    - 12 classes
    - {10, 20, 30, 40, 50, 60, 70, 80, 90, 100, national-speed limit,non-sign}
  - Results: ~97% (success)



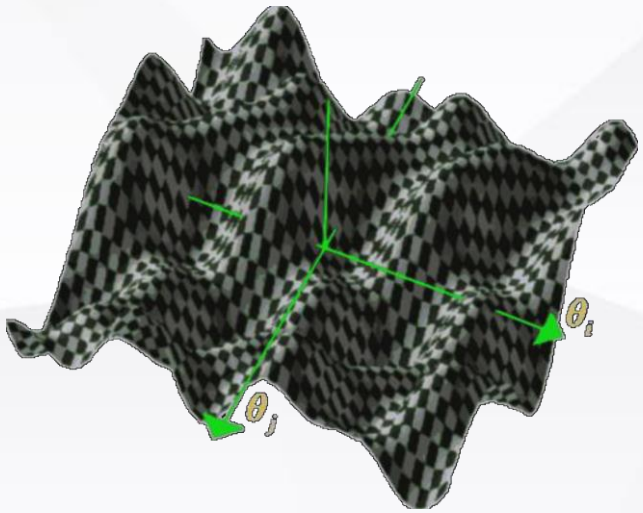
[Eichner / Breckon '08]

Machine Learning for computer vision

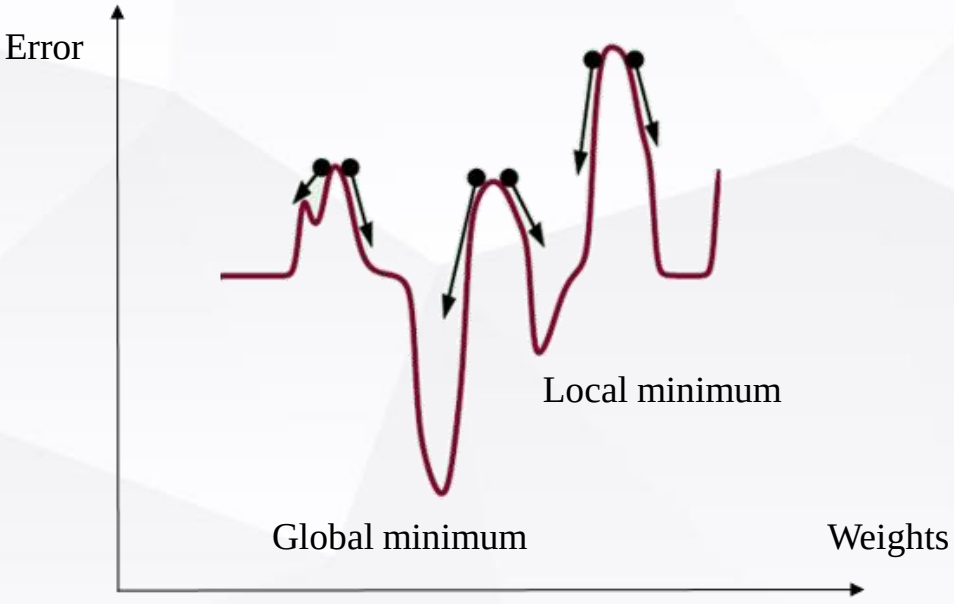
- Problems suited to ANNs
  - Termination of backpropagation
    - Too many iterations can lead to overfitting (to training data)
    - Too few iterations can fail to reduce output error sufficiently
  - Needs parameter selection
    - Learning rate (weight up-dates)
    - Network Topology (number of hidden nodes / number of layers)
    - Choice of activation function
    - ....
  - What is the network learning?
    - How can we be sure the correct (classification) function is being learned ?

Machine Learning for computer vision

- Problems suited to ANNs
  - May find local minimum within the search space of all possible weights (due to nature of backprop. gradient decent)
    - i.e. backpropagation is not guaranteed to find the best weights to learn the classification/regression function
    - Thus “learned” neural network may not find optimal solution to the classification/regression problem

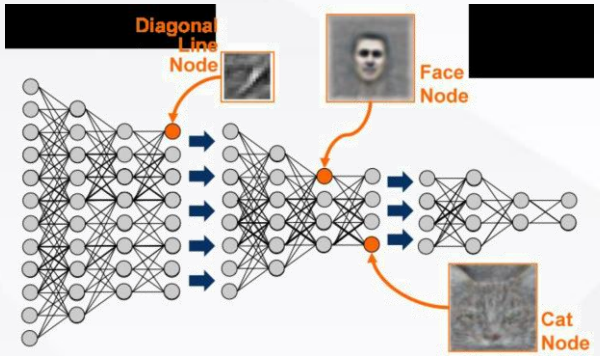


Weight space is always complex…



Machine Learning for computer vision

- … towards the *future* state of the art
  - Deep Learning (Deep Neural Networks)
    - multi-layer neural networks, varying layer sizes
    - varying levels of abstraction / intermediate feature representations
    - trained one layer at a time, followed by backprop.
    - complex and computationally demanding to train
  - Convolutional Neural Networks
    - leverage local spatial layout of features in input
    - locally adjacent neurons connected layer to layer instead of full layer to layer connectivity



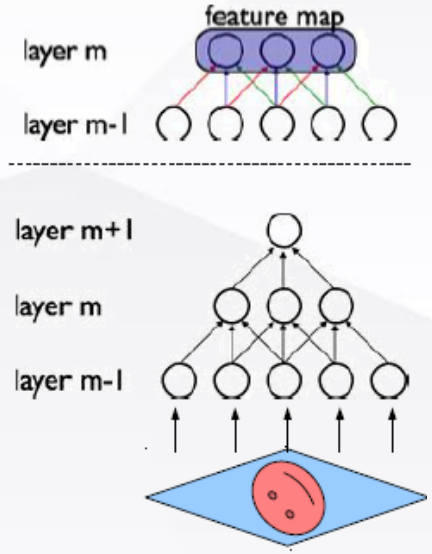


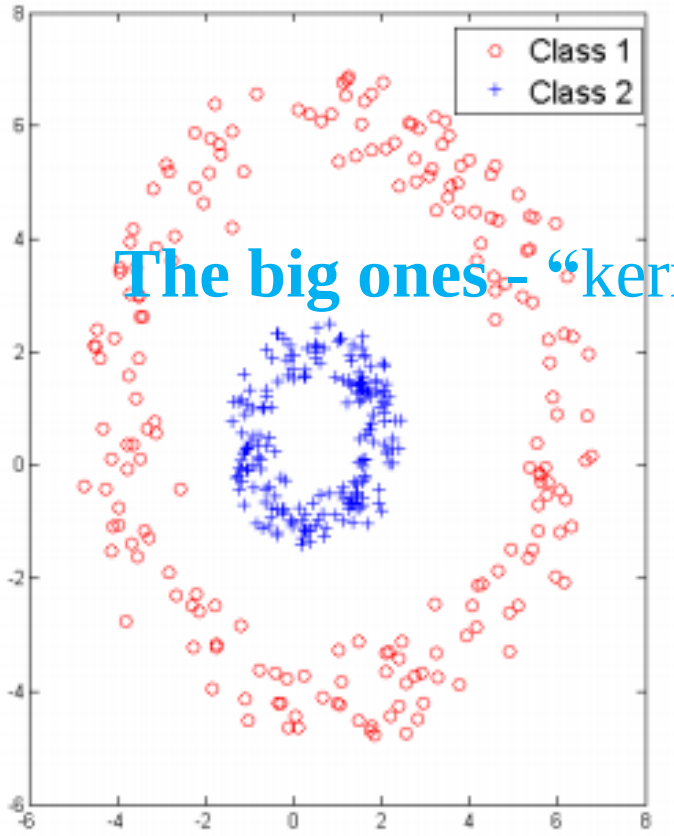
Image-Net Challenge - http://image-net.org/challenges/LSVRC/2013/ (1000 classes of object)

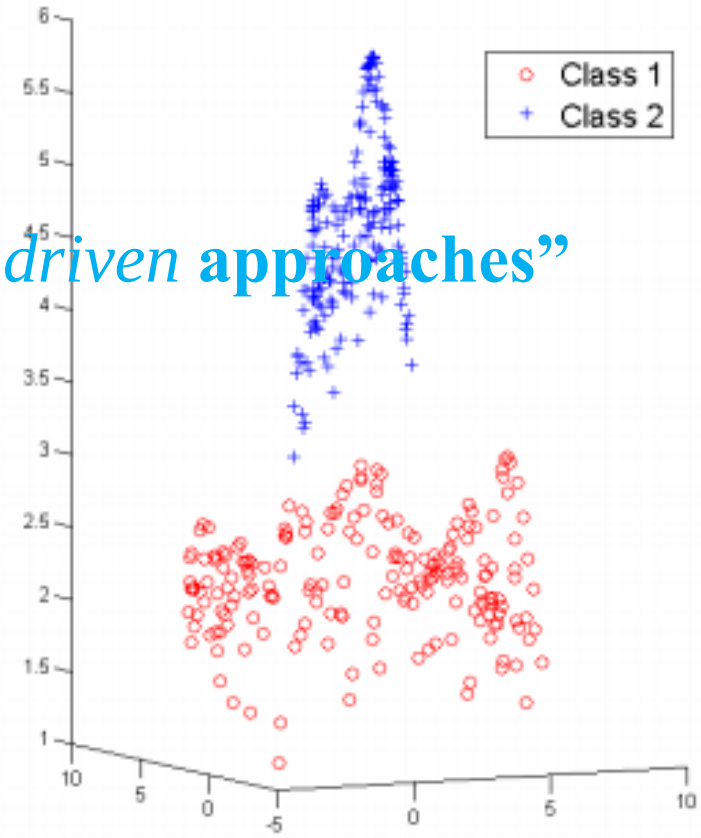
Deep Learning Neural Networks

Results are impressive.

