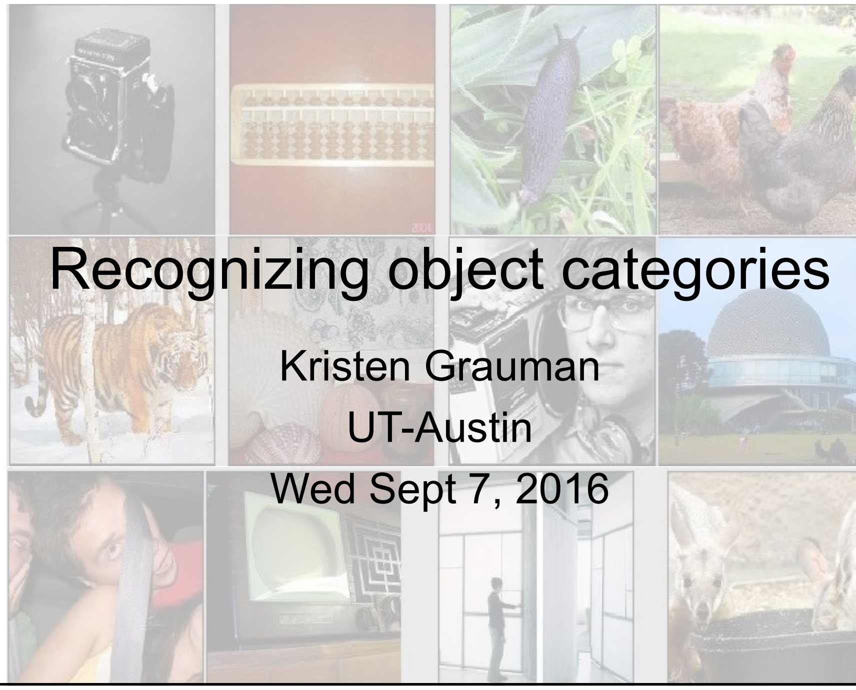


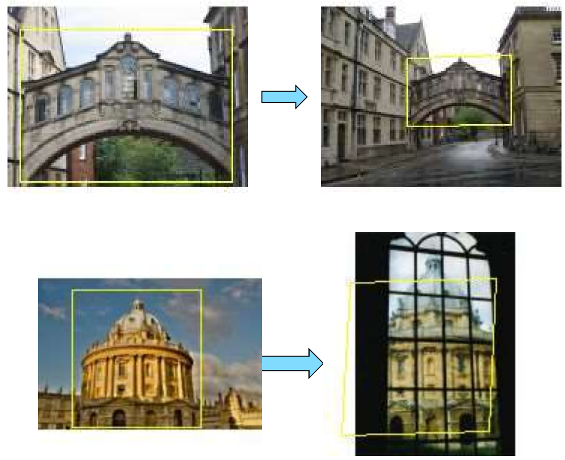
Recognizing object categories

Kristen Grauman
UT-Austin
Wed Sept 7, 2016

Announcements

- Reminders:
  - Assignment 1 due Sept 16 11:59 pm on Canvas
  - Optional CNN/Caffe tutorial on Monday Sept 12, 5-7 pm
  - No laptops, phones, tablets, etc. in class
- Thoughts on review sharing?
- Questions about presentations, experiments, discussion proponent/opponent?

Last time: Recognizing instances



Last time: Recognizing instances

- 1. Basics in feature extraction: filtering
- 2. Invariant local features
- 3. Recognizing object instances

Instance recognition: remaining issues

- How to summarize the content of an entire image? And gauge overall similarity?
- How large should the vocabulary be? How to perform quantization efficiently?
- Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?

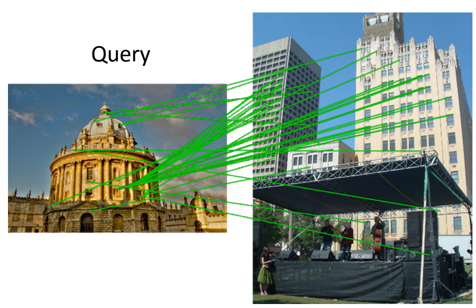
Kristen Grauman

Spatial Verification

Query

DB image with high BoW similarity

Query



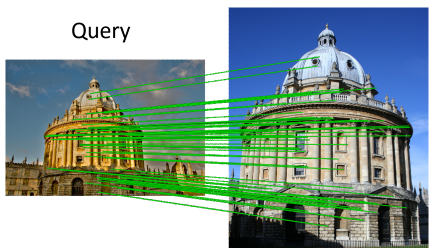
DB image with high BoW similarity

Both image pairs have many visual words in common.

Slide credit: Ondrej Chum

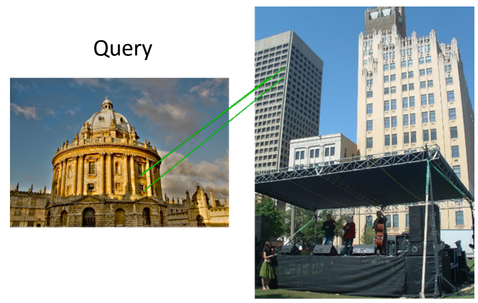
Spatial Verification

Query



DB image with high BoW similarity

Query



DB image with high BoW similarity

Only some of the matches are mutually consistent

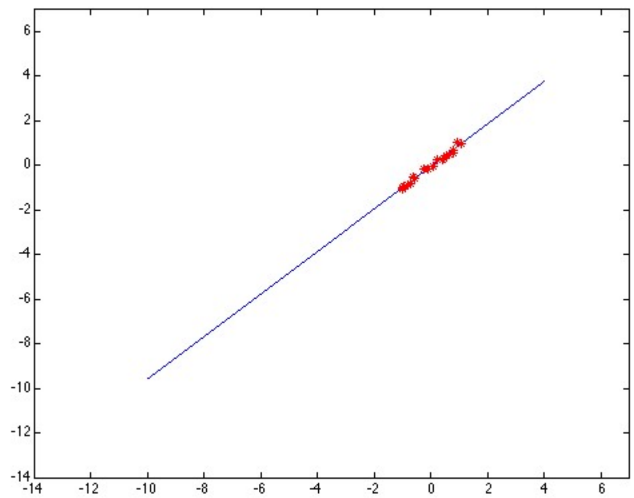
Slide credit: Ondrej Chum

Spatial Verification: two basic strategies

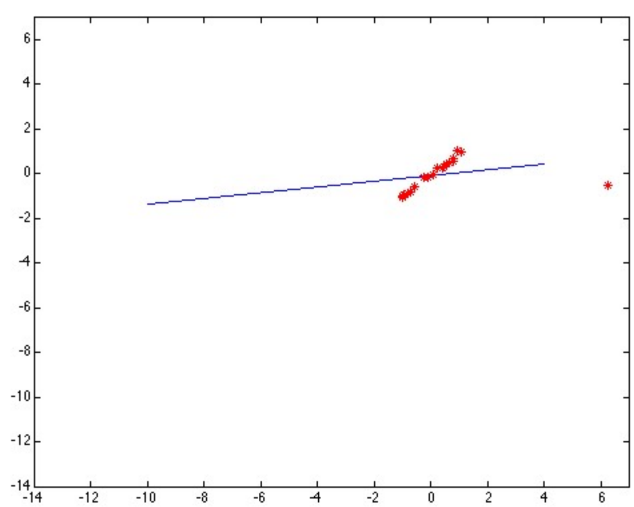
- RANSAC
- Generalized Hough Transform

Slide credit: Kristen Grauman

Outliers affect least squares fit



Outliers affect least squares fit



RANSAC

- RANdom Sample Consensus
- **Approach**: we want to avoid the impact of outliers, so let's look for "inliers", and use those only.
- **Intuition**: if an outlier is chosen to compute the current fit, then the resulting line won't have much support from rest of the points.

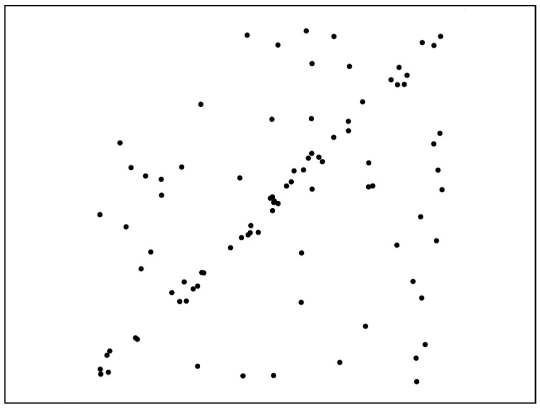
RANSAC for line fitting

Repeat ***N*** times:

- Draw ***s*** points uniformly at random
- Fit line to these ***s*** points
- Find inliers to this line among the remaining points (i.e., points whose distance from the line is less than ***t***)
- If there are ***d*** or more inliers, accept the line and refit using all inliers

Lana Lazebnik

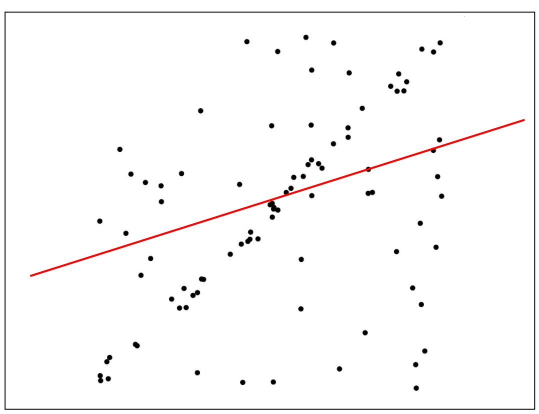
RANSAC for line fitting example



Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

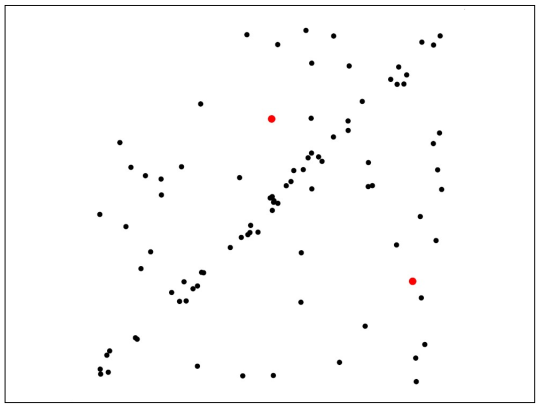


Least-squares fit

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

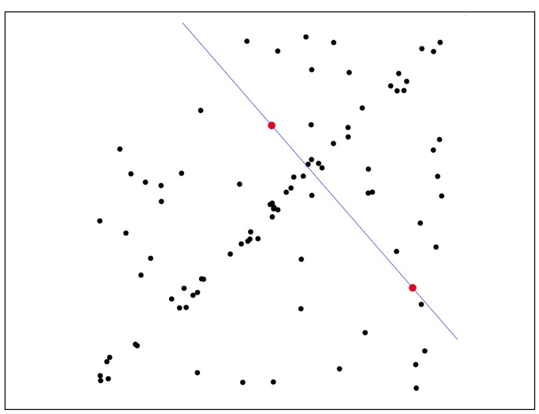


1. Randomly select minimal subset of points

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

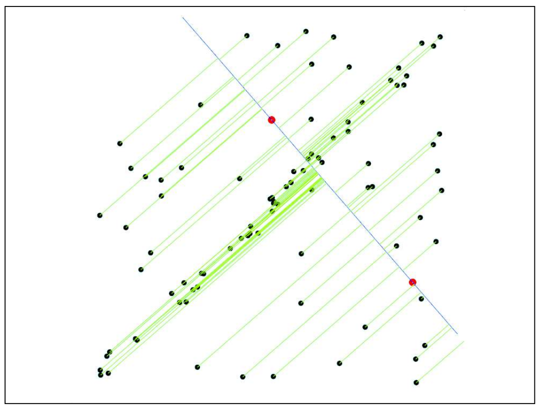


1. Randomly select minimal subset of points
2. Hypothesize a model

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

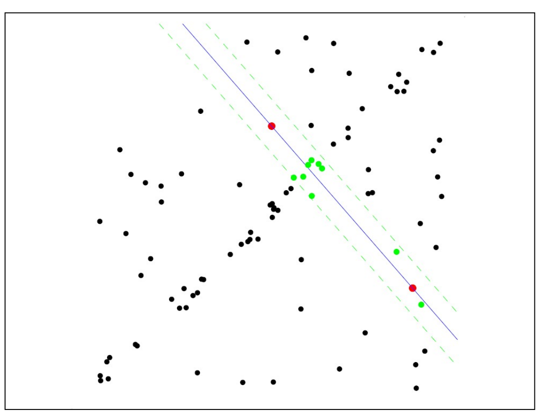


1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

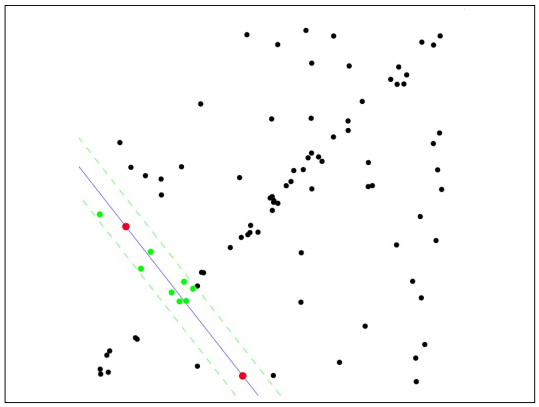


1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

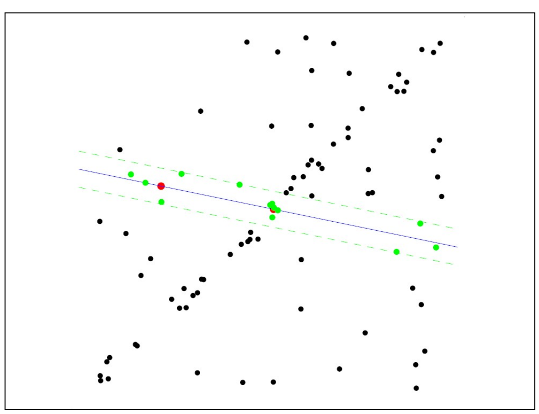


1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example



1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

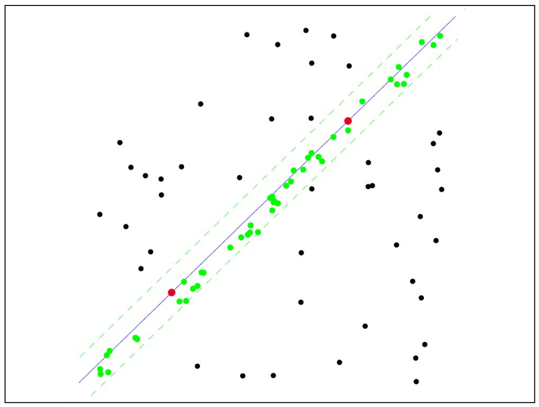
24

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

Uncontaminated sample



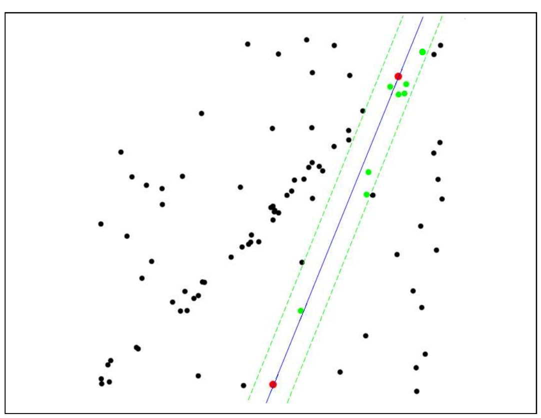
1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

25

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example



1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

Lana Lazebnik

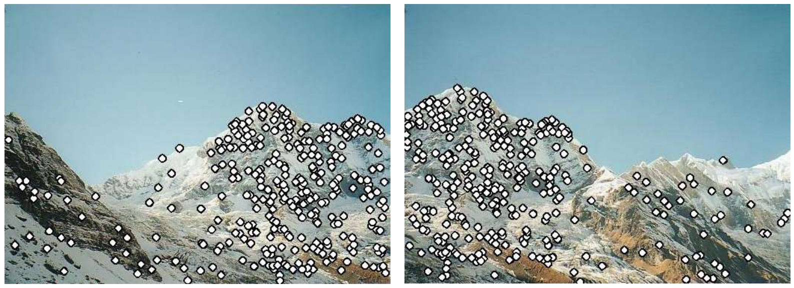
That is an example fitting a model (line)…

What about fitting a transformation (translation, affine…)?

Robust feature-based alignment

Source: L. Lazebnik

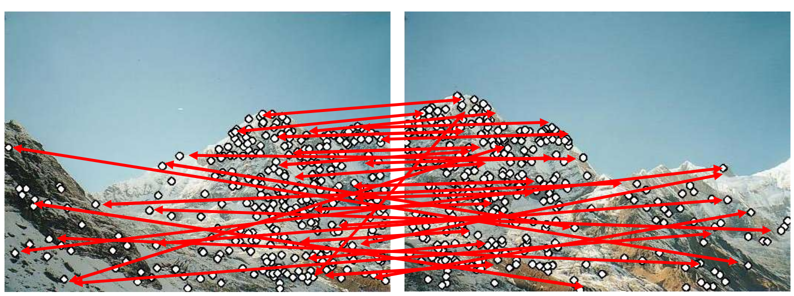
Robust feature-based alignment



- Extract features

Source: L. Lazebnik

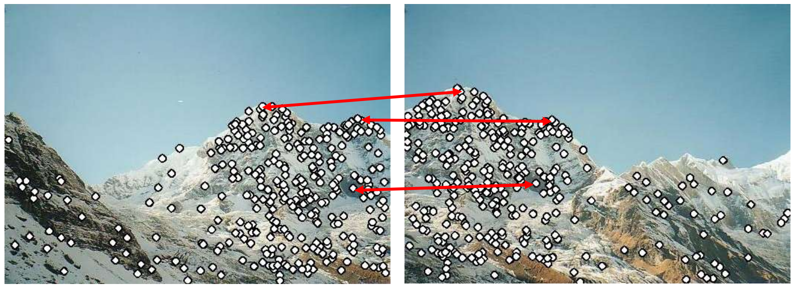
Robust feature-based alignment



- Extract features
- Compute *putative matches*

Source: L. Lazebnik

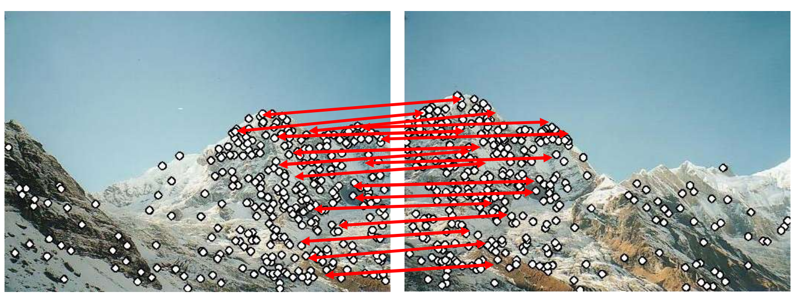
Robust feature-based alignment



- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)

Source: L. Lazebnik

Robust feature-based alignment



- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)
  - *Verify* transformation (search for other matches consistent with *T*)

Source: L. Lazebnik

Robust feature-based alignment

- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)
  - *Verify* transformation (search for other matches consistent with *T*)

Source: L. Lazebnik

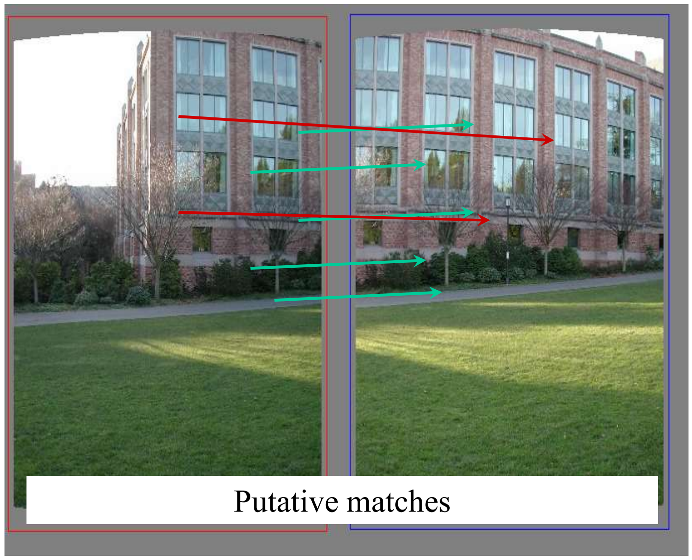
RANSAC: General form

- RANSAC loop:

1. Randomly select a *seed group* of points on which to base transformation estimate
2. Compute model from seed group
3. Find *inliers* to this transformation
4. If the number of inliers is sufficiently large, re-compute estimate of model on all of the inliers

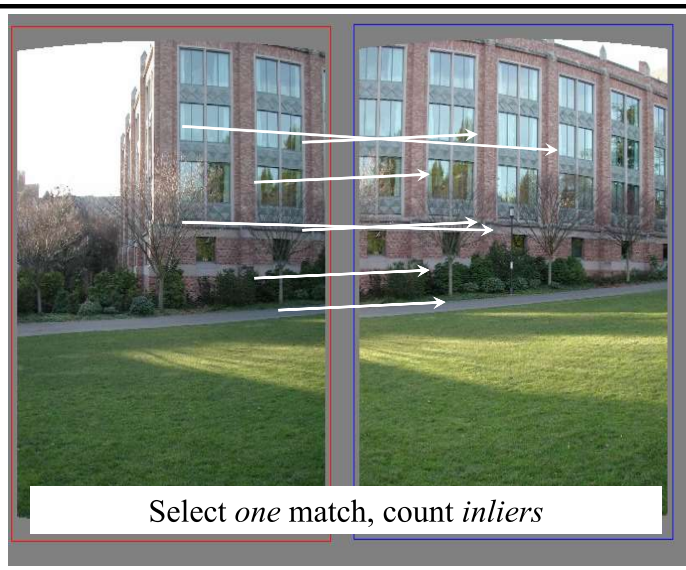
- Keep the model with the largest number of inliers

RANSAC example: Translation

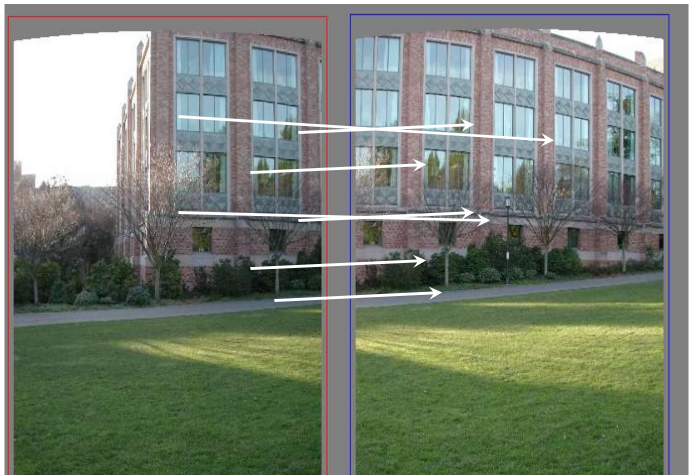


Source: Rick Szeliski

RANSAC example: Translation

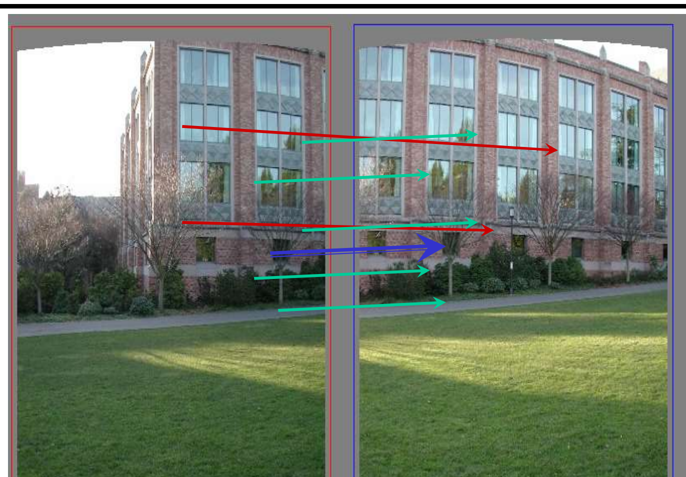


RANSAC example: Translation



Select *one* match, count *inliers*

RANSAC example: Translation

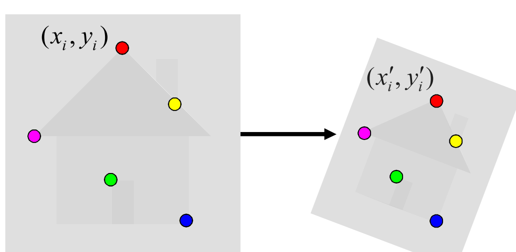


Find “average” translation vector

RANSAC verification

For matching specific scenes/objects, common to use an **affine transformation** for spatial verification

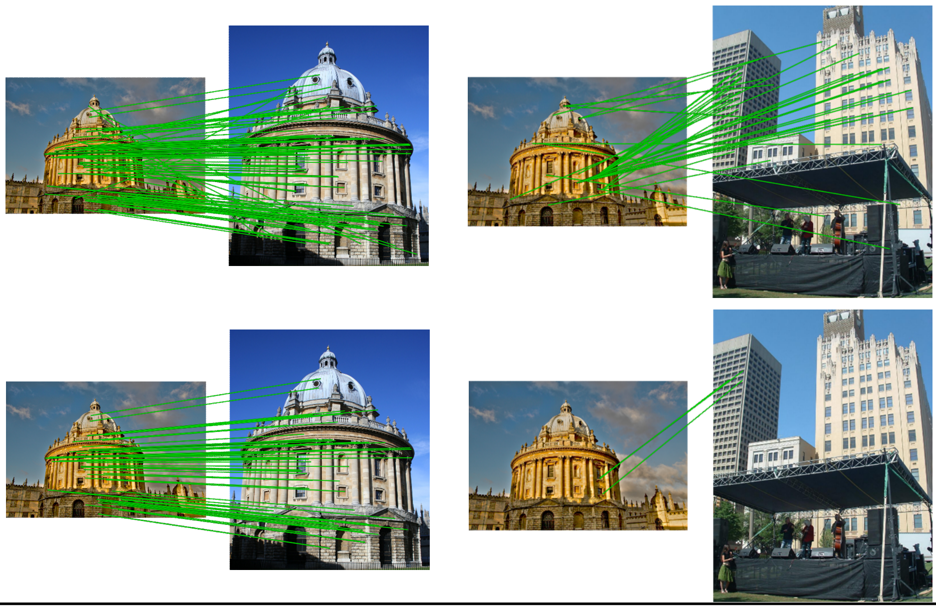
Fitting an affine transformation



Approximates viewpoint changes for roughly planar objects and roughly orthographic cameras.

$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

RANSAC verification



Spatial Verification: two basic strategies

- RANSAC
  - Typically sort by BoW similarity as initial filter
  - Verify by checking support (inliers) for possible affine transformations
    - e.g., "success" if find an affine transformation with > N inlier correspondences
- Generalized Hough Transform
  - Let each matched feature cast a vote on location, scale, orientation of the model object
  - Verify parameters with enough votes

Kristen Grauman

Spatial Verification: two basic strategies

- RANSAC
  - Typically sort by BoW similarity as initial filter
  - Verify by checking support (inliers) for possible affine transformations
    - e.g., “success” if find an affine transformation with > N inlier correspondences
- Generalized Hough Transform
  - Let each matched feature cast a vote on location, scale, orientation of the model object
  - Verify parameters with enough votes

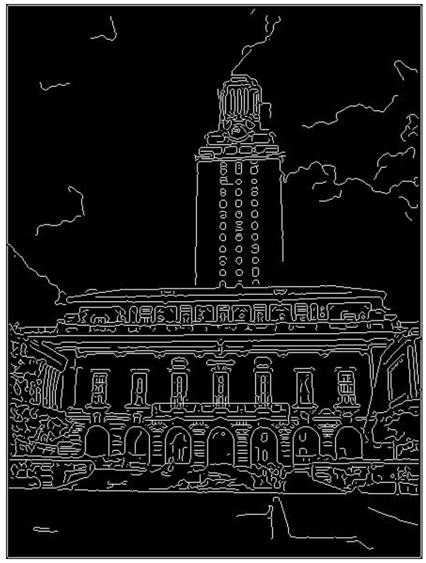
Kristen Grauman

Voting

- It’s not feasible to check all combinations of features by fitting a model to each possible subset.
- **Voting** is a general technique where we let the features *vote for all models that are compatible with it*.
  - Cycle through features, cast votes for model parameters.
  - Look for model parameters that receive a lot of votes.
- Noise & clutter features will cast votes too, *but* typically their votes should be inconsistent with the majority of “good” features.

Kristen Grauman

Difficulty of line fitting

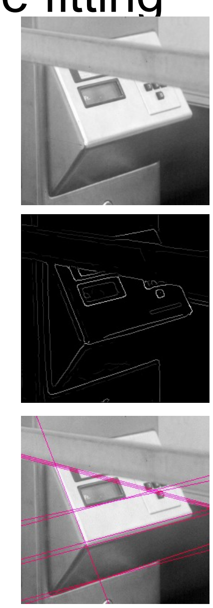


Kristen Grauman

Hough Transform for line fitting

- Given points that belong to a line, what is the line?
- How many lines are there?
- Which points belong to which lines?
- **Hough Transform** is a voting technique that can be used to answer all of these questions.

  Main idea:
  1. Record vote for each possible line on which each edge point lies.
  2. Look for lines that get many votes.



Kristen Grauman

Finding lines in an image: Hough space

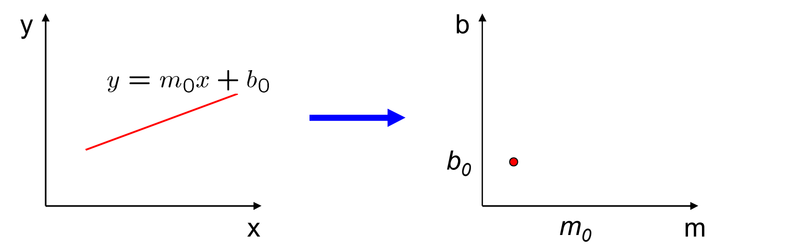


image space

Hough (parameter) space

Connection between image (x,y) and Hough (m,b) spaces

- A line in the image corresponds to a point in Hough space
- To go from image space to Hough space:
  - given a set of points (x,y), find all (m,b) such that y = mx + b

Slide credit: Steve Seitz

Finding lines in an image: Hough space

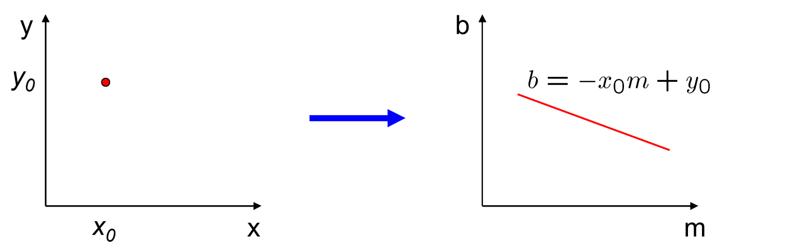


image space

Hough (parameter) space

Connection between image (x,y) and Hough (m,b) spaces

- A line in the image corresponds to a point in Hough space
- To go from image space to Hough space:
  - given a set of points (x,y), find all (m,b) such that y = mx + b
- What does a point (x_0, y_0) in the image space map to?
  - Answer: the solutions of $b = -x_0 m + y_0$
  - this is a line in Hough space

Slide credit: Steve Seitz

Finding lines in an image: Hough space

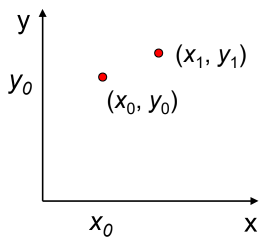
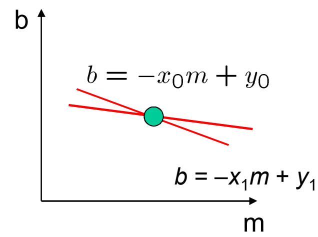


image space



Hough (parameter) space

What are the line parameters for the line that contains both (x_0, y_0) and (x_1, y_1)?

- It is the intersection of the lines $b = -x_0 m + y_0$ and $b = -x_1 m + y_1$

Finding lines in an image: Hough algorithm

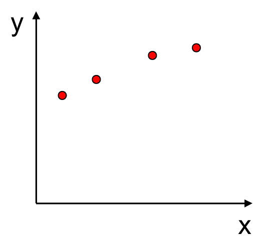
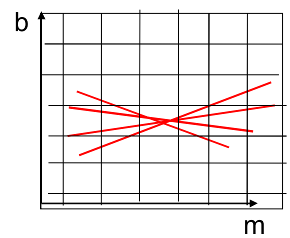


image space



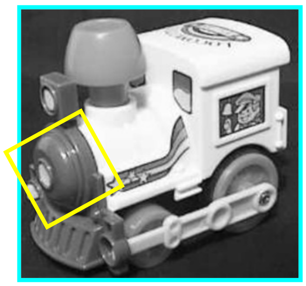
Hough (parameter) space

How can we use this to find the most likely parameters (m,b) for the most prominent line in the image space?

- Let each edge point in image space *vote* for a set of possible parameters in Hough space
- Accumulate votes in discrete set of bins; parameters with the most votes indicate line in image space.

Voting: Generalized Hough Transform

- If we use scale, rotation, and translation invariant local features, then each feature match gives an alignment hypothesis (for scale, translation, and orientation of model in image).



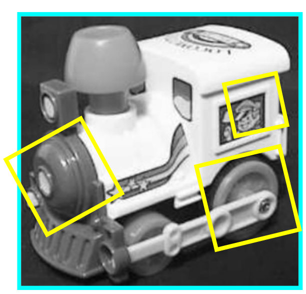
Model

Novel image

Adapted from Lana Lazebnik

Voting: Generalized Hough Transform

- A hypothesis generated by a single match may be unreliable,
- So let each match **vote** for a hypothesis in Hough space



Model

Novel image

Gen Hough Transform details (Lowe's system)

- **Training phase:** For each model feature, record 2D location, scale, and orientation of model (relative to normalized feature frame)
- **Test phase:** Let each match btwn a test SIFT feature and a model feature vote in a 4D Hough space
  - Use broad bin sizes of 30 degrees for orientation, a factor of 2 for scale, and 0.25 times image size for location
  - Vote for two closest bins in each dimension
- Find all bins with at least three votes and perform geometric verification
  - Estimate least squares *affine* transformation
  - Search for additional features that agree with the alignment

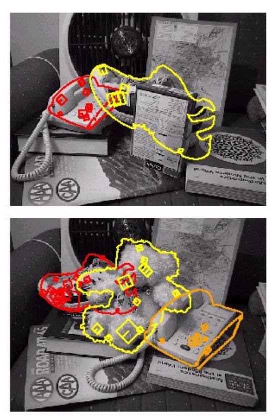
David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), pp. 91-110, 2004.

Slide credit: Lana Lazebnik

Example result

Background subtract for model boundaries

Objects recognized,



Recognition in spite of occlusion

[Lowe]

Gen Hough vs RANSAC

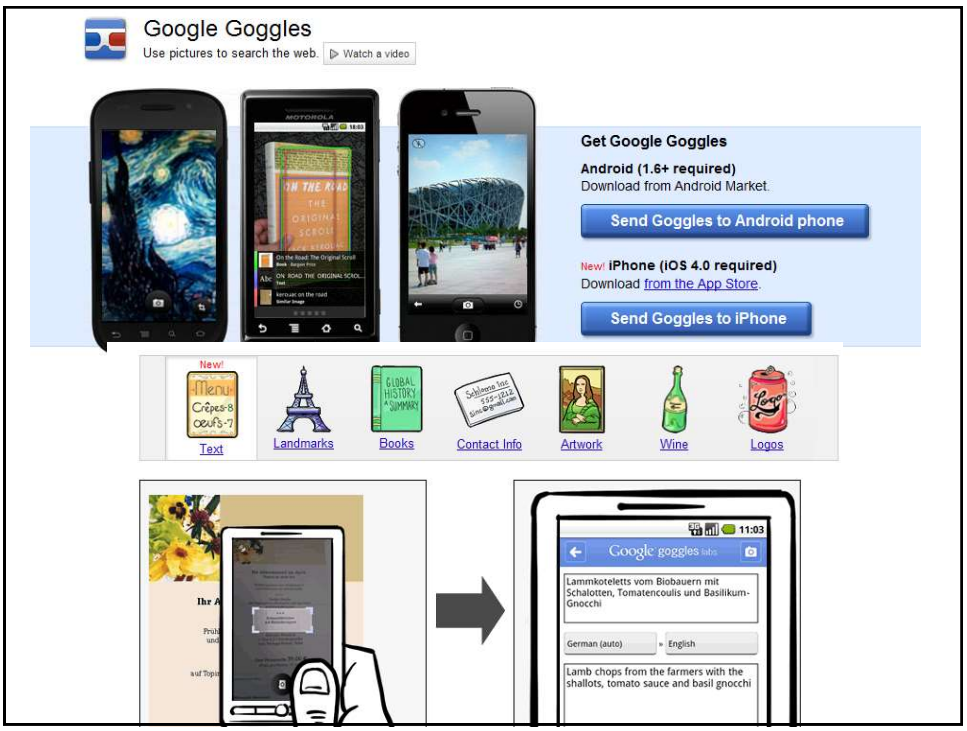
GHT

- Single correspondence -> vote for all consistent parameters
- Represents uncertainty in the model parameter space
- Linear complexity in number of correspondences and number of voting cells; beyond 4D vote space impractical
- Can handle high outlier ratio

RANSAC

- Minimal subset of correspondences to estimate model -> count inliers
- Represents uncertainty in image space
- Must search all data points to check for inliers each iteration
- Scales better to high-d parameter spaces

Kristen Grauman



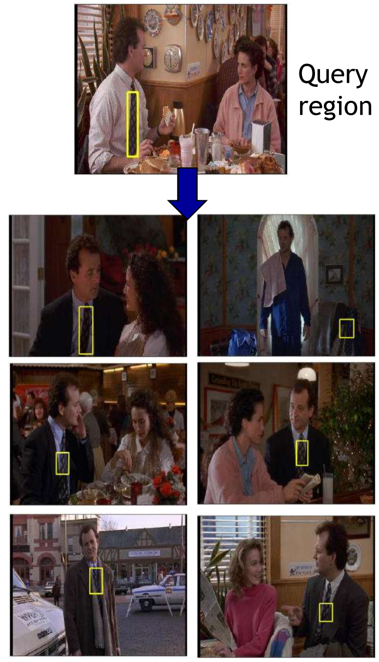
Visual Object Recognition Tutorial

Video Google System

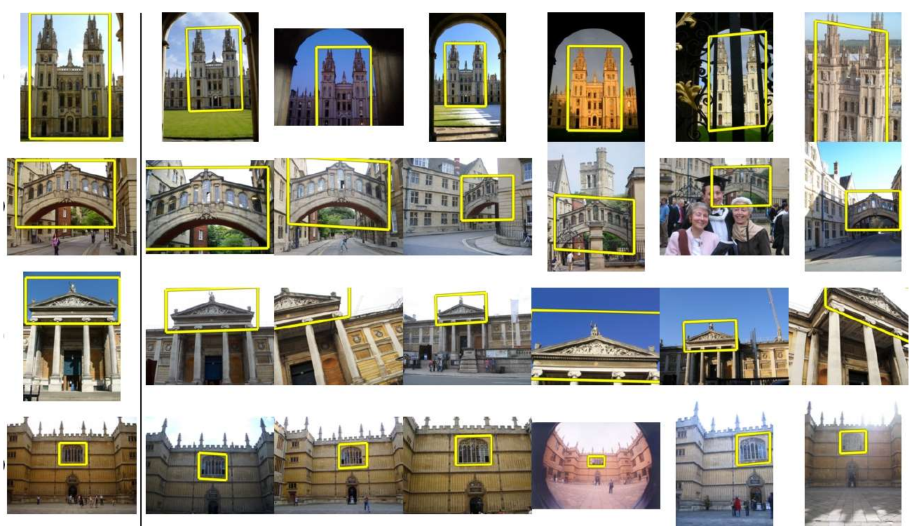
1. Collect all words within query region
2. Inverted file index to find relevant frames
3. Compare word counts
4. Spatial verification

Sivic & Zisserman, ICCV 2003

- **Demo online at :** http://www.robots.ox.ac.uk/~vgg/research/vgoogle/index.html



Object retrieval with large vocabularies and fast spatial matching, Philbin et al., CVPR 2007



Query   Results from 5k Flickr images (demo available for 100k set)

[Philbin CVPR'07]

World-scale mining of objects and events from community photo collections, Quack et al., CIVR 2008

Auto-annotate by connecting to content on Wikipedia!