But the same problems remain.

Machine Learning for computer vision





The big ones - "kernel *driven* **approaches"**

Machine Learning for computer vision

- Support Vector Machines
  - Basic Approach:
    - project instances into high dimensional space
    - learn linear separators (in high dim. space) with maximum margin
    - learning as optimizing bound on expected error
  - Positives
    - good performance on character recognition, text classification, ...
    - “appears” to avoid overfitting in high dimensional space
    - global optimization, thus avoids local minima
  - Negatives
    - applying trained classifier can be expensive (i.e. query time)

Machine Learning for computer vision

- Support Vector Machines

  - Note: "**Machines**" is just a sexy name (probably to make them sound different), they are really just computer algorithms like everything else in machine learning!

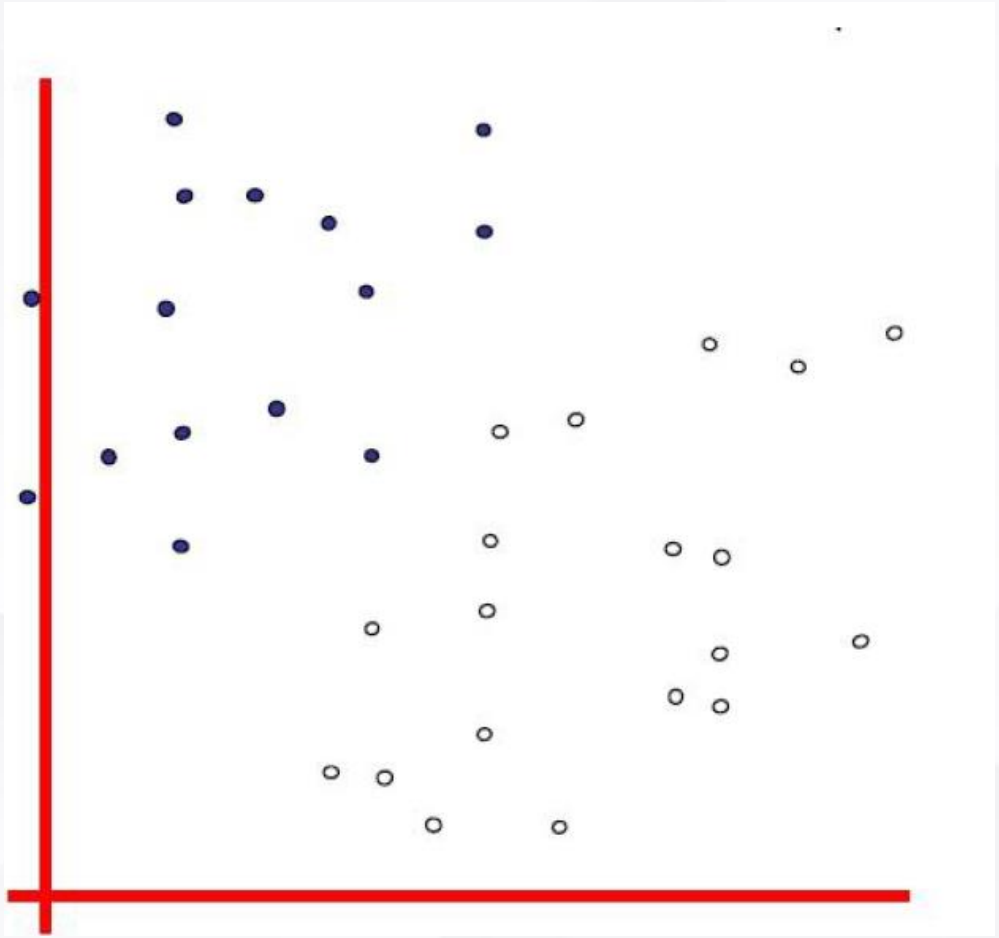
  - *… so don't get confused by the whole "machines" thing*

Machine Learning for computer vision

- Support Vector Machines
  - Simple example
    - denotes +1
    - denotes -1

e.g. gender recognition problem

{male, female}={+1,-1}



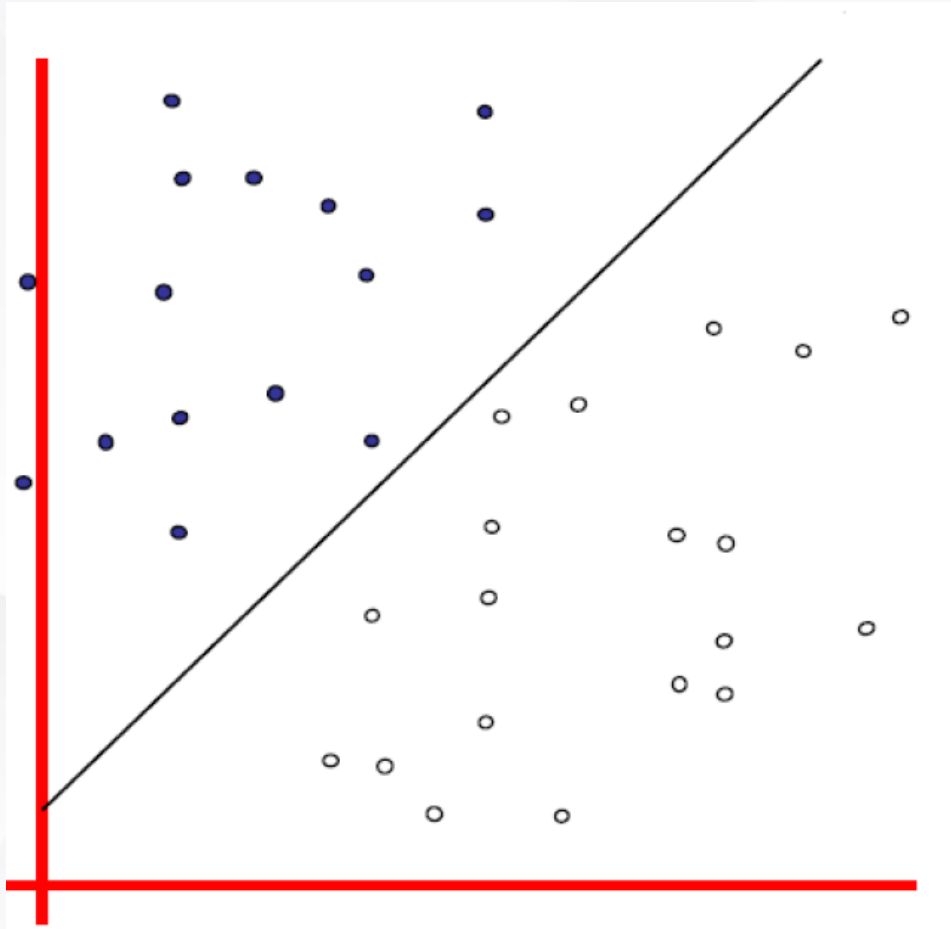
- How can we separate (classify) these data examples?)