Example Applications

Mobile tourist guide
- Self-localization
- Object/building recognition
- Photo/video augmentation

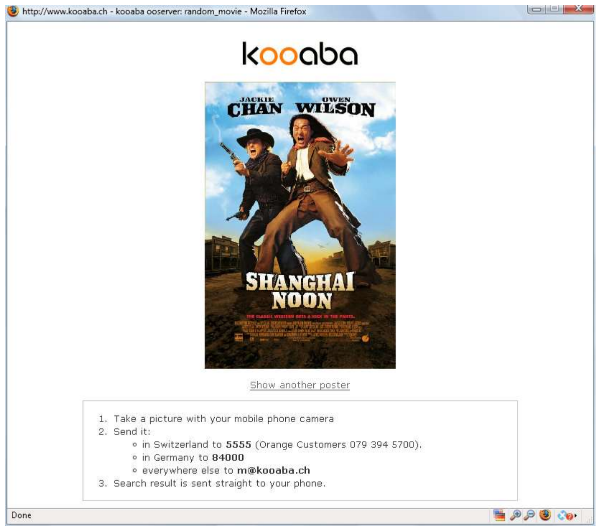
[Quack, Leibe, Van Gool, CIVR'08]

Visual Object Recognition Tutorial

Web Demo: Movie Poster Recognition

50'000 movie posters indexed

Query-by-image from mobile phone available in Switzer-land



http://www.kooaba.com/en/products_engine.html#

Recognition via feature matching+spatial verification

Pros:

- Effective when we are able to find reliable features within clutter
- Great results for matching specific instances

Cons:

- Scaling with number of models
- Spatial verification as post-processing – not seamless, expensive for large-scale problems
- Not suited for category recognition.

Kristen Grauman

Summary: instance recognition

- **Matching local invariant features**
  - Useful not only to provide matches for multi-view geometry, but also to find objects and scenes.
- **Bag of words** representation: quantize feature space to make discrete set of visual words
  - Summarize image by distribution of words
  - Index individual words
- **Inverted index**: pre-compute index to enable faster search at query time
- **[today] Recognition of instances via alignment:** matching local features followed by spatial verification
  - Robust fitting : RANSAC, GHT

Kristen Grauman

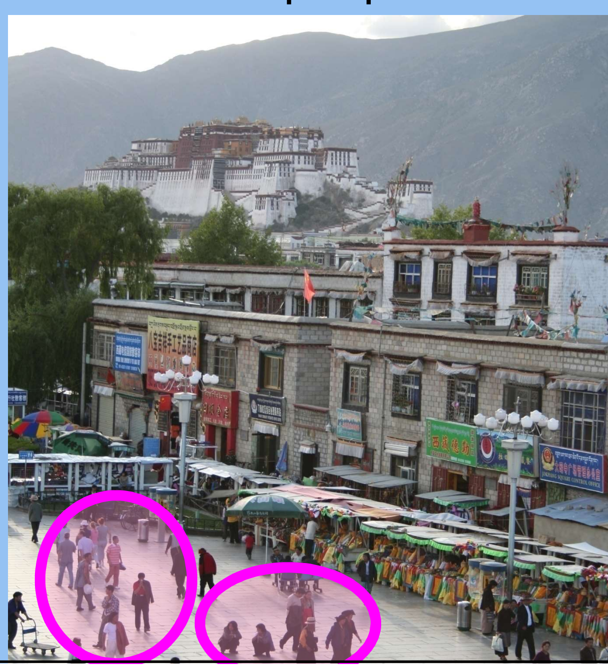
Rest of today

- Intro to categorization problem
- Object categorization as discriminative classification
  a) Boosting + fast face detection example
  b) Nearest neighbors + scene recognition example
  c) Support vector machines + pedestrian detection example
     i. Pyramid match kernels, spatial pyramid match
  d) Convolutional neural networks + ImageNet example
- Some new representations along the way
  - Rectangular filters
  - GIST
  - HOG

What does recognition involve?

Slide credit:
Fei-Fei Li

Detection: are there people?



Slide credit:
Fei-Fei Li

Activity: What are they doing?

Slide credit: Fei-Fei Li

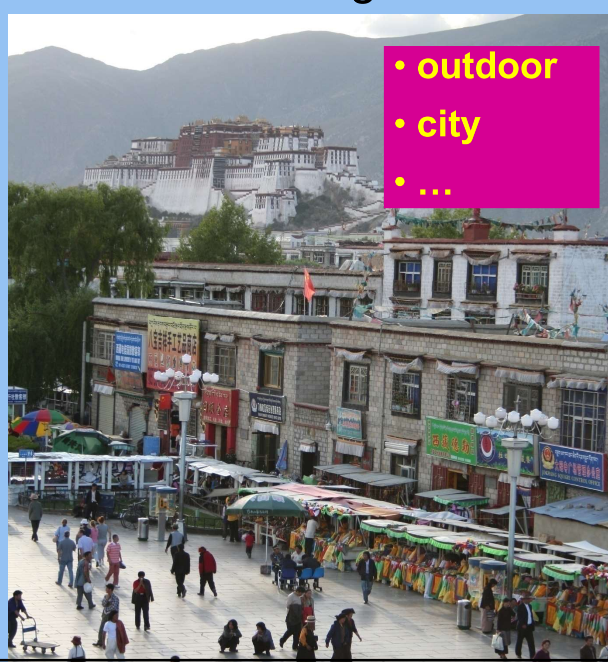
Object categorization

Slide credit: Fei-Fei Li

Instance recognition

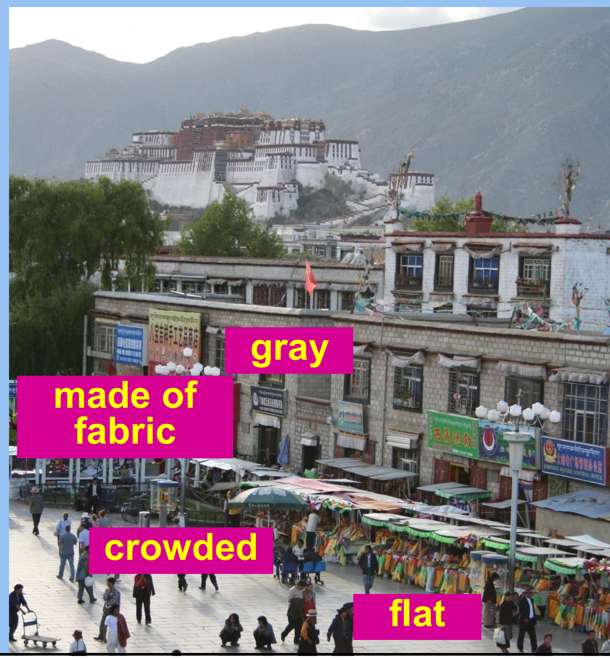
Slide credit:
Fei-Fei Li

Scene and context categorization



Slide credit:
Fei-Fei Li

Attribute recognition



Slide credit:
Fei-Fei Li

Object Categorization

- Task Description
  - “Given a small number of training images of a category, recognize a-priori unknown instances of that category and assign the correct category label.”
- Which categories are feasible visually?

Visual Object Recognition Tutorial

K. Grauman, B. Leibe

Visual Object Recognition Tutorial

Visual Object Categories

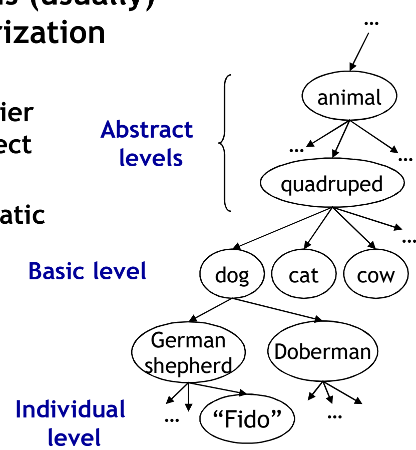
- Basic Level Categories in human categorization
  [Rosch 76, Lakoff 87]
  - The highest level at which category members have similar perceived shape
  - The highest level at which a single mental image reflects the entire category
  - The level at which human subjects are usually fastest at identifying category members
  - The first level named and understood by children
  - The highest level at which a person uses similar motor actions for interaction with category members

K. Grauman, B. Leibe

Visual Object Recognition Tutorial

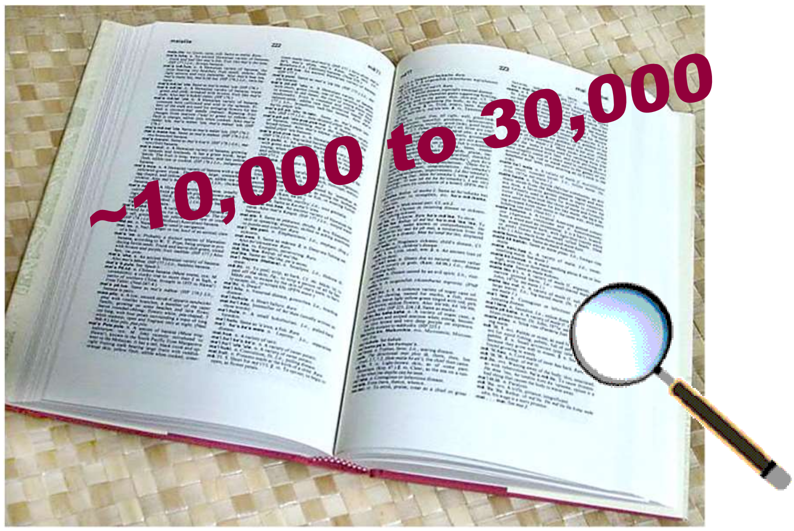
Visual Object Categories

- Basic-level categories in humans seem to be defined predominantly visually.
- There is evidence that humans (usually) start with basic-level categorization *before* doing identification.
  - ⇒ Basic-level categorization is easier and faster for humans than object identification!
  - ⇒ How does this transfer to automatic classification algorithms?



K. Grauman, B. Leibe

How many object categories are there?



Source: Fei-Fei Li, Rob Fergus, Antonio Torralba.

Biederman 1987

Visual Object Recognition Tutorial

Other Types of Categories

- Functional Categories
  - e.g. chairs = *"something you can sit on"*

K. Grauman, B. Leibe

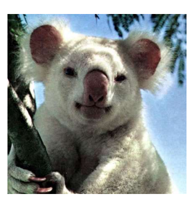
Challenges: robustness

Illumination

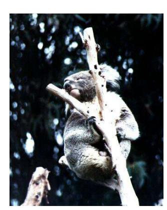
Object pose

Clutter

Occlusions



Intra-class appearance



Viewpoint

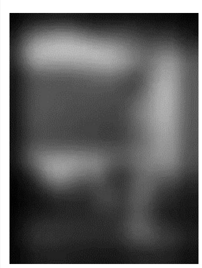
Challenges:
context and human experience

Context cues

Challenges:
context and human experience

Context cues

Function



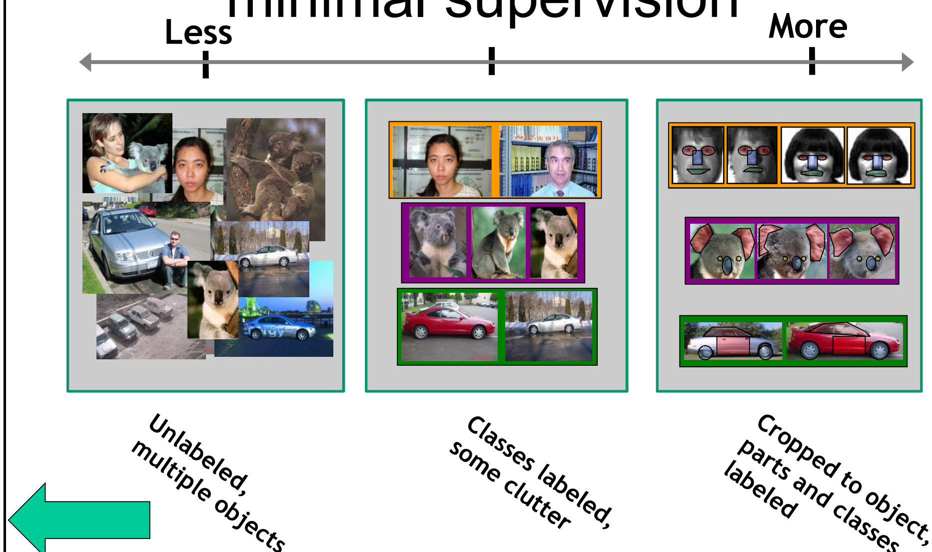
Dynamics

Video credit: J. Davis

Challenges: complexity

- Millions of pixels in an image
- 30,000 human recognizable object categories
- 30+ degrees of freedom in the pose of articulated objects (humans)
- Billions of images online
- 300 hours of new video on YouTube per minute
- …
- About half of the cerebral cortex in primates is devoted to processing visual information [Felleman and van Essen 1991]

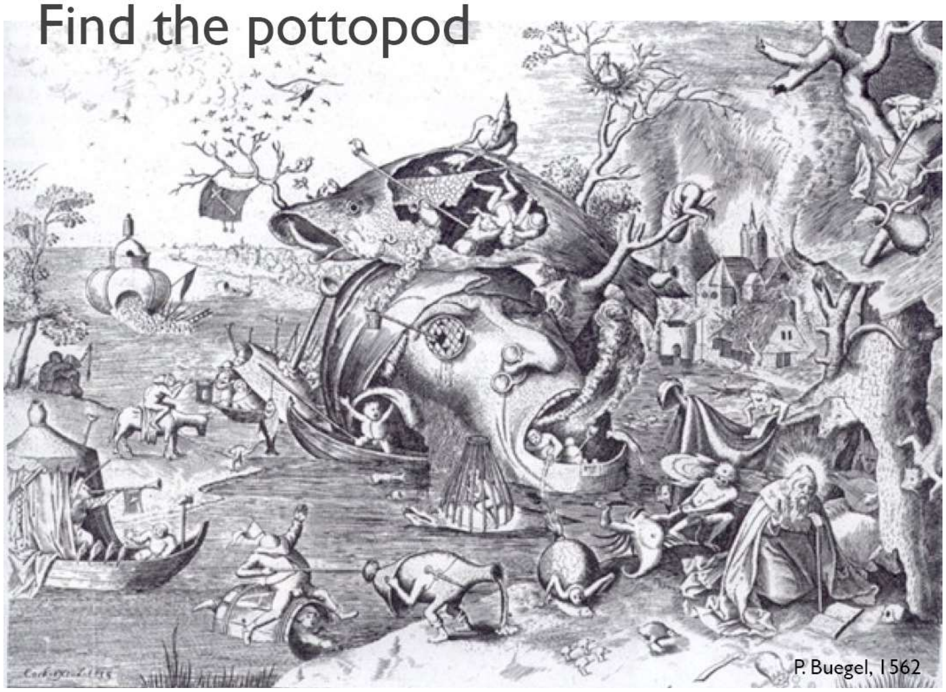
Challenges: learning with minimal supervision



This is a pottopod

Slide from Pietro Perona, 2004 Object Recognition workshop

Find the pottopod



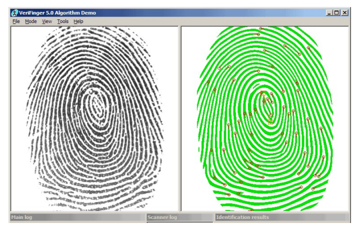
Slide from Pietro Perona, 2004 Object Recognition workshop

What kinds of things work best today?

Reading license plates, zip codes, checks

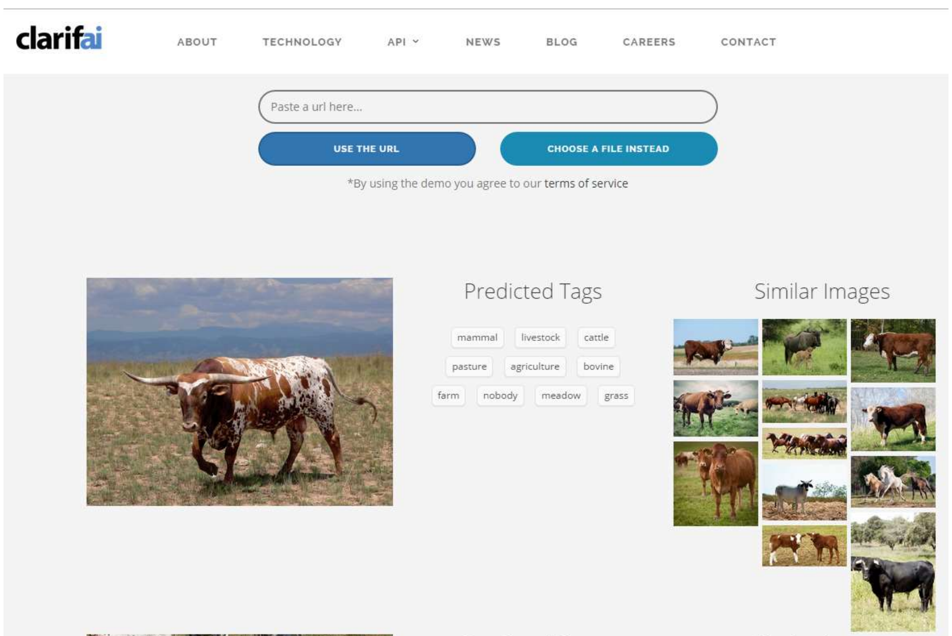
Frontal face detection

Recognizing flat, textured objects (like books, CD covers, posters)



Fingerprint recognition

What kinds of things work best today?



Evolution of methods

- Hand-crafted models
- 3D geometry
- Hypothesize and align

- Hand-crafted features
- Learned models
- Data-driven

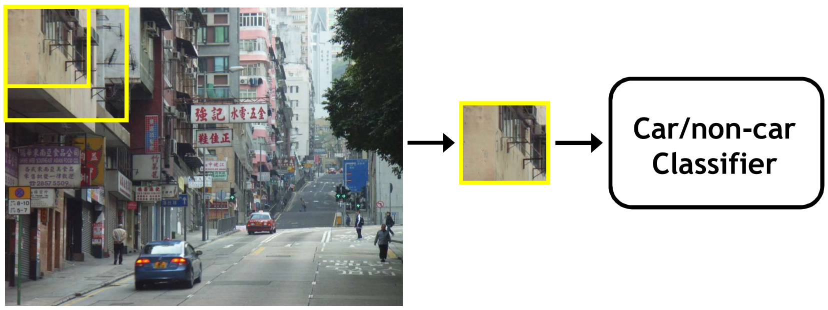
- "End-to-end" learning of features and models*,**

* Labeled data availability
** Architecture design decisions, parameters.

Generic category recognition: basic framework

- Build/train object model
  - (Choose a representation)
  - Learn or fit parameters of model / classifier
- Generate candidates in new image
- Score the candidates

Window-based models
Generating and scoring candidates

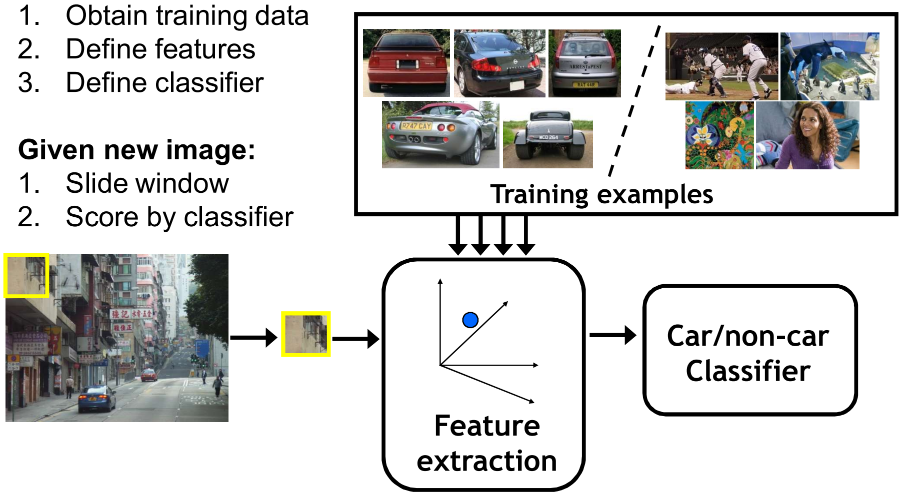


Kristen Grauman

Window-based object detection

Training:
1. Obtain training data
2. Define features
3. Define classifier

Given new image:
1. Slide window
2. Score by classifier



Kristen Grauman

Object recognition as classification

- What classifier?
  - Factors in choosing:
    - Generative or discriminative model?
    - Data resources – how much training data?
    - How is the labeled data prepared?
    - Training time allowance
    - Test time requirements – real-time?
    - Fit with the representation

Kristen Grauman

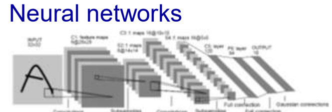
Discriminative classifies

Nearest neighbor

10^6 examples

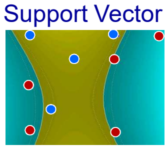
Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005, Hays 2008,
Torralba 2008,……

Neural networks



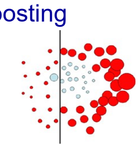
LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998,
Krizhevsky 2012…
…

Support Vector Machines



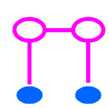
Guyon, Vapnik
Heisele, Serre, Poggio,
2001, Lazebnik 2006…

Boosting



Viola, Jones 2001,
Torralba et al. 2004,
Opelt et al. 2006,…

Conditional Random Fields



McCallum, Freitag, Pereira
2000; Kumar, Hebert 2003
…

Kristen Grauman

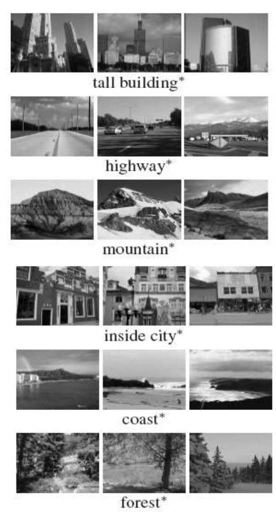
Slide adapted from Antonio Torralba

Object recognition as classification

- What categories are amenable to window-based classification?
  - **Similar to specific object matching,** we expect spatial layout to be roughly preserved.
  - **Unlike specific object matching**, by training classifiers we attempt to capture intra-class variation or determine required discriminative features.

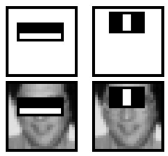
Kristen Grauman

What categories are amenable to window-based reps?



Kristen Grauman

Window-based models: Three landmark case studies

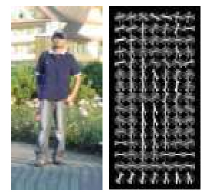


Boosting + face detection

Viola & Jones

NN + scene Gist classification

e.g., Hays & Efros



SVM + person detection

e.g., Dalal & Triggs

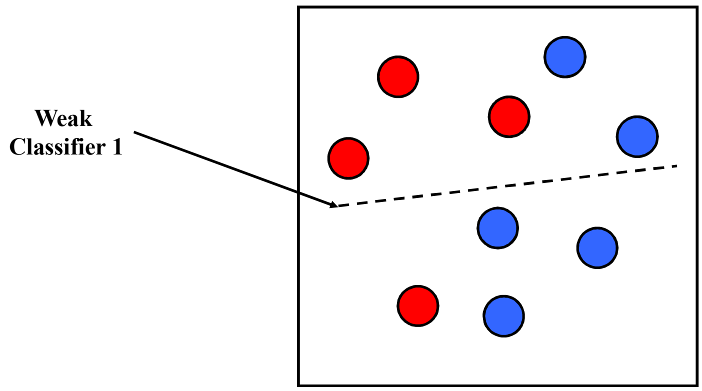
Viola-Jones face detector

Main idea:

- Represent local texture with efficiently computable “rectangular” features within window of interest
- Select discriminative features to be weak classifiers
- Use boosted combination of them as final classifier
- Form a cascade of such classifiers, rejecting clear negatives quickly

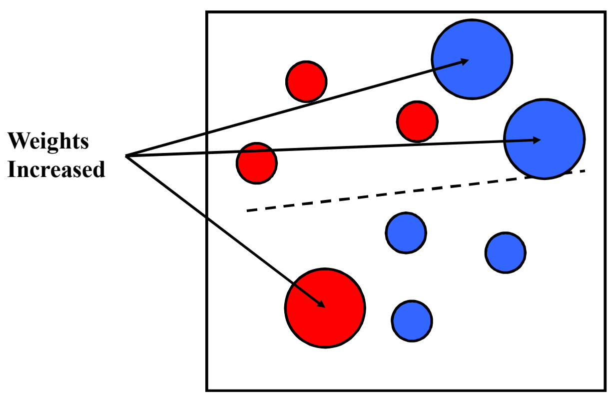
Kristen Grauman

Boosting intuition

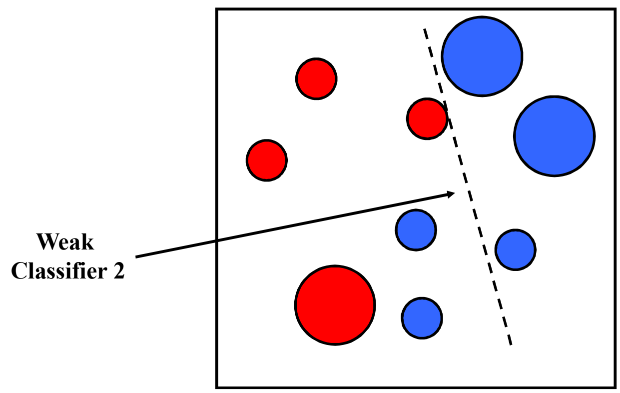


Slide credit: Paul Viola

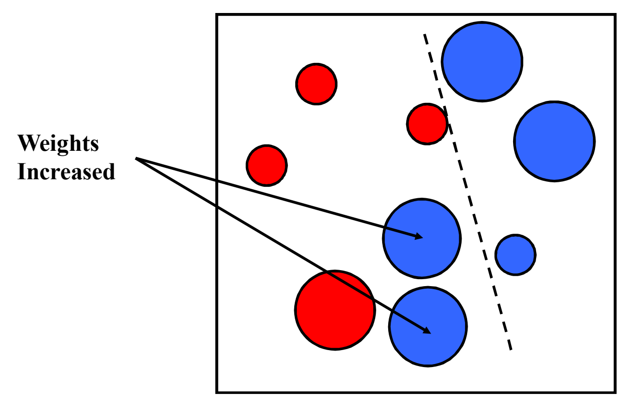
Boosting illustration



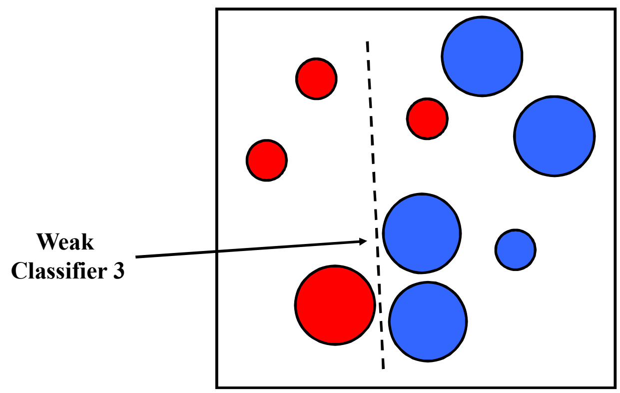
Boosting illustration



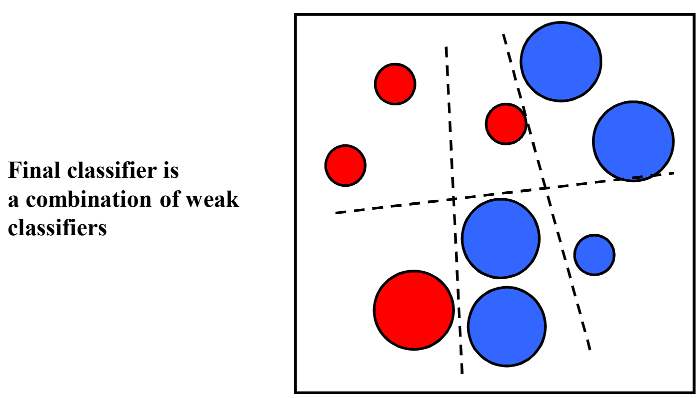
Boosting illustration



Boosting illustration



Boosting illustration



Boosting: training

- Initially, weight each training example equally
- In each boosting round:
  - Find the weak learner that achieves the lowest *weighted* training error
  - Raise weights of training examples misclassified by current weak learner
- Compute final classifier as linear combination of all weak learners (weight of each learner is directly proportional to its accuracy)

- Exact formulas for re-weighting and combining weak learners depend on the particular boosting scheme (e.g., AdaBoost)

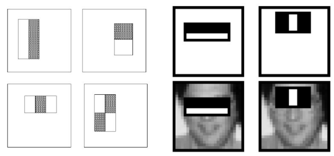
Slide credit: Lana Lazebnik

Boosting: pros and cons

- Advantages of boosting
  - Integrates classification with feature selection
  - Complexity of training is linear in the number of training examples
  - Flexibility in the choice of weak learners, boosting scheme
  - Testing is fast
  - Easy to implement
- Disadvantages
  - Needs many training examples
  - Often found not to work as well as an alternative discriminative classifier, support vector machine (SVM), or CNNs
    - especially for many-class problems

Slide credit: Lana Lazebnik

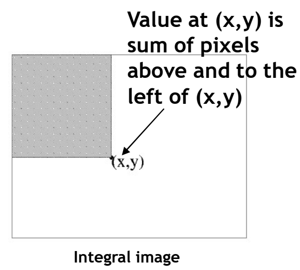
Viola-Jones detector: features



"Rectangular" filters

Feature output is difference between adjacent regions

Efficiently computable with integral image: any sum can be computed in constant time.

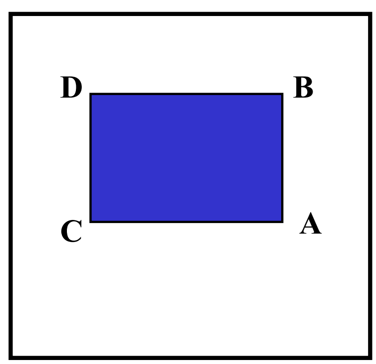


Kristen Grauman

Computing sum within a rectangle

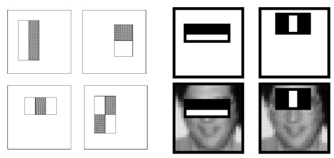
- Let A,B,C,D be the values of the integral image at the corners of a rectangle
- Then the sum of original image values within the rectangle can be computed as:

  sum = A – B – C + D
- Only 3 additions are required for any size of rectangle!



Lana Lazebnik

Viola-Jones detector: features

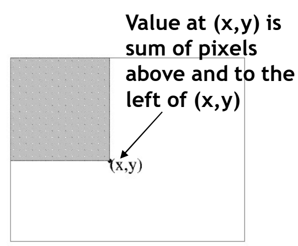


"Rectangular" filters

Feature output is difference between adjacent regions

Efficiently computable with integral image: any sum can be computed in constant time

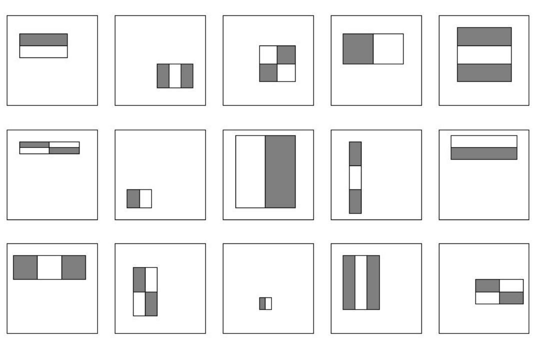
Avoid scaling images → scale features directly for same cost



Integral image

Kristen Grauman

Viola-Jones detector: features



Considering all possible filter parameters: position, scale, and type:

180,000+ possible features associated with each 24 x 24 window

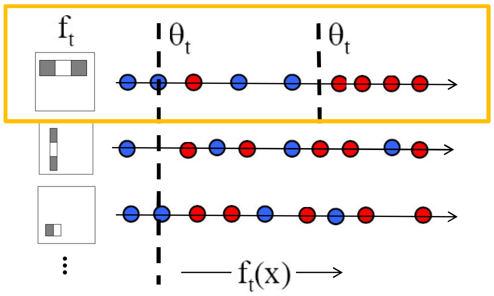
Which subset of these features should we use to determine if a window has a face?

Use AdaBoost both to select the informative features and to form the classifier

Kristen Grauman

Viola-Jones detector: AdaBoost

- Want to select the single rectangle feature and threshold that best separates positive (faces) and negative (non-faces) training examples, in terms of *weighted* error.



Outputs of a possible rectangle feature on faces and non-faces.

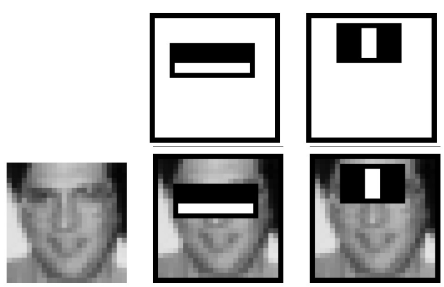
Resulting weak classifier:

$$h_t(x) = \begin{cases} +1 & \text{if } f_t(x) > \theta_t \\ -1 & \text{otherwise} \end{cases}$$

For next round, reweight the examples according to errors, choose another filter/threshold combo.

Kristen Grauman

Viola-Jones Face Detector: Results

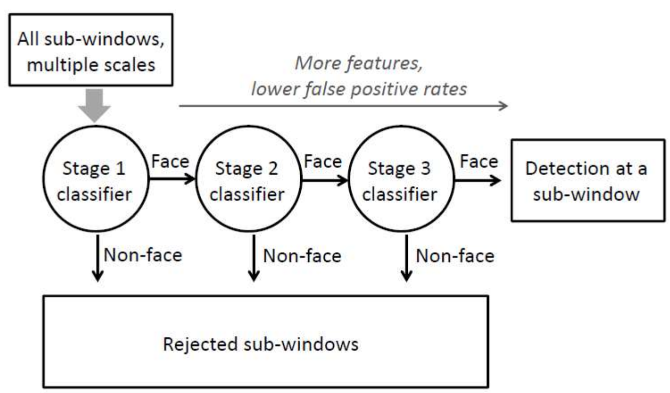


First two features selected

Visual Object Recognition Tutorial

- Even if the filters are fast to compute, each new image has a lot of possible windows to search.
- How to make the detection more efficient?

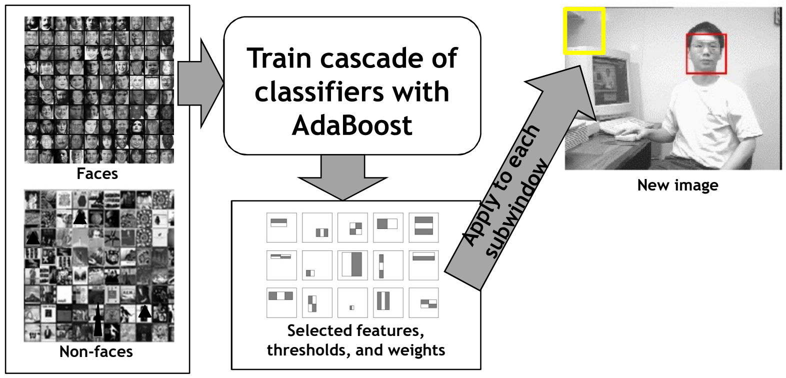
Cascading classifiers for detection



- Form a *cascade* with low false negative rates early on
- Apply less accurate but faster classifiers first to immediately discard windows that clearly appear to be negative

Kristen Grauman

Viola-Jones detector: summary



Train with 5K positives, 350M negatives
Real-time detector using 38 layer cascade
6061 features in all layers

[Implementation available in OpenCV: http://www.intel.com/technology/computing/opencv/]

Kristen Grauman

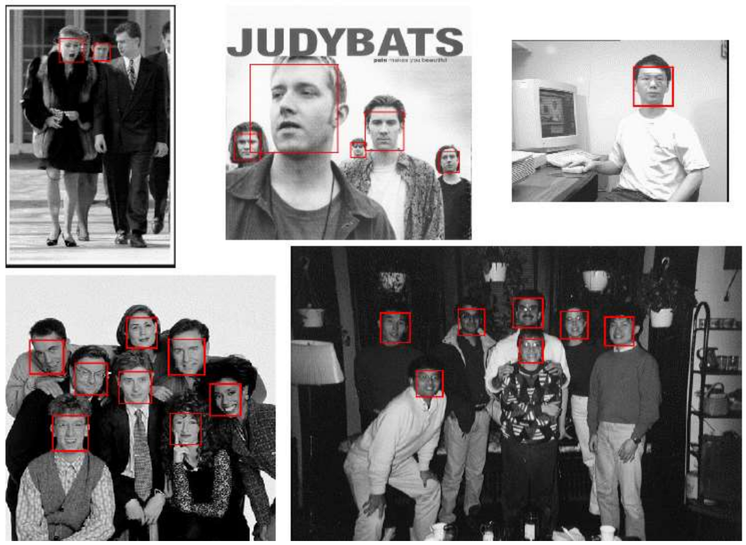
Viola-Jones detector: summary

- A seminal approach to real-time object detection
- Training is slow, but detection is very fast
- Key ideas
  - *Integral images* for fast feature evaluation
  - *Boosting* for feature selection
  - *Attentional cascade* of classifiers for fast rejection of non-face windows

P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features.* CVPR 2001.