Machine Learning for computer vision

- Support Vector Machines
  - Simple example
    - denotes +1
    - denotes -1

e.g. gender recognition problem

{male, female}={+1,-1}



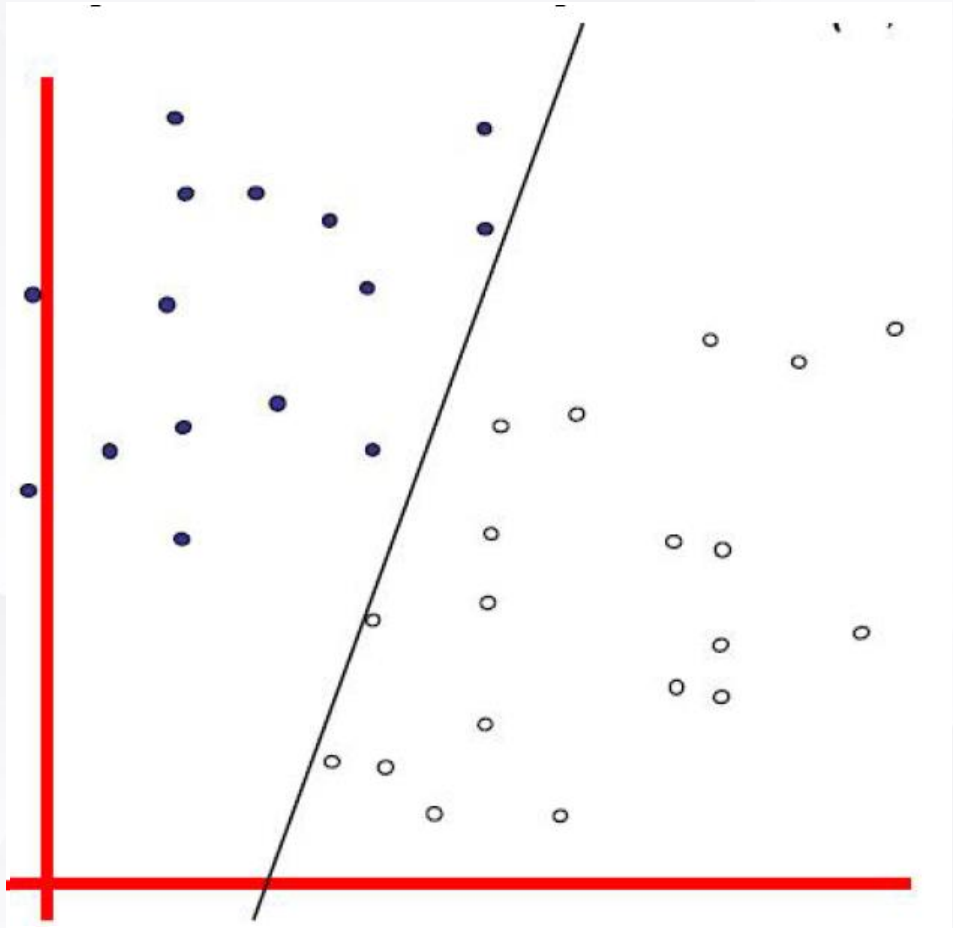
  - Linear separation

Machine Learning for computer vision

- Support Vector Machines
  - Simple example
    - denotes +1
    - denotes -1

e.g. gender recognition problem

{male, female}={+1,-1}



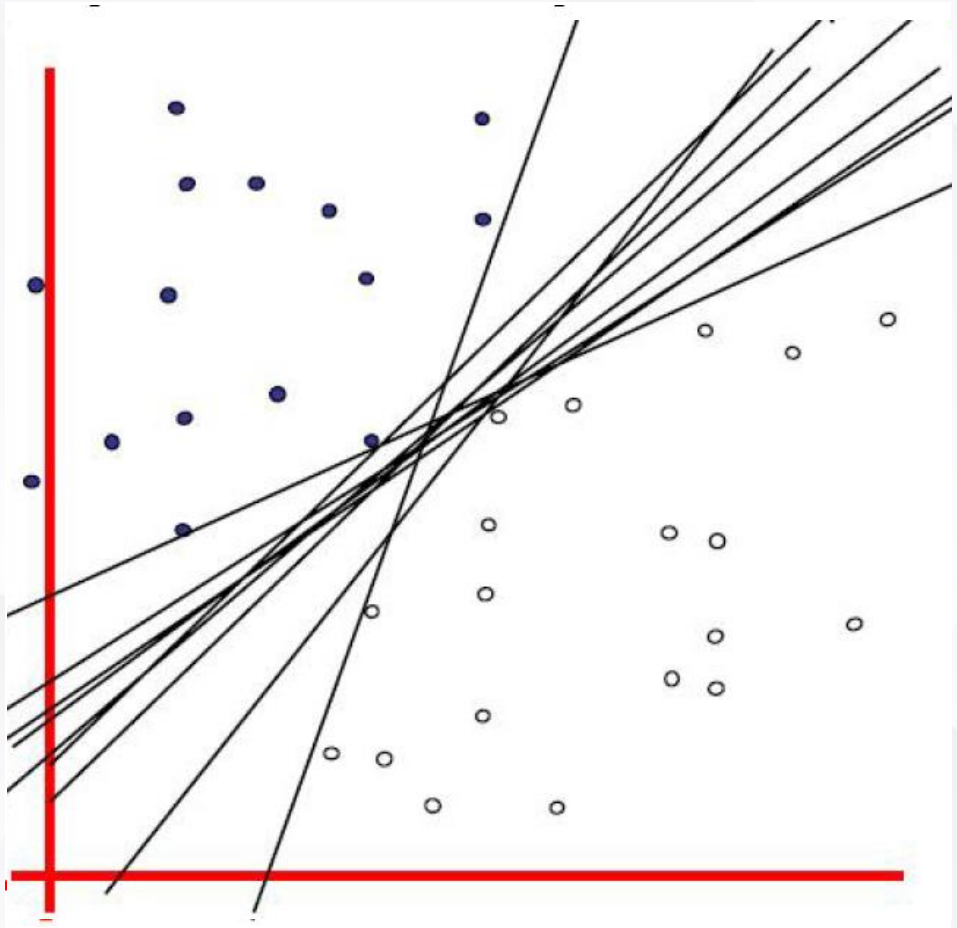
- Linear separation, but *which one*?

Machine Learning for computer vision

- Support Vector Machines
  - Simple example
    - denotes +1
    - denotes -1

e.g. gender recognition problem

{male, female}={+1,-1}



- Linear separation, but *which one*?

- Linear Separator

  - If we want a linear separator. We can use this as constraint satisfaction problem:

$$\vec{x_i} \cdot \vec{w} + b \geq +1 \; if \; y_i \equiv f(\vec{x_i}) = +1$$

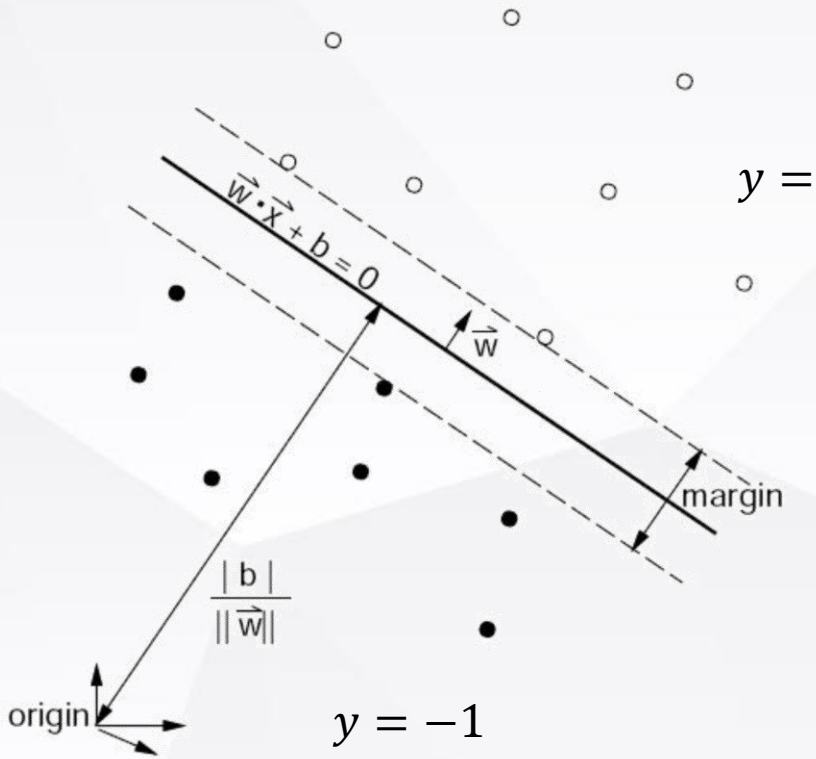
$$\vec{x_i} \cdot \vec{w} + b \leq -1 \; if \; y_i \equiv f(\vec{x_i}) = -1$$

Equivalenty:

$$y_i(\vec{x_i} \cdot \vec{w} + b) \geq 1$$

Instances (i.e, examples) $\{\mathbf{x_i}, \mathbf{y_i}\}$

- $\mathbf{x_i}$ = point in instance space (R_n) made up of n attributes
- $\mathbf{y_i}$ =class value for classification of $\mathbf{x_i}$