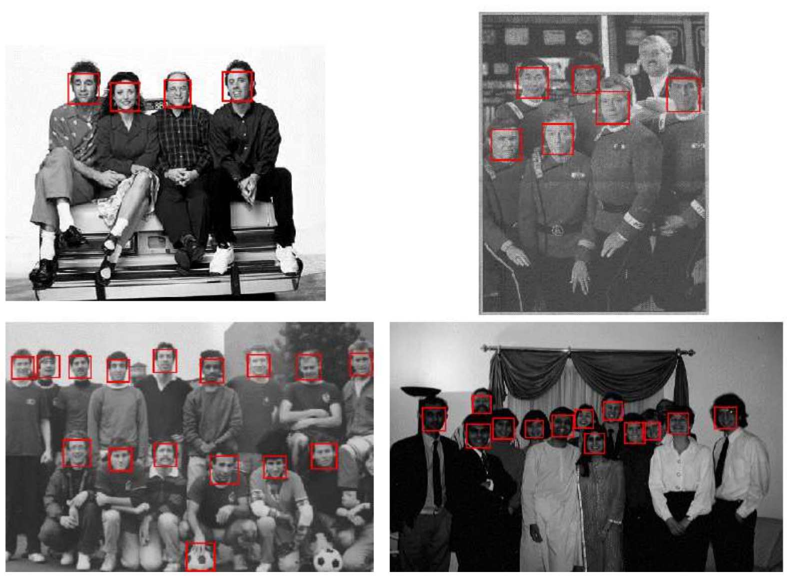
P. Viola and M. Jones. *Robust real-time face detection.* IJCV 57(2), 2004.

Viola-Jones Face Detector: Results



Visual Object Recognition Tutorial

Viola-Jones Face Detector: Results



Visual Object Recognition Tutorial

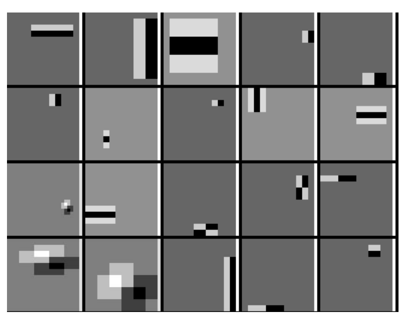
Visual Object Recognition Tutorial

Viola-Jones Face Detector: Results

Visual Object Recognition Tutorial

Detecting profile faces?

Can we use the same detector?

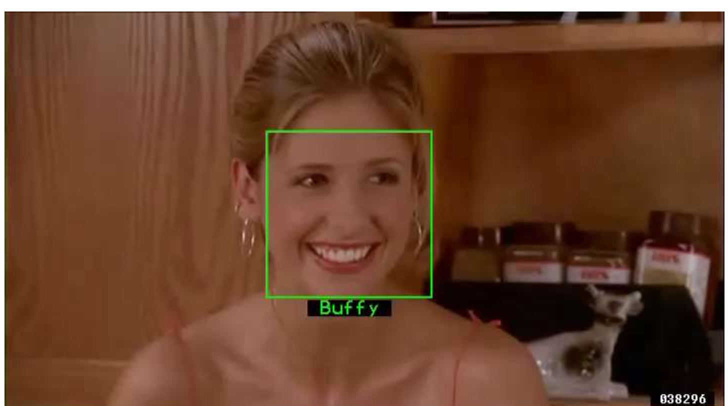


Viola-Jones Face Detector: Results

Visual Object Recognition Tutorial

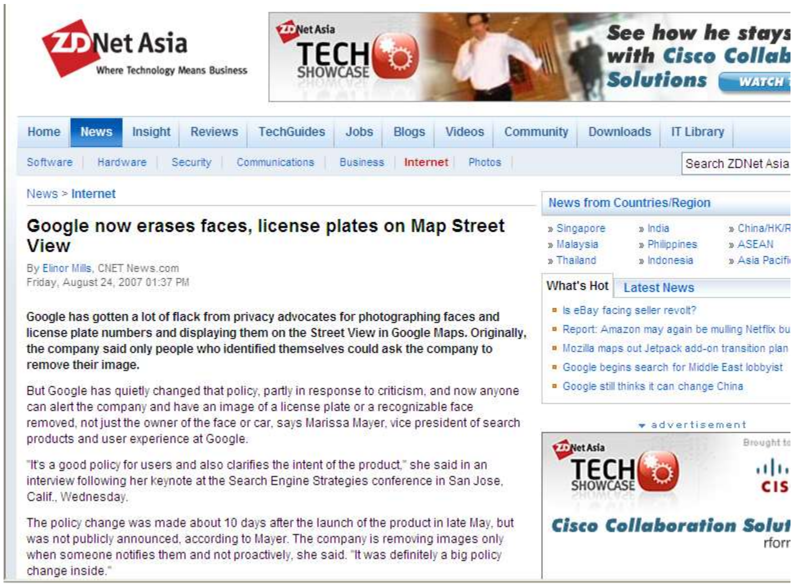
Paul

Example using Viola-Jones detector



Frontal faces detected and then tracked, "character" names inferred with alignment of script and subtitles.

Everingham, M., Sivic, J. and Zisserman, A.
"Hello! My name is... Buffy" - Automatic naming of characters in TV video, BMVC 2006. http://www.robots.ox.ac.uk/~vgg/research/nface/index.html



ZDNet Asia
Where Technology Means Business

Home | News | Insight | Reviews | TechGuides | Jobs | Blogs | Videos | Community | Downloads | IT Library

Software | Hardware | Security | Communications | Business | Internet | Photos

News > Internet

Google now erases faces, license plates on Map Street View

By Elinor Mills, CNET News.com
Friday, August 24, 2007 01:37 PM

Google has gotten a lot of flack from privacy advocates for photographing faces and license plate numbers and displaying them on the Street View in Google Maps. Originally, the company said only people who identified themselves could ask the company to remove their image.

But Google has quietly changed that policy, partly in response to criticism, and now anyone can alert the company and have an image of a license plate or a recognizable face removed, not just the owner of the face or car, says Marissa Mayer, vice president of search products and user experience at Google.

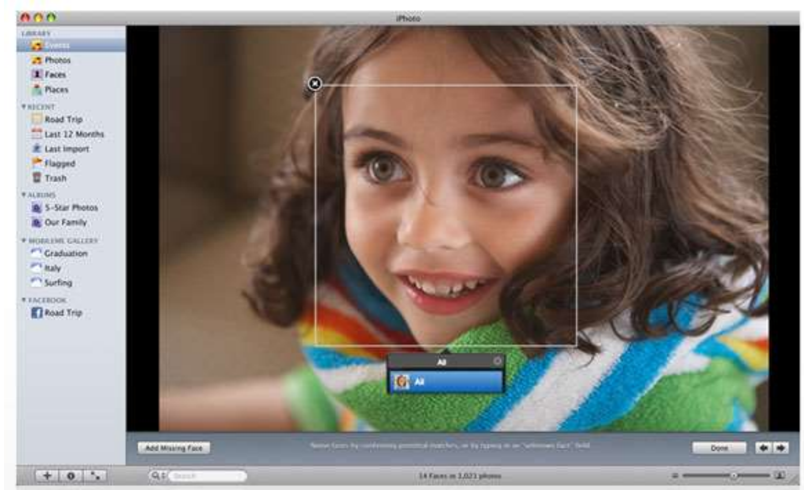
"It's a good policy for users and also clarifies the intent of the product," she said in an interview following her keynote at the Search Engine Strategies conference in San Jose, Calif., Wednesday.

The policy change was made about 10 days after the launch of the product in late May, but was not publicly announced, according to Mayer. The company is removing images only when someone notifies them and not proactively, she said. "It was definitely a big policy change inside."

News from Countries/Region
» Singapore » India » China/HK/R
» Malaysia » Philippines » ASEAN
» Thailand » Indonesia » Asia Pacifi

What's Hot | Latest News
- Is eBay facing seller revolt?
- Report: Amazon may again be mulling Netflix bu
- Mozilla maps out Jetpack add-on transition plan
- Google begins search for Middle East lobbyist
- Google still thinks it can change China

Consumer application: iPhoto



http://www.apple.com/ilife/iphoto/

Slide credit: Lana Lazebnik

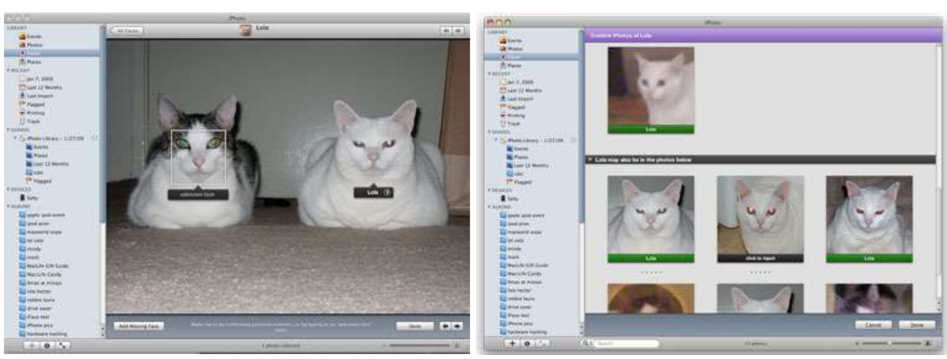
Consumer application: iPhoto

[Things iPhoto thinks are faces]

Slide credit: Lana Lazebnik

Consumer application: iPhoto

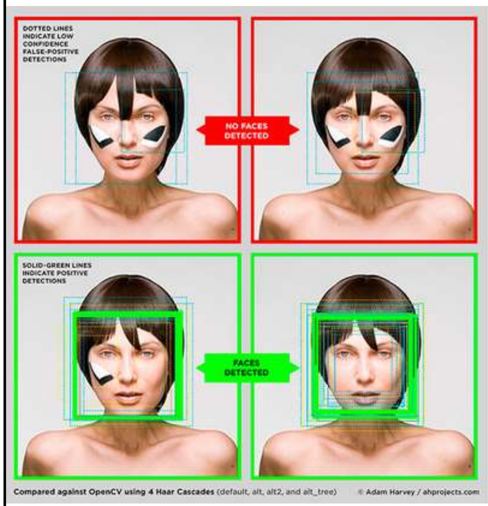
Can be trained to recognize pets!



http://www.maclife.com/article/news/iphotos_faces_recognizes_cats

Slide credit: Lana Lazebnik

Privacy Gift Shop – CV Dazzle



http://www.wired.com/2015/06/facebook-can-recognize-even-dont-show-face/
Wired, June 15, 2015

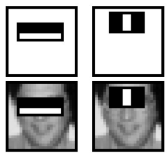
Slide credit: Kristen Grauman

Privacy Visor

http://www.3ders.org/articles/20150812-japan-3d-printed-privacy-visors-will-block-facial-recognition-software.html

Slide credit: Kristen Grauman

Window-based models: Three landmark case studies

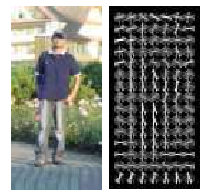


Boosting + face detection

Viola & Jones

NN + scene Gist classification

e.g., Hays & Efros



SVM + person detection

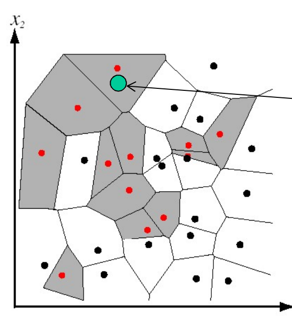
e.g., Dalal & Triggs

Slide credit: Kristen Grauman

Nearest Neighbor classification

- Assign label of nearest training data point to each test data point

Black = negative
Red = positive



from Duda *et al.*

Novel test example

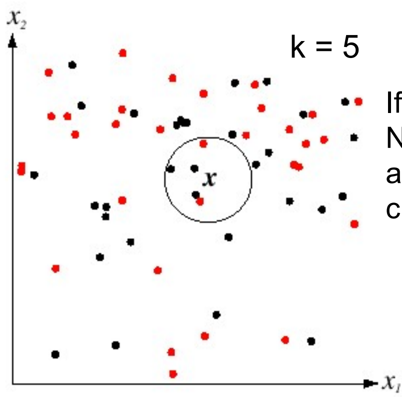
Closest to a positive example from the training set, so classify it as positive.

Voronoi partitioning of feature space for 2-category 2D data

K-Nearest Neighbors classification

- For a new point, find the k closest points from training data
- Labels of the k points "vote" to classify

Black = negative
Red = positive

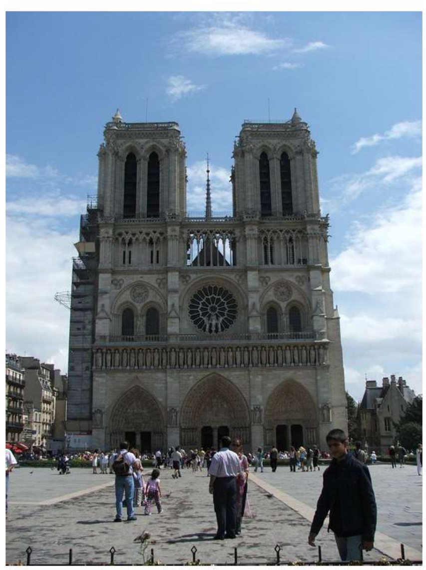


k = 5

If query lands here, the 5 NN consist of 3 negatives and 2 positives, so we classify it as negative.

Source: D. Lowe

Where in the World?



[Hays and Efros. **im2gps**: Estimating Geographic Information from a Single Image. CVPR 2008.]

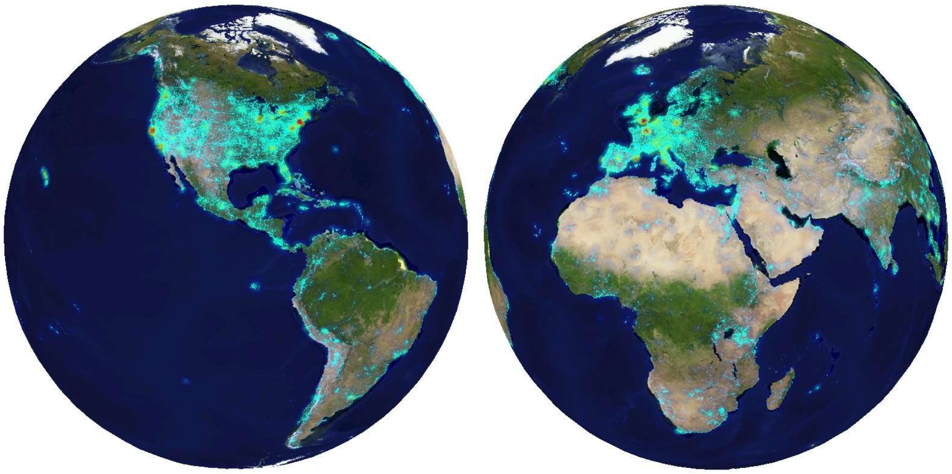
Where in the World?

Slide credit: James Hays

Where in the World?

Slide credit: James Hays

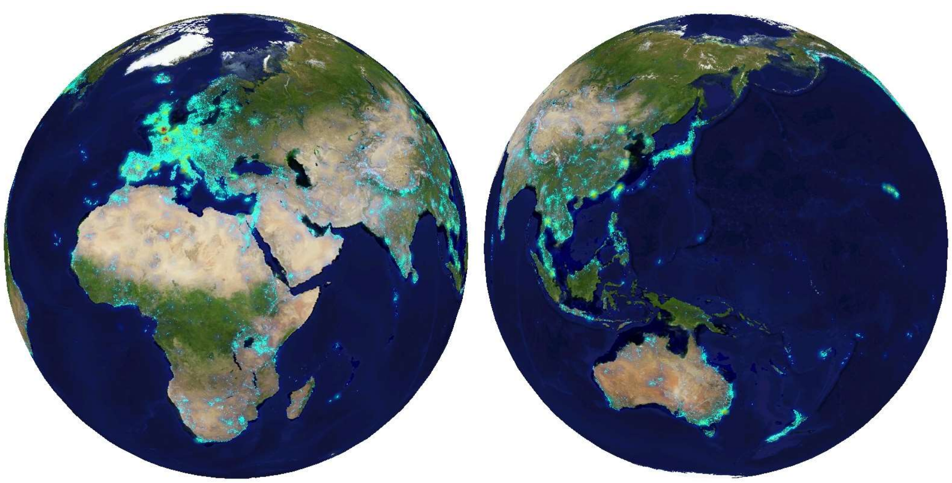
6+ million geotagged photos
by 109,788 photographers



Annotated by Flickr users

Slide credit: James Hays

6+ million geotagged photos
by 109,788 photographers



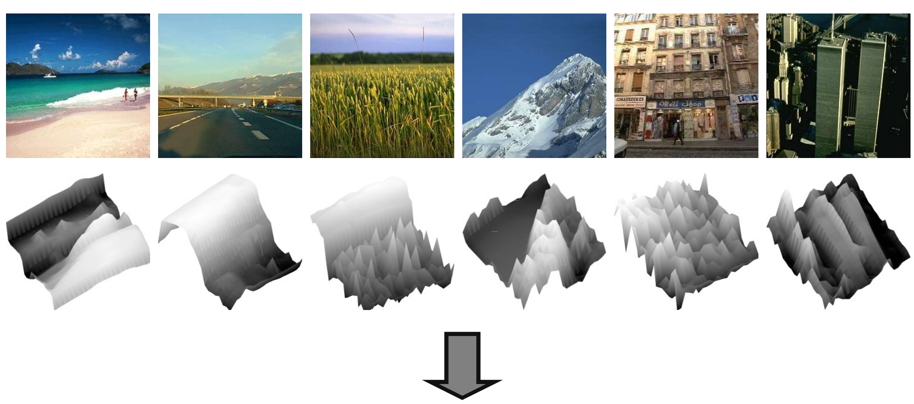
Annotated by Flickr users

Slide credit: James Hays

Which scene properties are relevant?