Note: we have a vector of weights coefficients

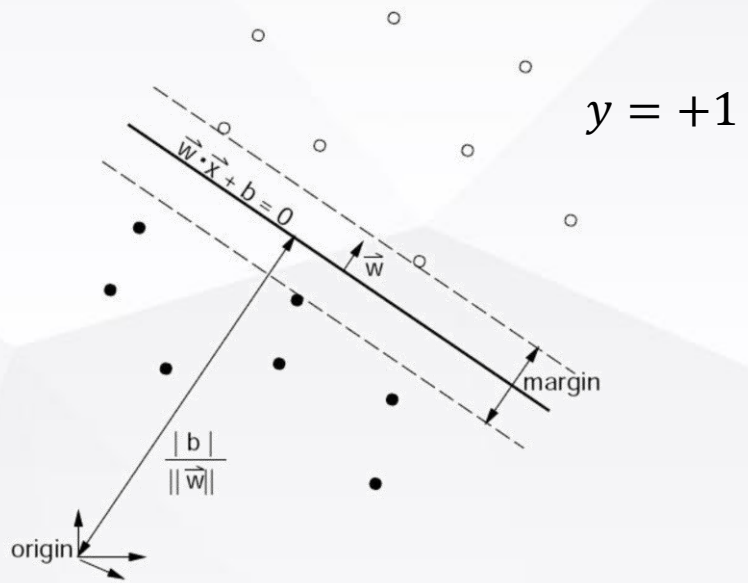
- Linear Separator

  - Now find the hyperplane (separator) with maximum margin.
  - thus if size of margin is defined as: $\frac{2}{||\vec{w}||}$
  - So view our problem as a constrained optimization problem:

$$min(||\vec{w}||^2)$$

$$y_i(\vec{x_i} \cdot \vec{w} + b) \geq 1, \forall i$$

"hyperplane" = separator boundary
hyperplane in R2 == 2D line



- What about Non-separable Training Sets ? Separator
  - Add a "slack variable" $\xi_i \geq 0$ for each $< x_i, y_i >$:

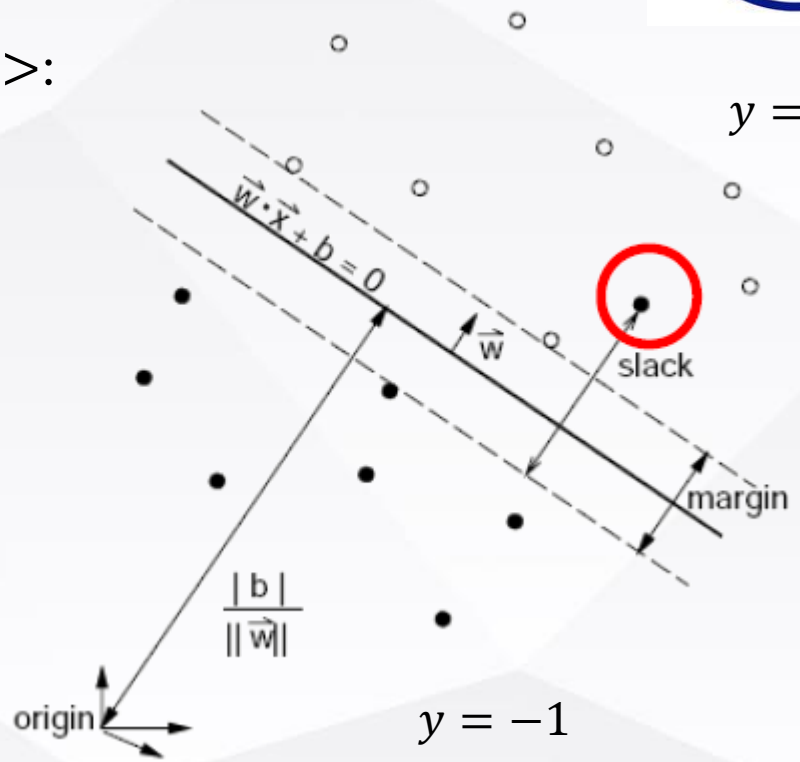
$$min\left(||\vec{w}||^2 + C\left(\sum_i \xi_i\right)^k\right), subject\ to$$

$$\vec{x_i} \cdot \vec{w} + b \geq +1 - \xi\ if\ y_i = +1$$

$$\vec{x_i} \cdot \vec{w} + b \leq -1 + \xi\ if\ y_i = -1$$

Penalty term > 0 for each "wrong side of the boundary" case.

C is picked by hand.

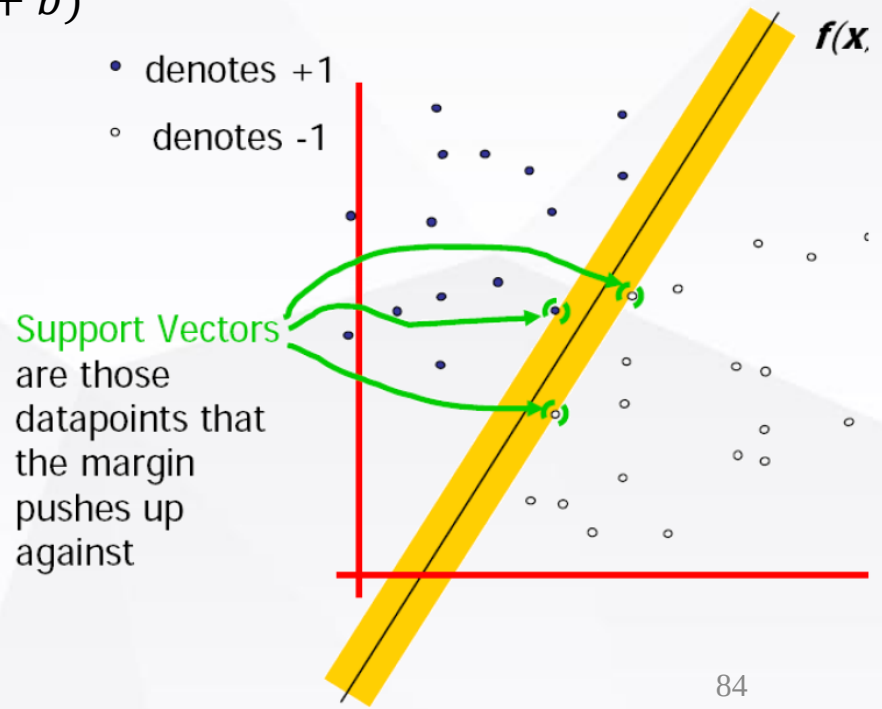


  - As a matter of experience:
    - Smaller C leads to smaller margin with smaller error
    - Larger C leads to larger margin with larger error

- Linear Separator
  - Separator margin determined by just a few examples
  - We call these support vectors
    - We can define separator in terms of support vectors and classifier examples x as:

$$f(\vec{x}) \leftarrow sgn(\sum_{s_i \in support\ vectors} w_i \overrightarrow{s_i} \cdot \vec{x} + b)$$

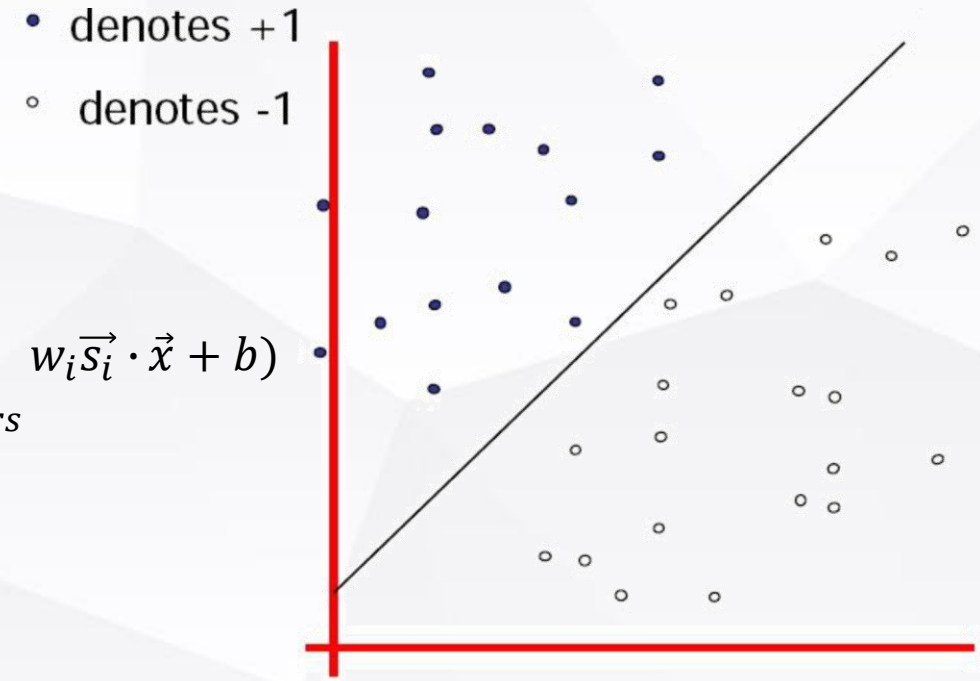
    - Support vectors = sub-set of training instances that define decision boundary between classes



- How do we classify a new example ?
  - New unseen (test) example attribute vector x
  - Set of support vectors $\{s_i\}$ with weights $\{w_i\}$, bias b
    - define the “hyperplane” boundary in the original dimension (this is a linear case)

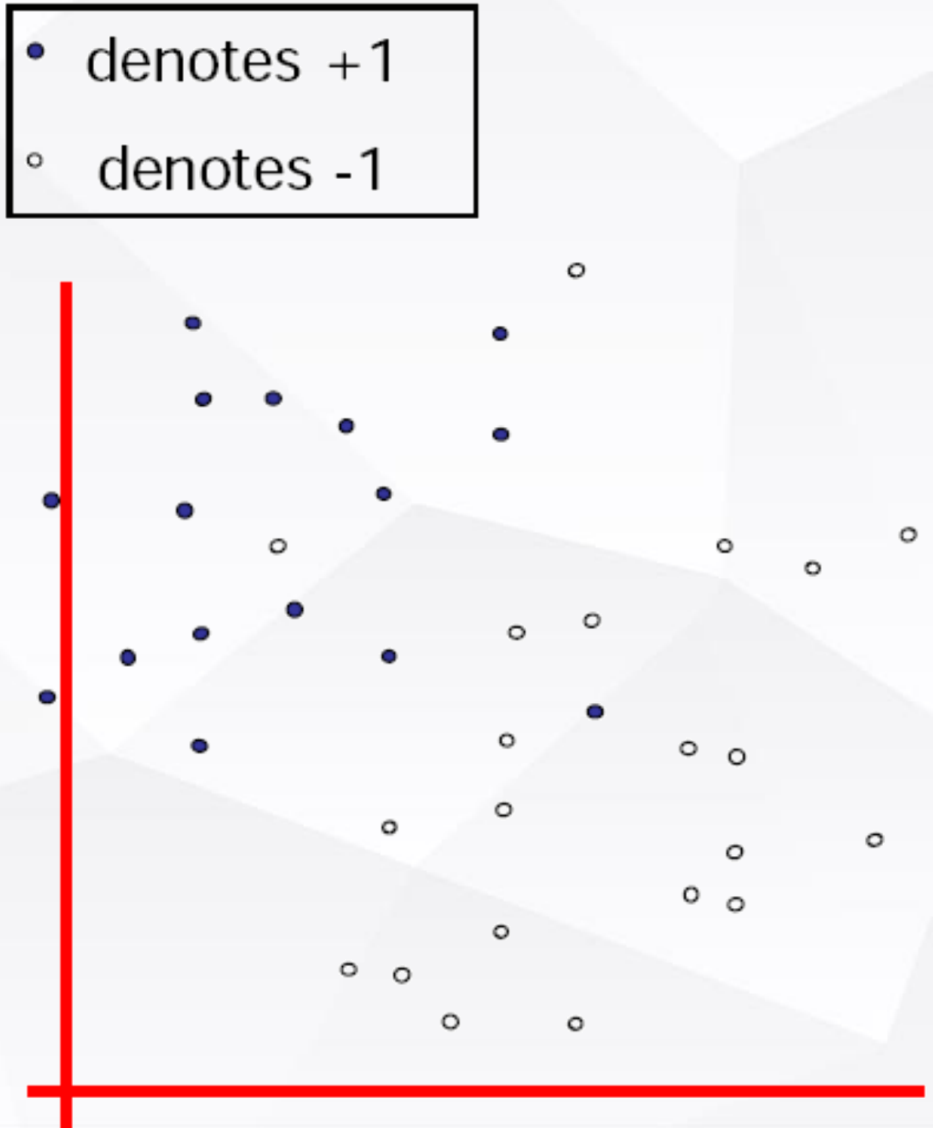
$$f(\overrightarrow{x_{new}}) = sgn(\sum_{s_i \in support\ vectors} w_i \overrightarrow{s_i} \cdot \vec{x} + b)$$

$$i.e.\ f(\overrightarrow{x_{new}}) = \{-1, +1\}$$



Machine Learning for computer vision

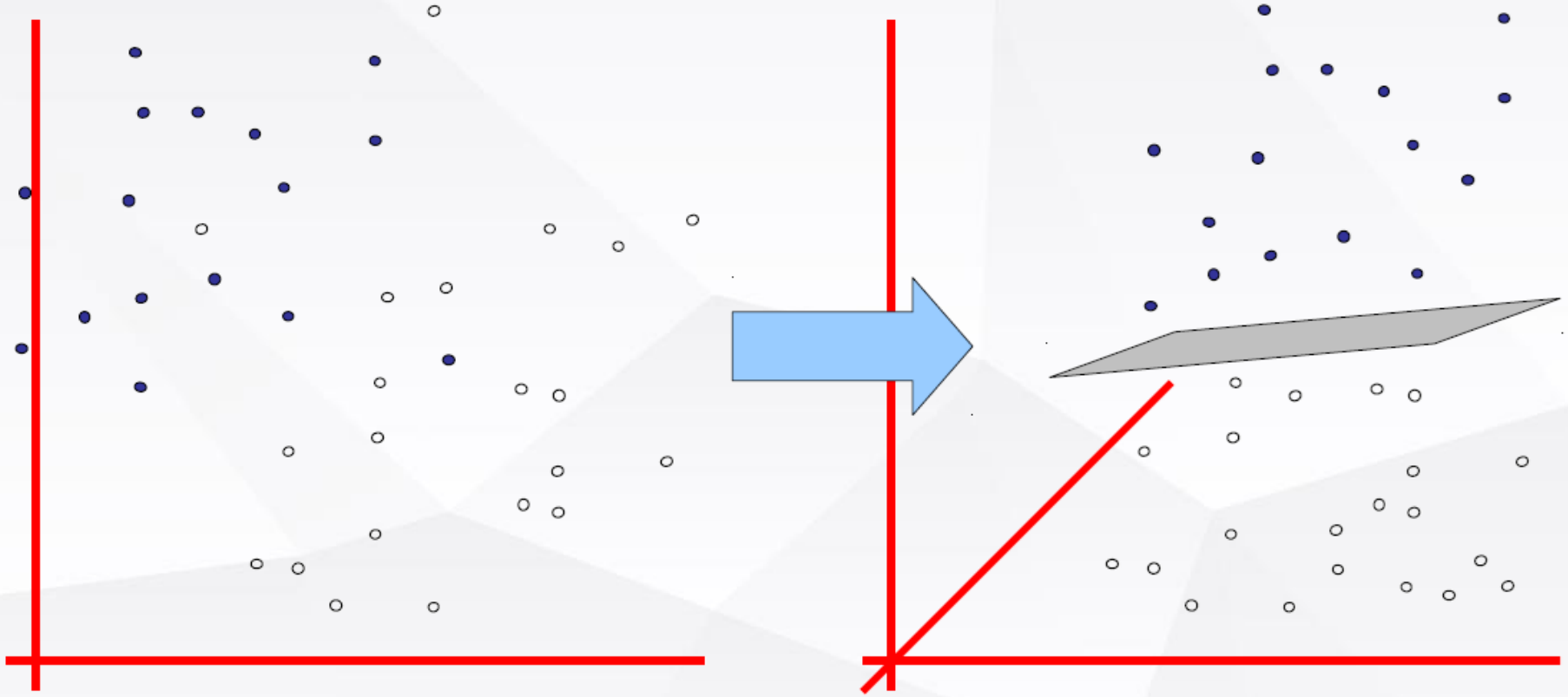
- What about this ..... ?



Not linearly
Separable !