Spatial Envelope Theory of Scene Representation

Oliva & Torralba (2001)

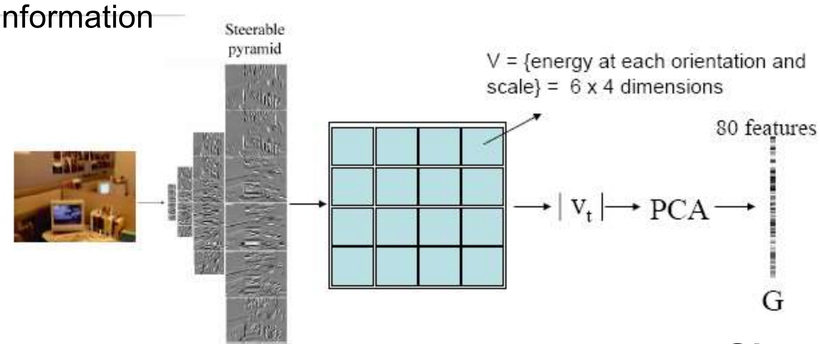


A scene is a single surface that can be represented by global (statistical) descriptors

Slide Credit: Aude Olivia

Global texture: capturing the "Gist" of the scene

Capture global image properties while keeping some spatial information



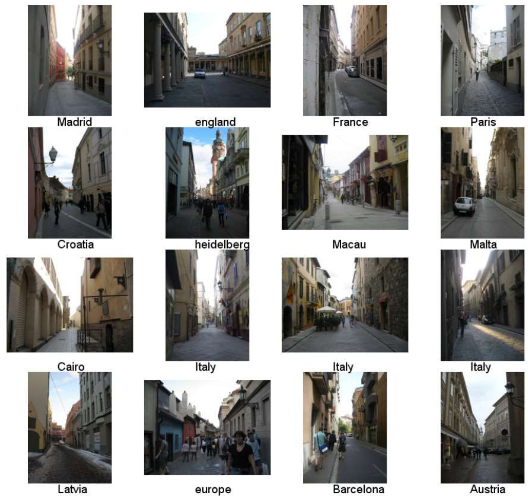
Gist descriptor

Oliva & Torralba IJCV 2001, Torralba et al. CVPR 2003

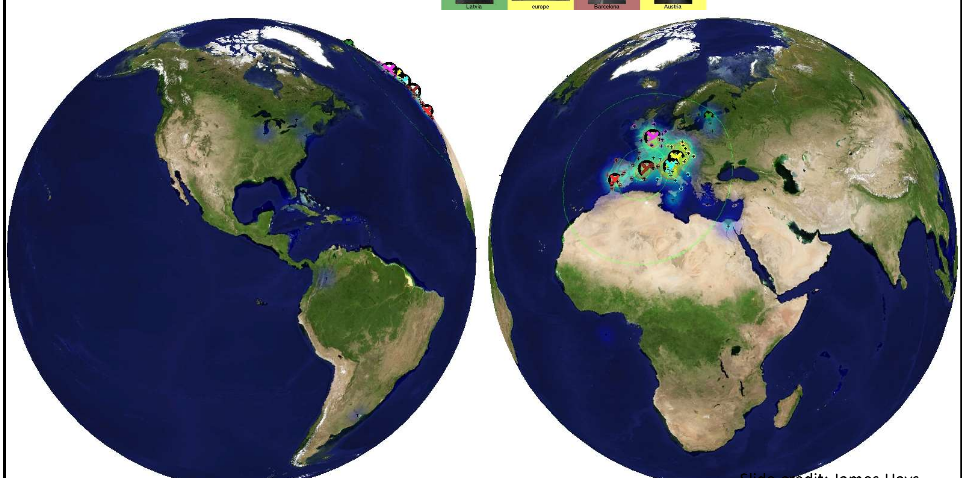
Which scene properties are relevant?

- **Gist scene descriptor**
- **Color Histograms** - L*A*B* 4x14x14 histograms
- **Texton Histograms** – 512 entry, filter bank based
- **Line Features** – Histograms of straight line stats

Scene Matches

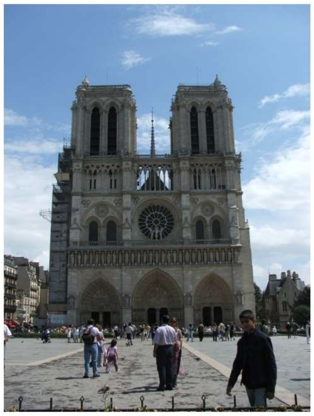


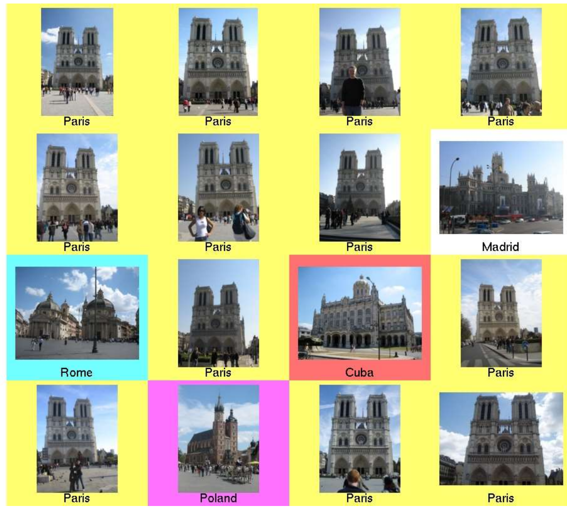
[Hays and Efros. **im2gps**: Estimating Geographic Information from a Single Image. CVPR 2008.] Slide credit: James Hays



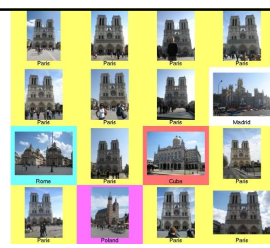
Slide credit: James Hays

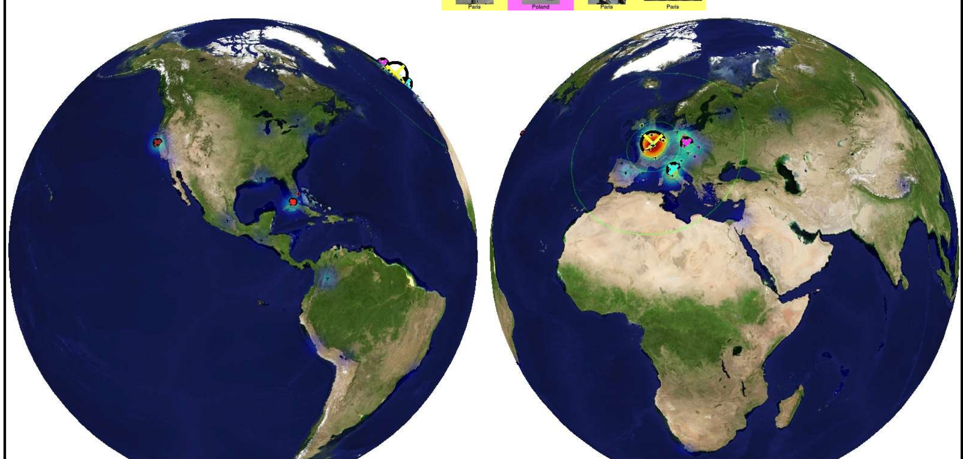
Scene Matches





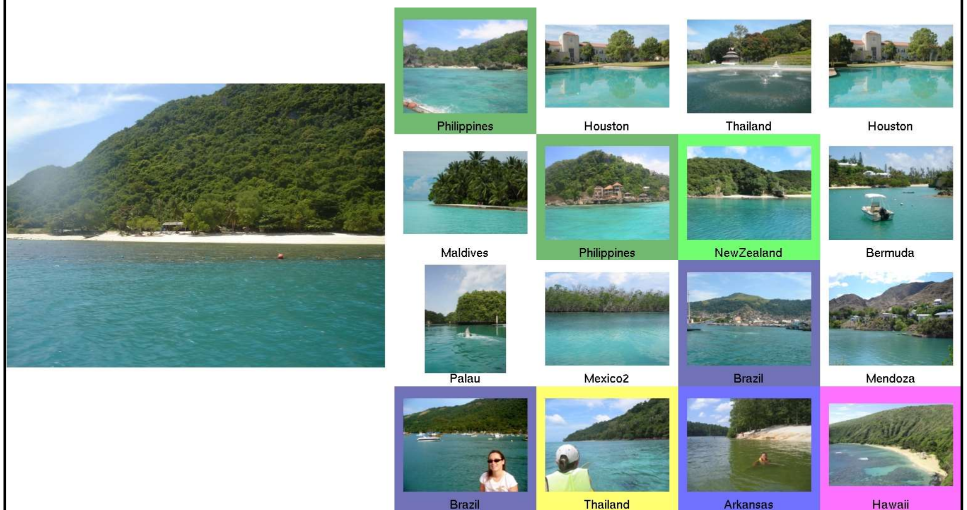
[Hays and Efros. **im2gps**: Estimating Geographic Information from a Single Image. CVPR 2008.] Slide credit: James Hays





[Hays and Efros. **im2gps**: Estimating Geographic Information from a Single Image. CVPR 2008.] Slide credit: James Hays

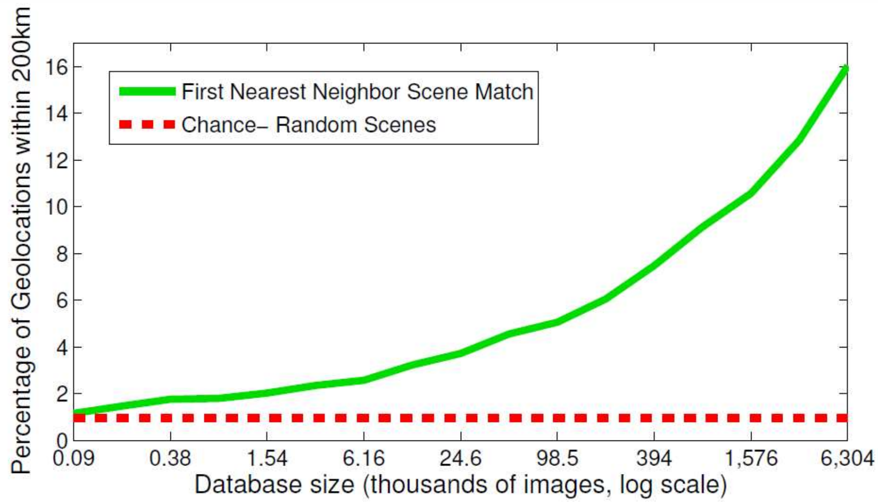
Scene Matches



[Hays and Efros. **im2gps**: Estimating Geographic Information from a Single Image. CVPR 2008.] Slide credit: James Hays

[Hays and Efros. **im2gps**: Estimating Geographic Information from a Single Image. CVPR 2008.] Slide credit: James Hays

The Importance of Data



[Hays and Efros. **im2gps**: Estimating Geographic Information from a Single Image. CVPR 2008.]

Slide credit: James Hays

Nearest neighbors: pros and cons

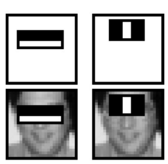
- **Pros**:
  - Simple to implement
  - Flexible to feature / distance choices
  - Naturally handles multi-class cases
  - Can do well in practice with enough representative data
- **Cons:**
  - Large search problem to find nearest neighbors
  - Storage of data
  - Must know we have a meaningful distance function

Kristen Grauman

Today

- Intro to categorization problem
- Object categorization as discriminative classification
  - Boosting + fast face detection example
  - Nearest neighbors + scene recognition example
  - Support vector machines + pedestrian detection example
    - Pyramid match kernels, spatial pyramid match
  - Convolutional neural networks + ImageNet example
- Some new representations along the way
  - Rectangular filters
  - GIST
  - HOG

Window-based models: Three landmark case studies

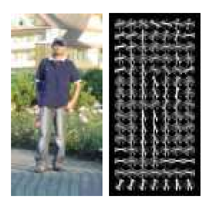


Boosting + face detection

Viola & Jones

NN + scene Gist classification

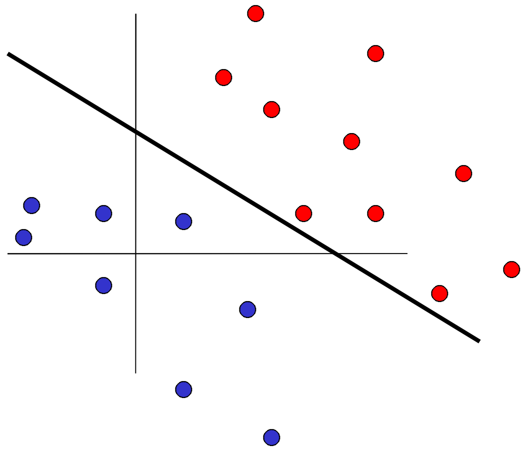
e.g., Hays & Efros



SVM + person detection

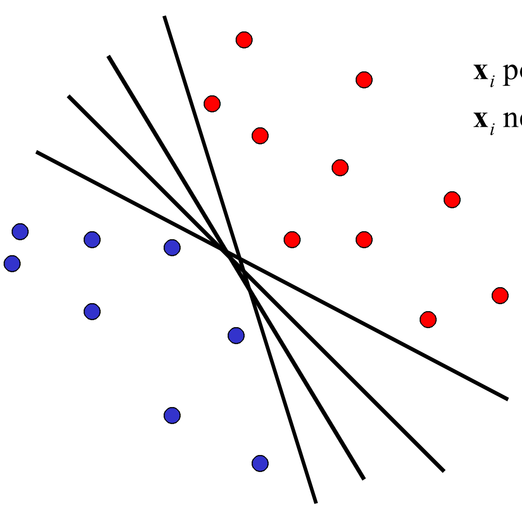
e.g., Dalal & Triggs

Linear classifiers



Linear classifiers

- Find linear function to separate positive and negative examples

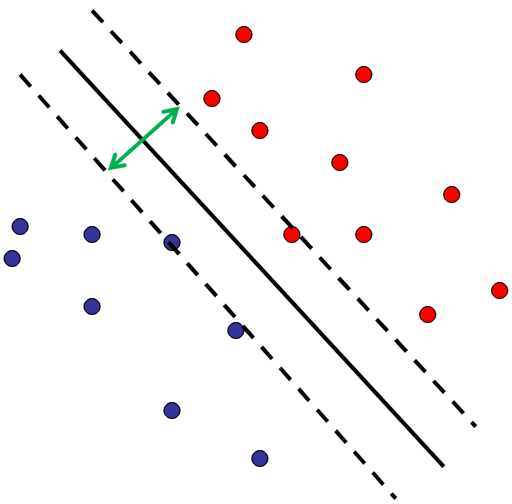


$$\mathbf{x}_i \text{ positive}: \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0$$

$$\mathbf{x}_i \text{ negative}: \quad \mathbf{x}_i \cdot \mathbf{w} + b < 0$$

Which line is best?

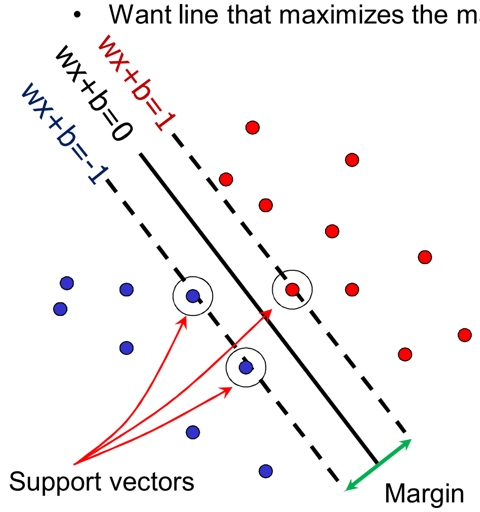
Support Vector Machines (SVMs)



- Discriminative classifier based on *optimal separating line (for 2d case)*
- Maximize the *margin* between the positive and negative training examples

Support vector machines

- Want line that maximizes the margin.



$\mathbf{x}_i$ positive $(y_i = 1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

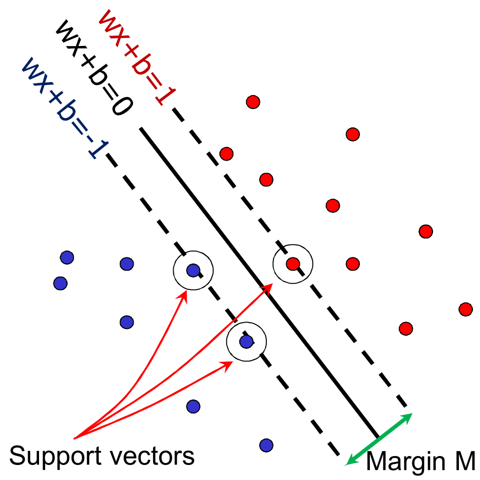
$\mathbf{x}_i$ negative $(y_i = -1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

For support, vectors, $\quad \mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Support vector machines

- Want line that maximizes the margin.



$\mathbf{x}_i$ positive $(y_i = 1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

$\mathbf{x}_i$ negative $(y_i = -1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

For support, vectors, $\quad \mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

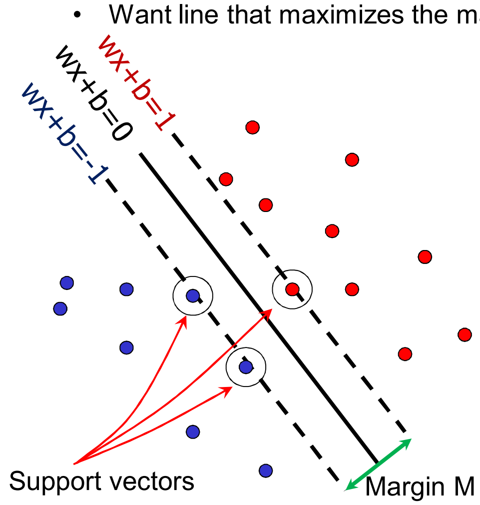
Distance between point and line: $\quad \dfrac{|\mathbf{x}_i \cdot \mathbf{w} + b|}{\|\mathbf{w}\|}$

For support vectors:

$$\frac{\mathbf{w}^T\mathbf{x} + b}{\|\mathbf{w}\|} = \frac{\pm 1}{\|\mathbf{w}\|} \qquad M = \left| \frac{1}{\|\mathbf{w}\|} - \frac{-1}{\|\mathbf{w}\|} \right| = \frac{2}{\|\mathbf{w}\|}$$

Support vector machines

- Want line that maximizes the margin.



$\mathbf{x}_i$ positive $(y_i = 1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

$\mathbf{x}_i$ negative $(y_i = -1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

For support, vectors, $\quad \mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

Distance between point and line: $\quad \dfrac{|\mathbf{x}_i \cdot \mathbf{w} + b|}{\|\mathbf{w}\|}$

Therefore, the margin is $2 / \|\mathbf{w}\|$

Finding the maximum margin line

1. Maximize margin $2/\|\mathbf{w}\|$
2. Correctly classify all training data points:

$$\mathbf{x}_i \text{ positive } (y_i = 1): \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$$
$$\mathbf{x}_i \text{ negative } (y_i = -1): \quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$$

Quadratic optimization problem:

Minimize $\frac{1}{2}\mathbf{w}^T\mathbf{w}$

Subject to $y_i(\mathbf{w}\cdot\boldsymbol{x}_i+b) \geq 1$

Finding the maximum margin line

- Solution: $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$

learned weight

Support vector

Finding the maximum margin line

- Solution: $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$

  $b = y_i - \mathbf{w} \cdot \mathbf{x}_i$ (for any support vector)

  $$\mathbf{w} \cdot \mathbf{x} + b = \sum_i \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b$$

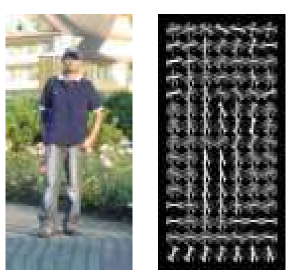
- Classification function:

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

$$= \text{sign}\left(\sum_{\text{i}} \alpha_{\text{i}} y_i \boxed{\mathbf{x}_i \cdot \mathbf{x}} + b\right)$$

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery,

Person detection with HoG's & linear SVM's

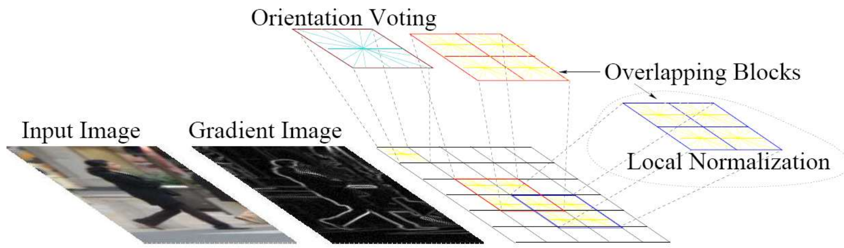


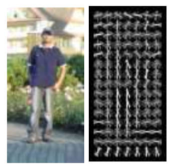
- Map each grid cell in the input window to a histogram counting the gradients per orientation.
- Train a linear SVM using training set of pedestrian vs. non-pedestrian windows.