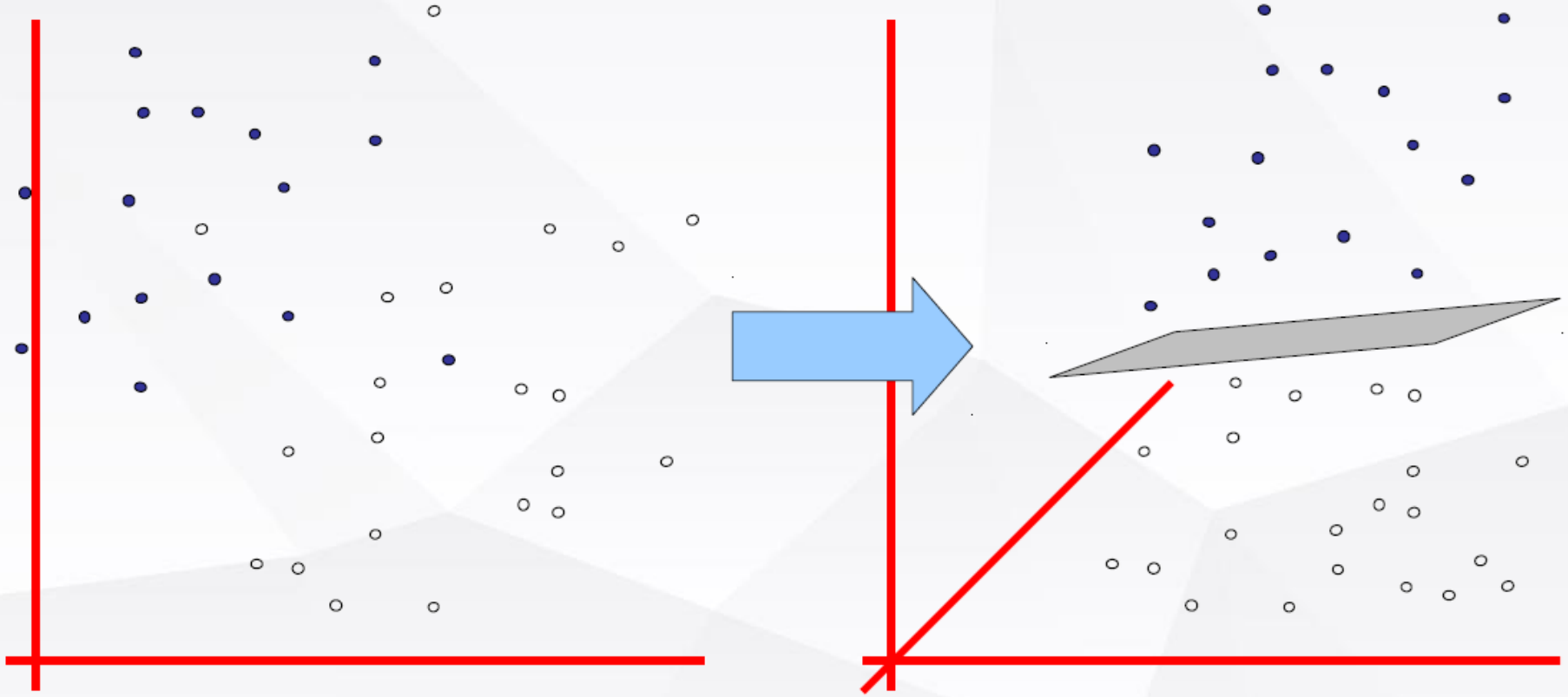
Machine Learning for computer vision

- e.g. projection from 2D to 3D
  - Linearly separable with a hyperplane!



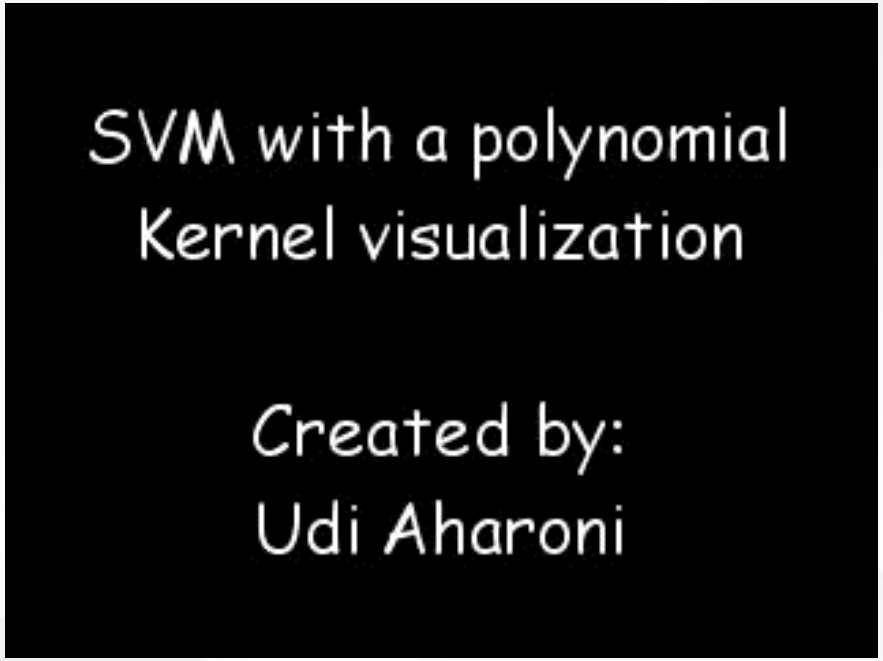
Machine Learning for computer vision

- e.g. projection from 2D to 3D
  - Linearly separable with a hyperplane!



Machine Learning for computer vision

- e.g. projection from 2D to 3D
  - Linearly separable with a hyperplane!



https://www.youtube.com/watch?v=3liCbRZPrZA

- Non Linear SVMs
  - Just as before but now with the kernel ....

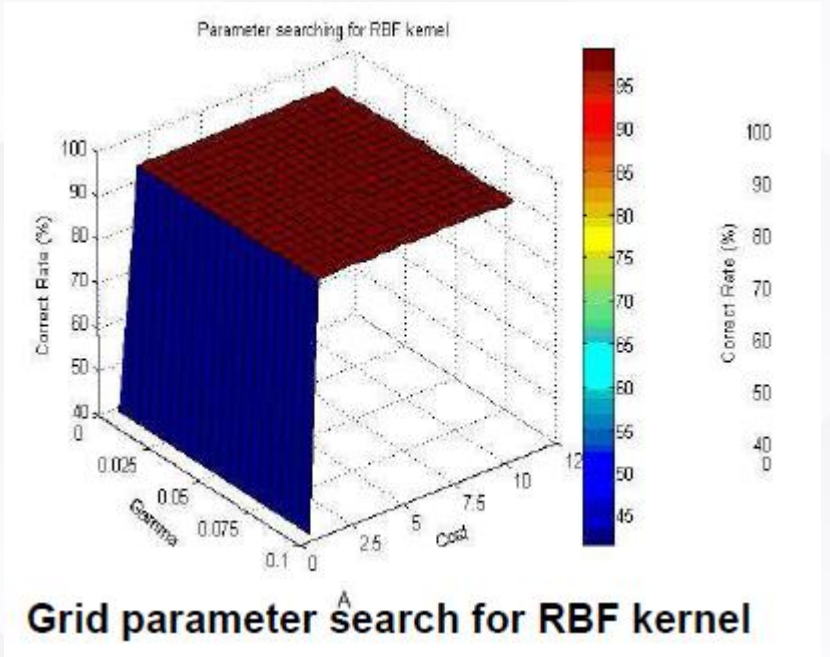
Linear SVM $$f(\vec{x}) \leftarrow sgn(\sum_{s_i \in support\ vectors} w_i \overrightarrow{s_i} \cdot \vec{x} + b)$$

Nonlinear SVM $$f(\vec{x}) \leftarrow sgn(\sum_{s_i \in support\ vectors} w_i K(\overrightarrow{s_i} \cdot \vec{x}) + b)$$

Linear SVM is just a special case of $K(\overrightarrow{s_i} \cdot \vec{x})$

Machine Learning for computer vision

- Choosing Kernels?
  - Kernel functions (commonly used)
    - Polynomial function of degree p
    - Gaussian radial basis function (size σ)
- Choosing Parameters?
  - Commonly chosen by grid search of parameters space