Dalal & Triggs, CVPR 2005

Code available: http://pascal.inrialpes.fr/soft/olt/

HoG descriptor





Dalal & Triggs, CVPR 2005

Code available: http://pascal.inrialpes.fr/soft/olt/

Person detection with HoGs & linear SVMs

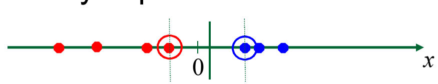
- Histograms of Oriented Gradients for Human Detection, Navneet Dalal, Bill Triggs, International Conference on Computer Vision & Pattern Recognition - June 2005
- http://lear.inrialpes.fr/pubs/2005/DT05/

Questions

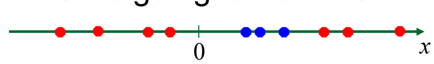
- What if the data is not linearly separable?
- What if we have more than just two categories?

Non-linear SVMs

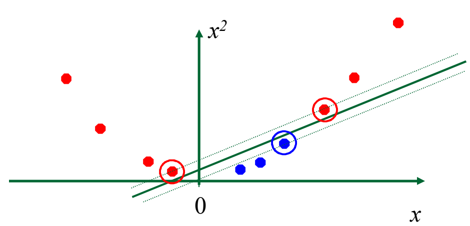
- Datasets that are linearly separable with some noise work out great:



- But what are we going to do if the dataset is just too hard?



- How about… mapping data to a higher-dimensional space:



Nonlinear SVMs

- *The kernel trick*: instead of explicitly computing the lifting transformation $\varphi(\mathbf{x})$, define a kernel function K such that

$$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$$

- This gives a nonlinear decision boundary in the original feature space:

$$\sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

Example

2-dimensional vectors $x=[x_1 \; x_2]$;

let $K(x_i,x_j)=(1 + x_i^T x_j)^2$

Need to show that $K(x_i,x_j)= \varphi(x_i)^T \varphi(x_j)$:

$$K(x_i,x_j)=(1 + x_i^T x_j)^2,$$

$$= 1+ x_{i1}^2 x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2} + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$$

$$= [1 \;\; x_{i1}^2 \;\; \sqrt{2}\, x_{i1}x_{i2} \;\; x_{i2}^2 \;\; \sqrt{2}x_{i1} \;\; \sqrt{2}x_{i2}]^T$$

$$[1 \;\; x_{j1}^2 \;\; \sqrt{2}\, x_{j1}x_{j2} \;\; x_{j2}^2 \;\; \sqrt{2}x_{j1} \;\; \sqrt{2}x_{j2}]$$

$$= \varphi(x_i)^T \varphi(x_j),$$

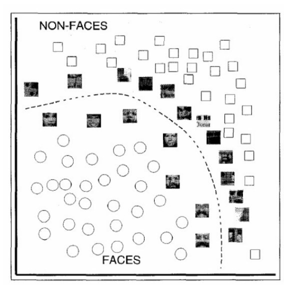
where $\varphi(x) = [1 \;\; x_1^2 \;\; \sqrt{2}\, x_1x_2 \;\; x_2^2 \;\; \sqrt{2}x_1 \;\; \sqrt{2}x_2]$

Examples of kernel functions

- Linear: $$K(x_i, x_j) = x_i^T x_j$$
- Gaussian RBF: $$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$
- Histogram intersection: $$K(x_i, x_j) = \sum_k \min(x_i(k), x_j(k))$$

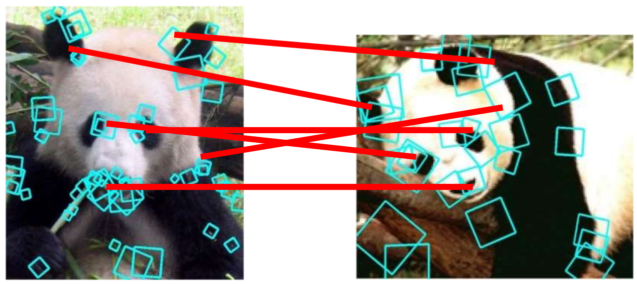
SVMs for recognition

1. Define your representation for each example.
2. Select a kernel function.
3. Compute pairwise kernel values between labeled examples
4. Use this "kernel matrix" to solve for SVM support vectors & weights.
5. To classify a new example: compute kernel values between new input and support vectors, apply weights, check sign of output.



Kristen Grauman

What about a *matching* kernel?

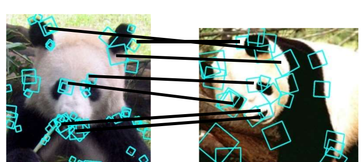


$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$ $\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$

Local feature correspondence useful similarity measure for generic object categories

Kristen Grauman

Partially matching sets of features



$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$ $\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$

Optimal match: $O(m^3)$
Greedy match: $O(m^2 \log m)$
Pyramid match: $O(m)$

(**m**=num pts)

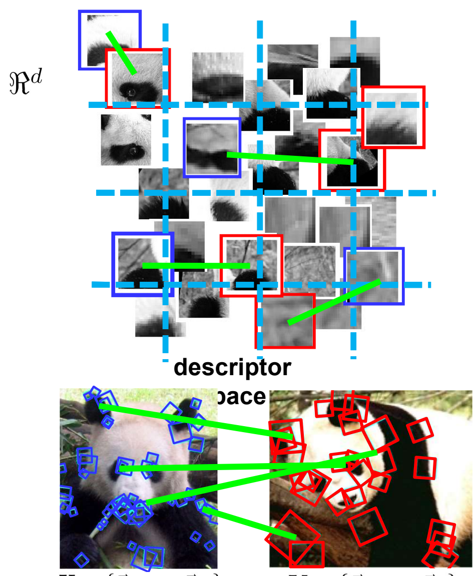
$$\min_{\pi: \mathbf{X} \to \mathbf{Y}} \sum_{\mathbf{x}_i \in \mathbf{X}} ||\mathbf{x}_i - \pi(\mathbf{x}_i)||$$

nate matching kernel that makes it practical to compare large sets of features based on their partial correspondences.

[Previous work: Indyk & Thaper, Bartal, Charikar, Agarwal & Varadarajan, …]

Kristen Grauman

Pyramid match: main idea

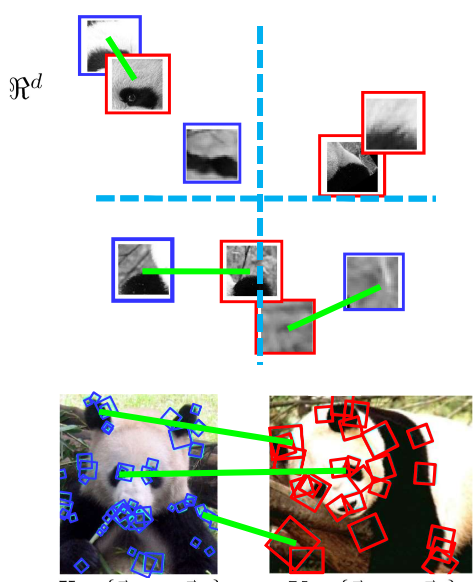


$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$ $\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$

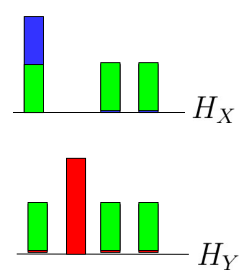
Feature space partitions serve to “match” the local descriptors within successively wider regions.

Kristen Grauman

Pyramid match: main idea



$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$ $\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$



$$\mathcal{I}(H_X, H_Y) = \sum_j \min\left(H_X(j), H_Y(j)\right)$$
$$= 3$$

Histogram intersection counts number of possible matches at a given partitioning.

Kristen Grauman

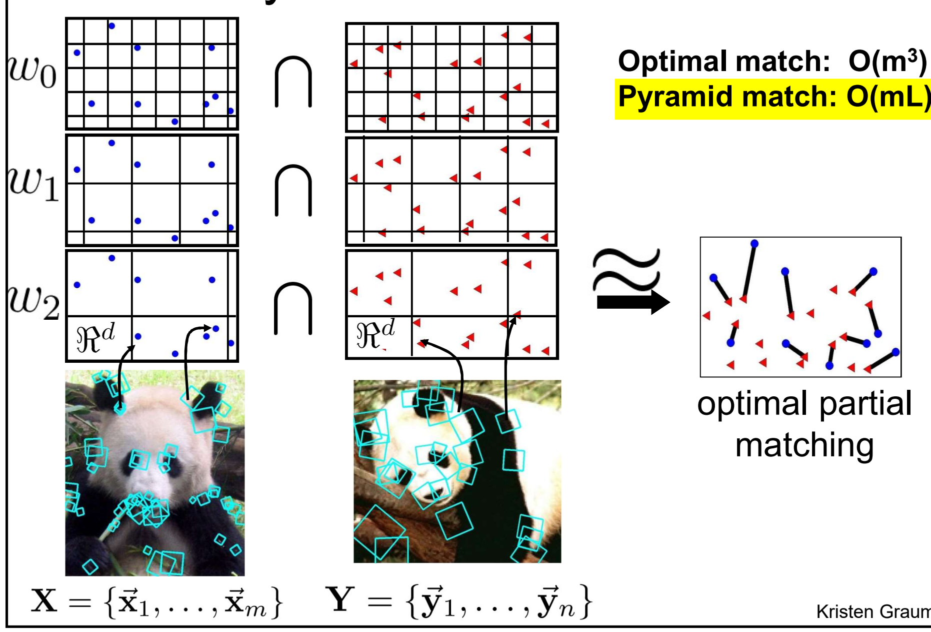
Pyramid match kernel

$$K_\Delta(X,Y) = \sum_{i=0}^{L} \underbrace{2^{-i}}_{\substack{\text{measures}\\ \text{difficulty of a}\\ \text{match at level } i}} \underbrace{\left(\mathcal{I}\left(H_X^{(i)}, H_Y^{(i)}\right) - \mathcal{I}\left(H_X^{(i-1)}, H_Y^{(i-1)}\right) \right)}_{\substack{\text{number of newly matched}\\ \text{pairs at level } i}}$$

- For similarity, weights inversely proportional to bin size (or may be learned)
- Normalize these kernel values to avoid favoring large sets

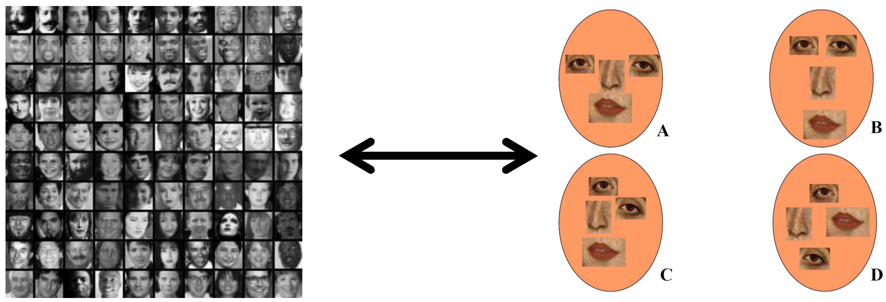
[Grauman & Darrell, ICCV 2005]

Pyramid match kernel



Kristen Grauman

Unordered sets of local features: **No** spatial layout preserved!

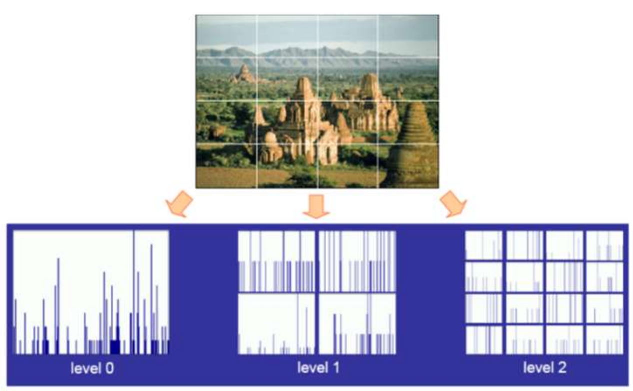


Too much?

Too little?

Spatial pyramid match

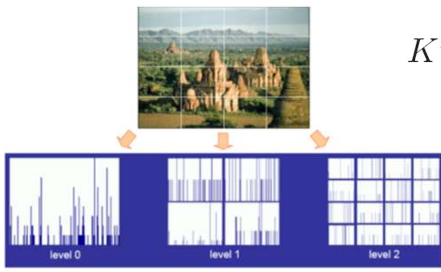
- Make a pyramid of bag-of-words histograms.
- Provides some loose (global) spatial layout information



[Lazebnik, Schmid & Ponce, CVPR 2006]

Spatial pyramid match

- Make a pyramid of bag-of-words histograms.
- Provides some loose (global) spatial layout information



$$K^L(X,Y) = \sum_{m=1}^{M} \kappa^L(X_m, Y_m)$$

Sum over PMKs computed in *image coordinate* space, one per word.

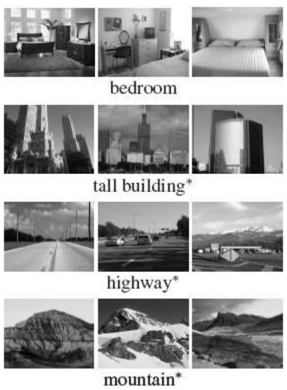
[Lazebnik, Schmid & Ponce, CVPR 2006]

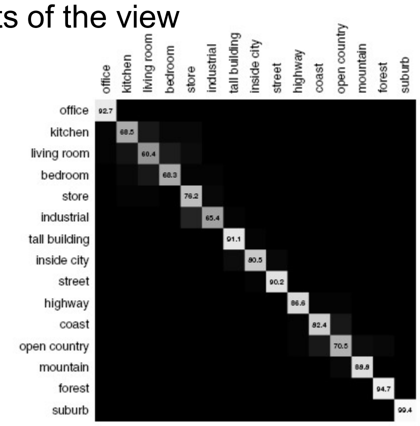
Spatial pyramid match

- Can capture **scene** categories well---texture-like patterns but with some variability in the positions of all the local

Spatial pyramid match

- Can capture **scene** categories well---texture-like patterns but with some variability in the positions of all the local pieces.
- Sensitive to global shifts of the view





Confusion table

Questions

- What if the data is not linearly separable?
- **What if we have more than just two categories?**

Multi-class SVMs

- Achieve multi-class classifier by combining a number of binary classifiers
- **One vs. all**
  - Training: learn an SVM for each class vs. the rest
  - Testing: apply each SVM to test example and assign to it the class of the SVM that returns the highest decision value
- **One vs. one**
  - Training: learn an SVM for each pair of classes
  - Testing: each learned SVM "votes" for a class to assign to the test example

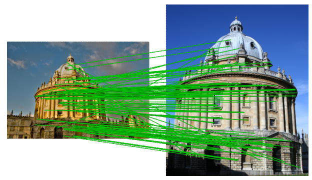
Kristen Grauman

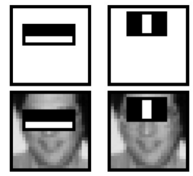
SVMs: Pros and cons

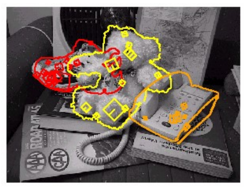
- Pros
  - Kernel-based framework is very powerful, flexible
  - Often a sparse set of support vectors – compact at test time
  - Work very well in practice, even with very small training sample sizes
- Cons
  - No "direct" multi-class SVM, must combine two-class SVMs
  - Can be tricky to select best kernel function for a problem
  - Computation, memory
    - During training time, must compute matrix of kernel values for every pair of examples
    - Learning can take a very long time for large-scale problems

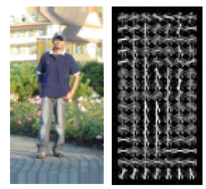
Adapted from Lana Lazebnik

Basic recognition models so far









Instances: recognition by alignment

Categories: Holistic appearance models (and sliding window detection)

Kristen Grauman

Summary so far

- Basic pipeline for window-based detection
  - Model/representation/classifier choice
  - Sliding window and classifier scoring
- Discriminative classifiers for window-based representations
  - Boosting
    - Viola-Jones face detector example
  - Nearest neighbors
    - Scene recognition example
    - 80M Tiny Images studies
  - Support vector machines
    - HOG person detection example
    - Pyramid match kernel

Today

- Intro to categorization problem
- Object categorization as discriminative classification
  - Boosting + fast face detection example
  - Nearest neighbors + scene recognition example
  - Support vector machines + pedestrian detection example
    - Pyramid match kernels, spatial pyramid match
  - Convolutional neural networks + ImageNet example
- Some new representations along the way
  - Rectangular filters
  - GIST
  - HOG

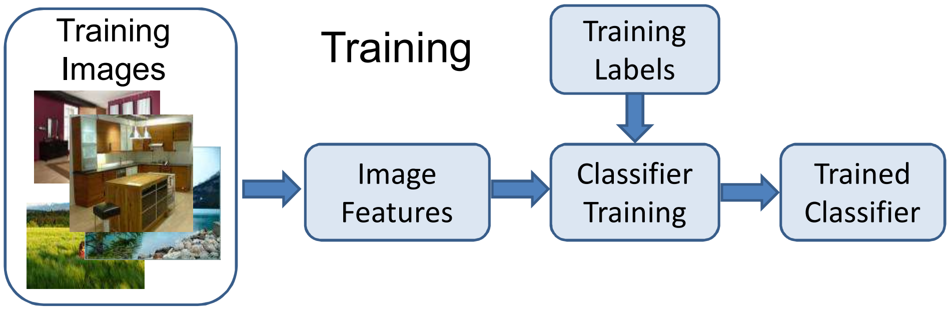
Evolution of methods

- Hand-crafted models
- 3D geometry
- Hypothesize and align

- Hand-crafted features
- Learned models
- Data-driven

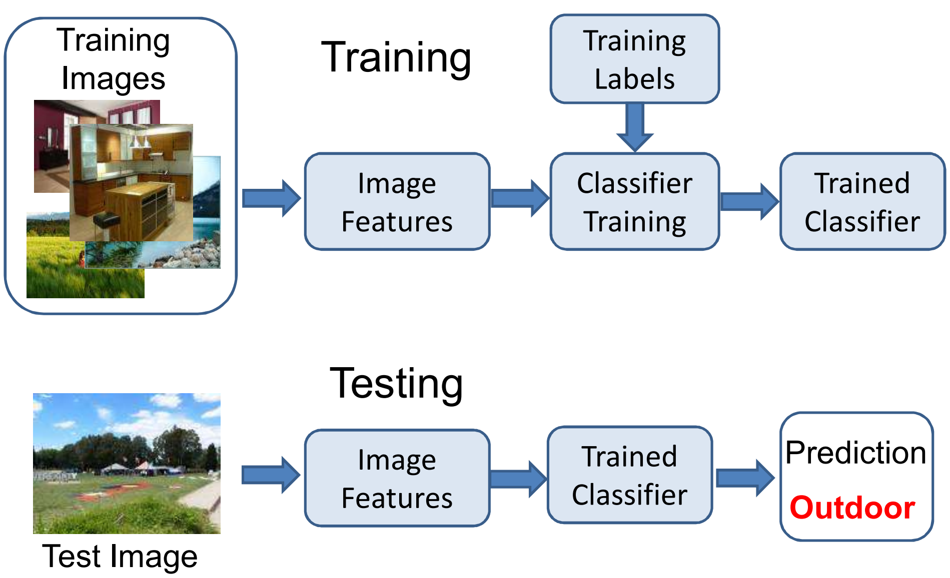
- "End-to-end" learning of features and models*,**

Traditional Image Categorization: Training phase

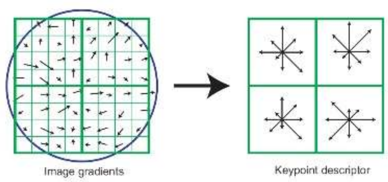


Jia-Bin Huang and Derek Hoiem, UIUC

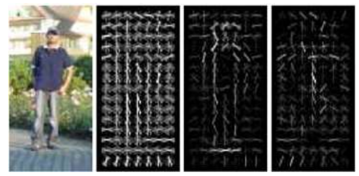
Traditional Image Categorization: Testing phase



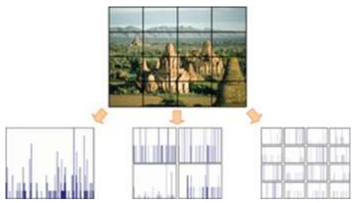
Features have been key



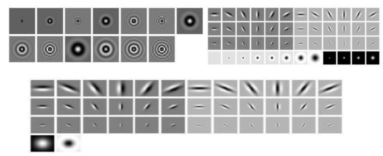
SIFT [Lowe IJCV 04]



HOG [Dalal and Triggs CVPR 05]



SPM [Lazebnik et al. CVPR 06]



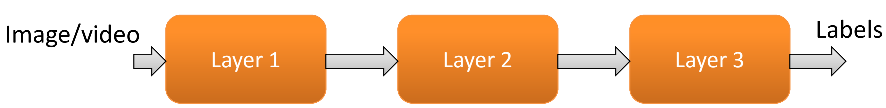
Textons

and many others:

SURF, MSER, LBP, Color-SIFT, Color histogram, GLOH, …..