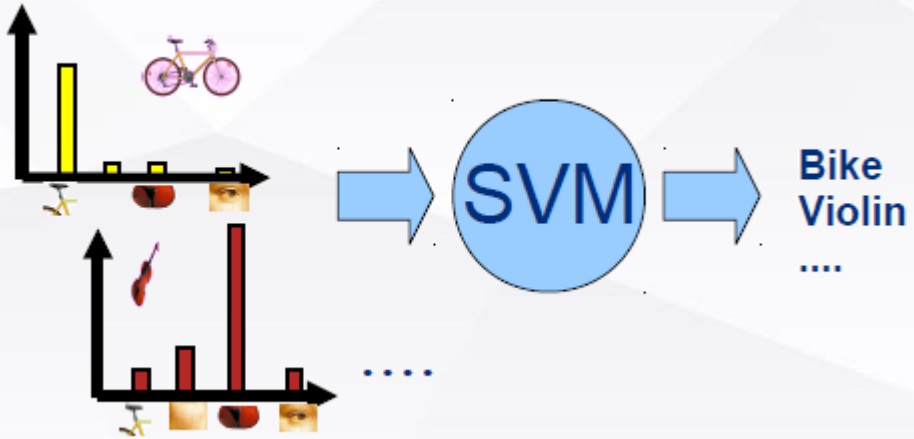


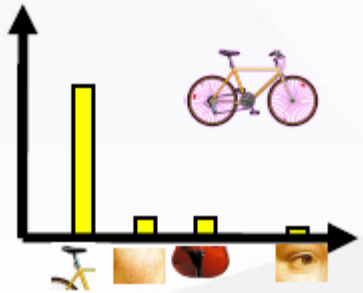
Grid parameter search for RBF kernel

Machine Learning for computer vision

- Application to Image Classification
  - Common Model - Bag of Visual Words
  - 1. build histograms of feature occurrence over training data (features: SIFT, SURF, MSER ....)
  - 2. Use histograms as input to SVM (or other ML approach)

Cluster Features in R_n space





Cluster *"membership"* creates a **histogram of feature occurrence**

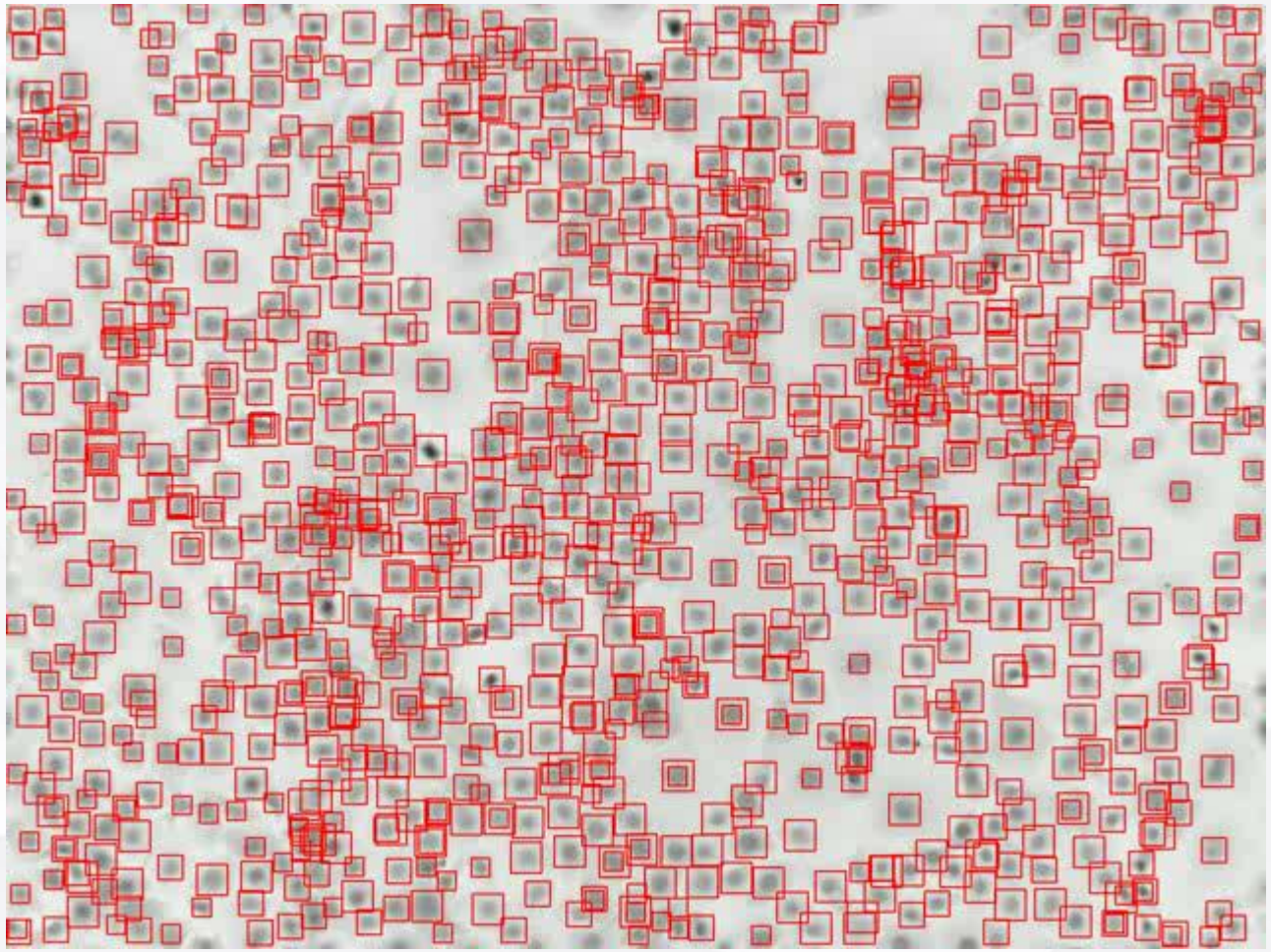
Machine Learning for computer vision

- Application to Image Classification
  - Bag of Words Model (SURF + SVM + Decision Forest)

https://www.durham.ac.uk/toby.breckon/demos/multimodal/

Machine Learning for computer vision

- Searching for Cell Nuclei Locations with SVM



http://www.durham.ac.uk/toby.breckon/demos/cell

Machine Learning for computer vision

- "Approx" State of the Art
  - Recent approaches – SVM (+ variants), Decision Forests (+ variants), Boosted Approaches, Bagging ....
    - outperform standard Neural Networks
    - can find maximally optimal solution (SVM)
    - less prone to over-fitting (in theory)
    - allow for extraction of meaning (e.g. if-then-else rules for tree based approaches)
  - Several other ML approaches
    - clustering – k-NN, k-Means in multi-dimensional space
    - graphical models
    - Bayesian methods
    - Gaussian Processes

but then **Deep Learning** (generally) outperforms everything…

Machine Learning for computer vision

- But how do we *evaluate* how well it is working?
  - and produce convincing results for our papers and funders…

Machine Learning for computer vision

- Evaluating Machine Learning
  - For classification problems ....
  - True Positives (TP) 
    - example correctly classified as an +ve instance of given class A
  - False Positives (FP)
    - example wrongly classified as an +ve instance of given class A
    - i.e. it is not an instance of class A
  - True Negatives (TN) 
    - example correctly classified as an -ve instance of given class A
  - False Negatives (FP)
    - example wrongly classified as an -ve instance of given class A
    - i.e. classified as not class A but is a true class A

- Evaluating Machine Learning
  - Confusion Matrices
  - Table of TP, FP, TN, FN
  - can also show TP, FP, TN, FN weighted by cost of mis-classification Vs. true classification

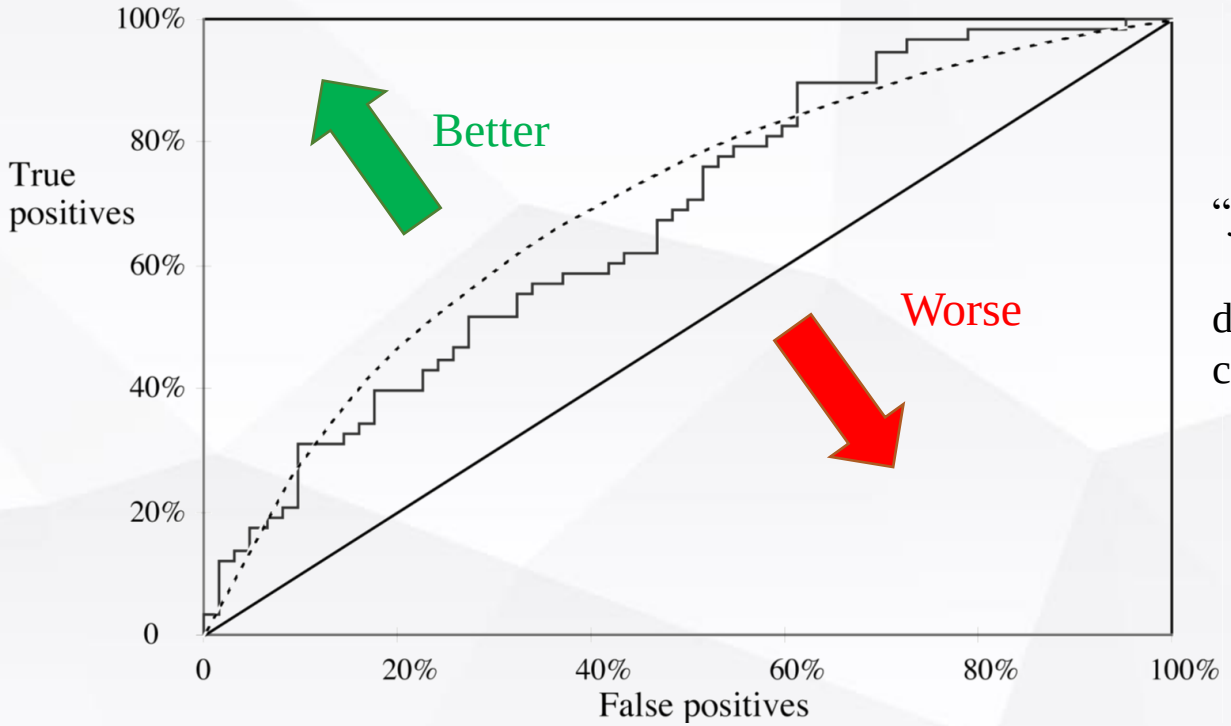
		Predicted class	
Actual class		Yes	No
	Yes	True positive	False negative
	No	False postivie	True negative

A common name for "actual class" (i.e. the true class) is "*ground truth*" or "*the ground truth data*".

Machine Learning for computer vision

- Receiver Operating Characteristic (ROC) Curve
  - used to show trade-off between hit rate and false alarm rate over noisy channel (in communications originally)
    - here a “noisy” (i.e. error prone) classifier

% TP Vs. %FP

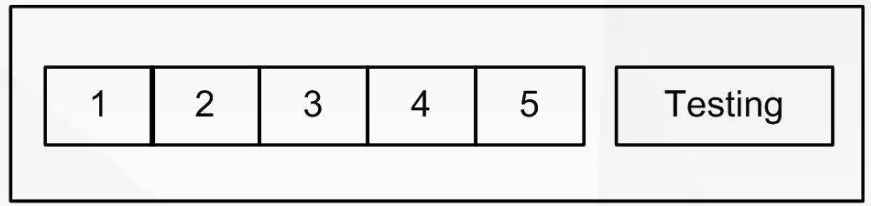


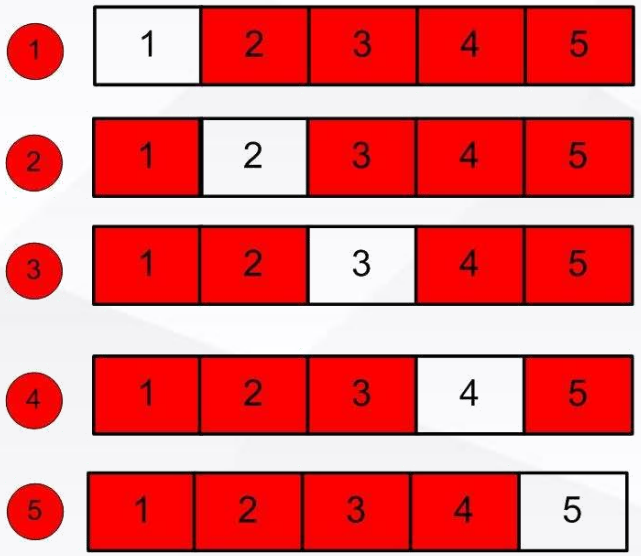
“jagged steps” = actual data

dashed line= averaged result (over multiple cross-validation folds) or best line fit

Machine Learning for computer vision

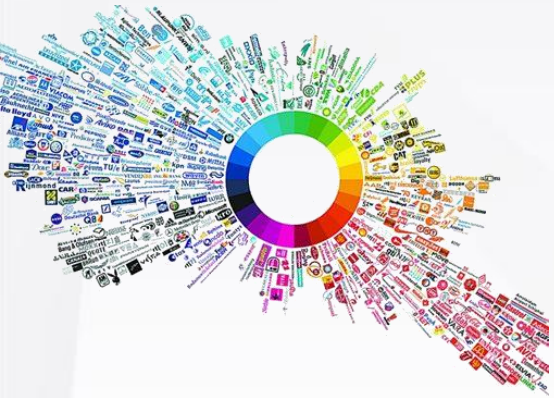
- Evaluating Machine Learning
  - The key is:
  - Robust experimentation on independent training and testing sets





Machine Learning for computer vision

- Data is important!
  - **No free lunch!**
  - .... the idea that it is impossible to get something for *nothing*
    - This is very true in Machine Learning
    - approaches that train quickly or require little memory or few training examples produce poor results
    - if you have poor data → you get poor learning
    - problems with data = problems with learning
    - problems ={not enough data, poorly labelled, biased, unrepresentative … }

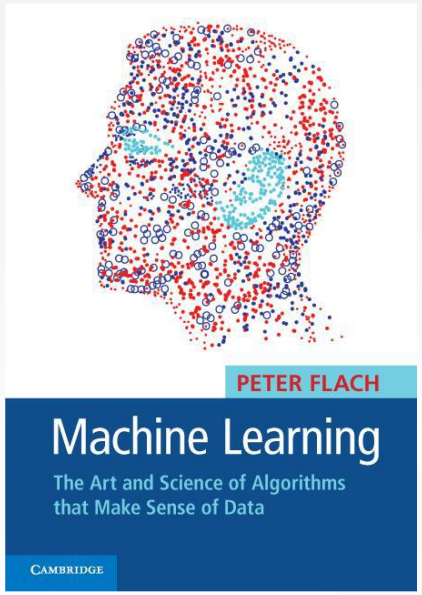


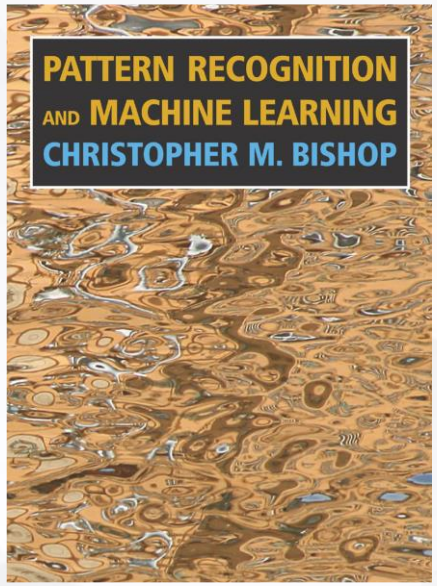
Machine Learning for computer vision

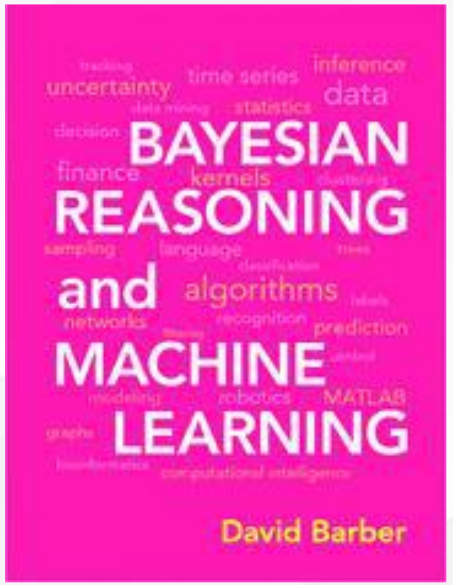
- What we have seen…
  - The power of combining simple things ….
    - Ensemble Classifiers
    - Decision Forests / Random Forests
    - concept extends to all ML approaches
  - Neural Inspired Approaches
    - Neural Networks
    - many, many variants
  - Kernel Inspired Approaches
    - Support Vector Machines
    - beginning of the story

Machine Learning for computer vision

- Further Reading…







Machine Learning for computer vision

- Further Reading…
  - **Decision Forests: A Unified Framework for Classification, Regression, Density**
  - **Estimation, Manifold Learning and Semi-Supervised Learning** - A. Criminisi et al., in Foundations and Trends in Computer Graphics and Vision, 2012.
  - **OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks,** P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun:
  - International Conference on Learning Representations 2014.
  - **Image Classification using Random Forests and Ferns**
    - A Bosch, A Zisserman, X Munoz – Int. Conf. Comp. Vis., 2007.
  - **Robust Real-time Object Detection**
    - P Viola, M Jones - International Journal of Computer Vision, 2004.
  - **The PASCAL Visual Object Classes (VOC) challenge**
    - M Everingham, L Van Gool, C Williams, J Winn, A Zisserman - International Journal of Computer Vision, 2010.

Deep Learning in Computer Vision

Hot topic – very fast moving

But it's *another* story…

Thank You