Learning a Hierarchy of Feature Extractors

- Each layer of hierarchy extracts features from output of previous layer
- All the way from pixels → classifier
- Layers have the (nearly) same structure



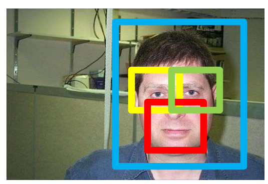
- Train all layers jointly

Slide: Rob Fergus

Learning Feature Hierarchy

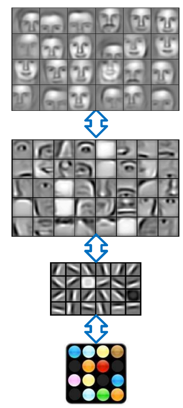
Goal: Learn **useful higher-level features** from images

Input data



Lee et al., ICML 2009;
CACM 2011

Feature representation



3rd layer
"Objects"

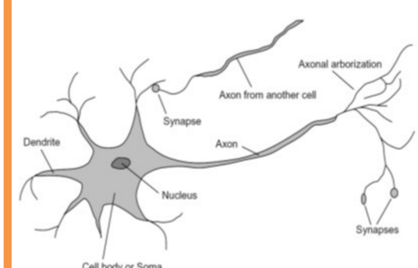
2nd layer
"Object parts"

1st layer
"Edges"

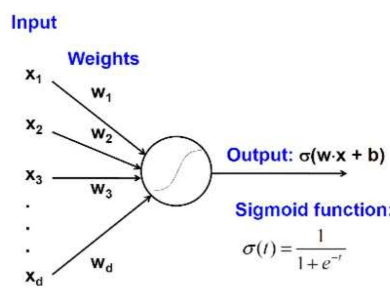
Pixels

Slide: Rob Fergus

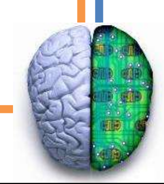
Biological neuron and Perceptrons



A biological neuron



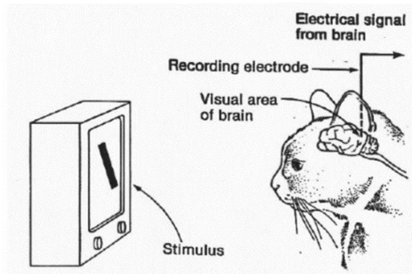
An artificial neuron (Perceptron)
- a linear classifier



Jia-Bin Huang and Derek Hoiem, UIUC

Simple, Complex and Hypercomplex cells

David H. Hubel and Torsten Wiesel

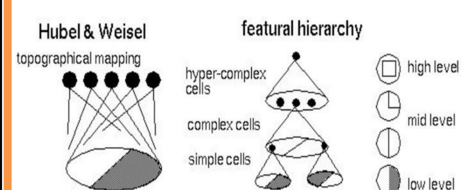


Suggested a **hierarchy** of **feature detectors** in the visual cortex, with higher level features responding to patterns of activation in lower level cells, and propagating activation upwards to still higher level cells.

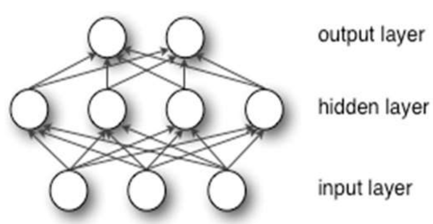
David Hubel's Eye, Brain, and Vision

Jia-Bin Huang and Derek Hoiem, UIUC

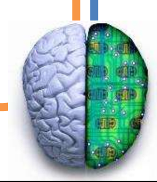
Hubel/Wiesel Architecture and Multi-layer Neural Network



Hubel and Weisel's architecture



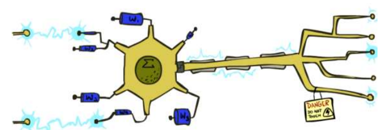
Multi-layer Neural Network
- A *non-linear* classifier



Jia-Bin Huang and Derek Hoiem, UIUC

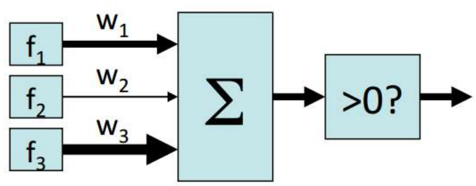
Neuron: Linear Perceptron

- Inputs are feature values
- Each feature has a weight
- Sum is the activation



$$\text{activation}_w(x) = \sum_i w_i \cdot f_i(x) = w \cdot f(x)$$

- If the activation is:
  - Positive, output +1
  - Negative, output -1



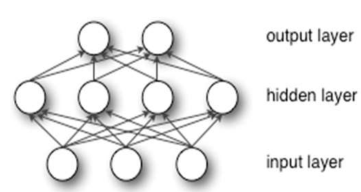
Slide credit: Pieter Abeel and Dan Klein

Multi-layer Neural Network

- A non-linear classifier
- **Training:** find network weights **w** to minimize the loss, e.g., error between true training labels y_i and estimated labels $f_{\mathbf{w}}(\mathbf{x}_i)$

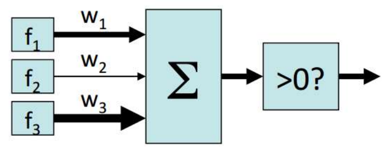
$$E(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - f_{\mathbf{w}}(\mathbf{x}_i)\right)^2$$

- Minimization can be done by gradient descent provided f is differentiable
- This training method is called back-propagation



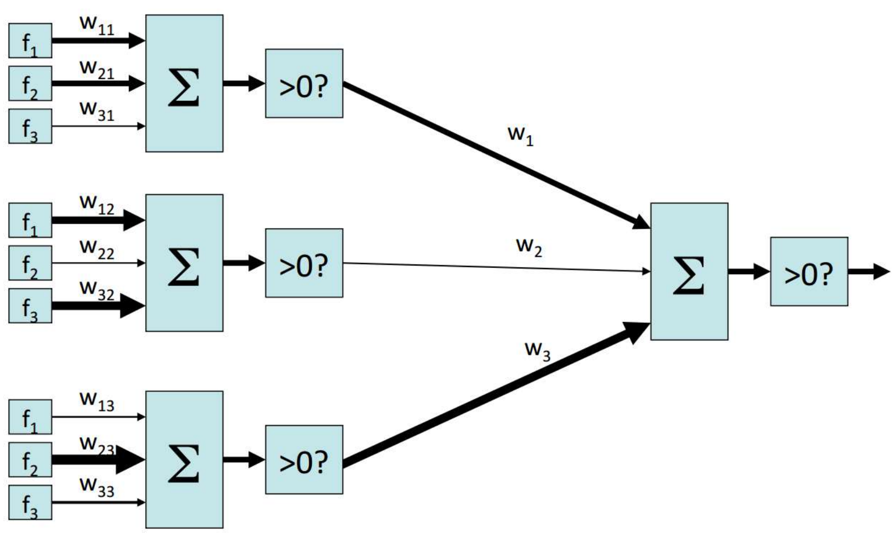
Jia-Bin Huang and Derek Hoiem, UIUC

Two-layer perceptron network



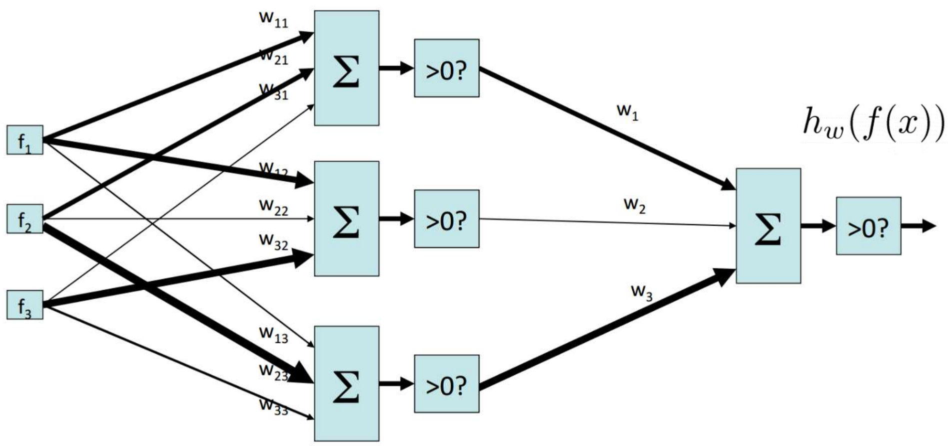
Slide credit: Pieter Abeel and Dan Klein

Two-layer perceptron network



Slide credit: Pieter Abeel and Dan Klein

Two-layer perceptron network



Slide credit: Pieter Abeel and Dan Klein

Learning w

- Training examples

$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})$$

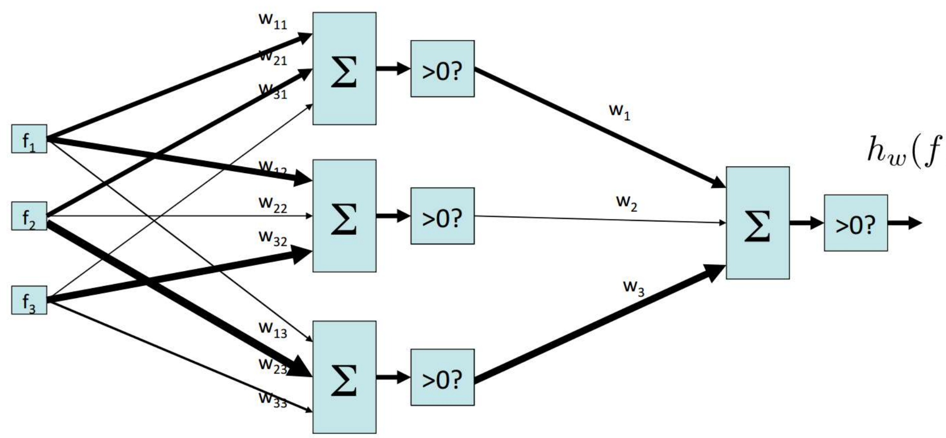
- Objective: a misclassification loss, e.g.

$$\min_w \sum_{i=1}^{m} \left(y^{(i)} - h_w(f(x^{(i)})) \right)^2$$

- Procedure:
  - Gradient descent / hill climbing

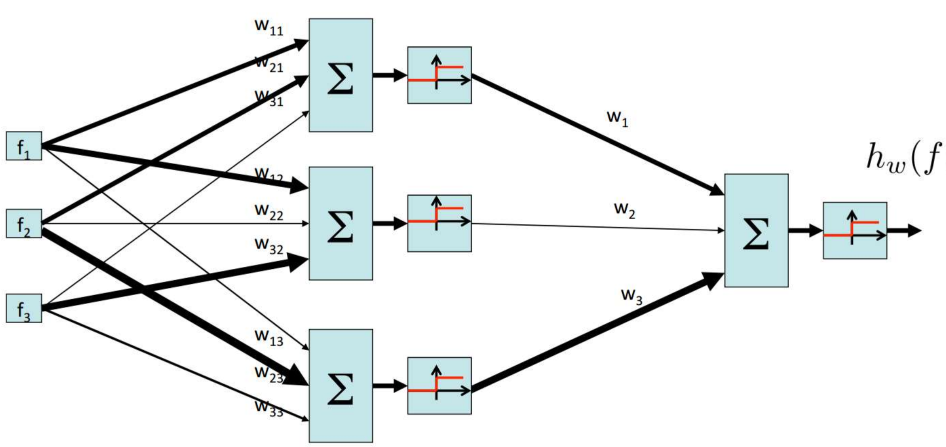
Slide credit: Pieter Abeel and Dan Klein

Two-layer perceptron network



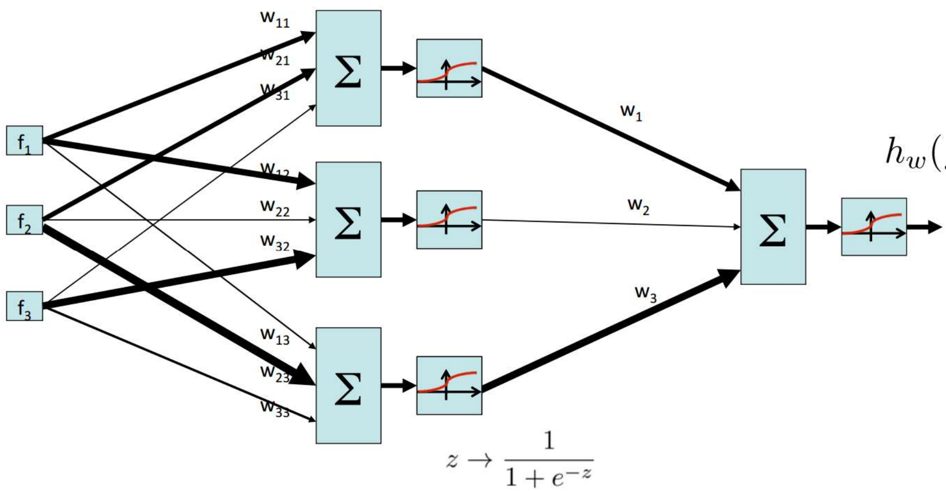
Slide credit: Pieter Abeel and Dan Klein

Two-layer perceptron network



Slide credit: Pieter Abeel and Dan Klein

Two-layer neural network



Slide credit: Pieter Abeel and Dan Klein

Neural network properties

- Theorem (Universal function approximators): A two-layer network with a sufficient number of neurons can approximate any continuous function to any desired accuracy
- Practical considerations:
  - Can be seen as learning the features
  - Large number of neurons
    - Danger for overfitting
  - Hill-climbing procedure can get stuck in bad local optima

Approximation by Superpositions of Sigmoidal Function,1989

Slide credit: Pieter Abeel and Dan Klein

Convolutional Neural Networks
(CNN, ConvNet, DCN)

- CNN = a multi-layer neural network with
  - **Local** connectivity:
    - Neurons in a layer are only connected to a small region of the layer before it
  - **Share** weight parameters across spatial positions:
    - Learning shift-invariant filter kernels

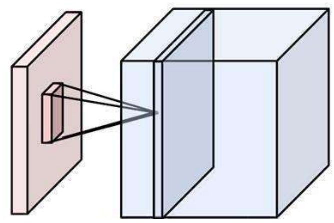
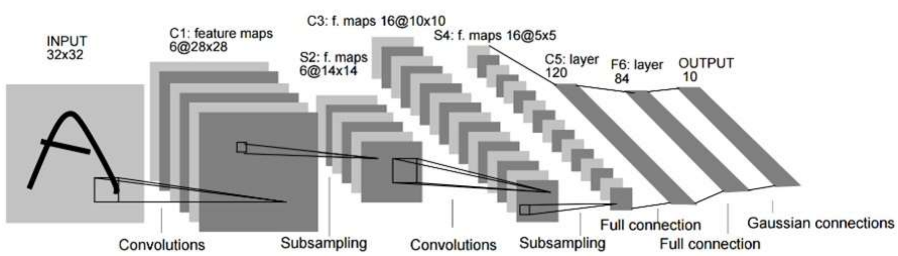


Image credit: A. Karpathy

Jia-Bin Huang and Derek Hoiem, UIUC

LeNet [LeCun et al. 1998]



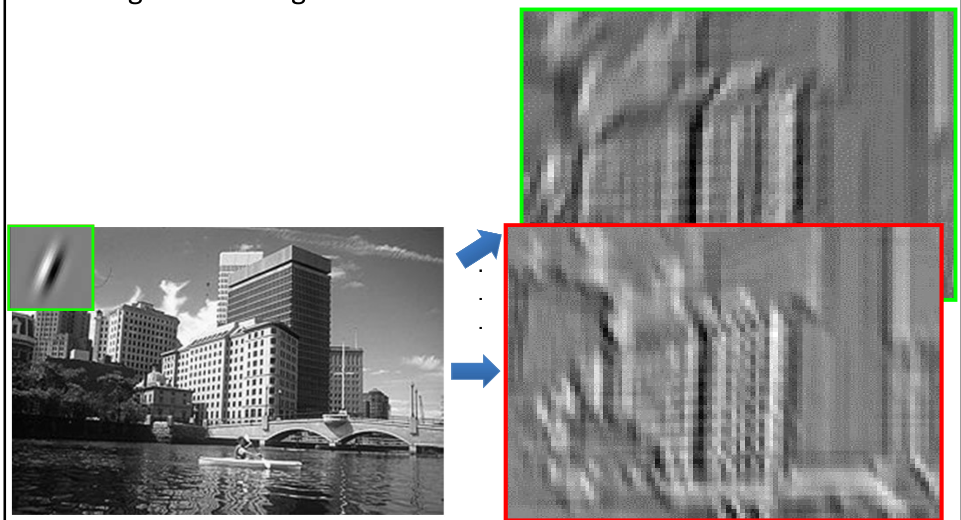
LeNet-1 from 1993

Gradient-based learning applied to document recognition [LeCun, Bottou, Bengio, Haffner 1998]

Jia-Bin Huang and Derek Hoiem, UIUC

What is a Convolution?

- Weighted moving sum

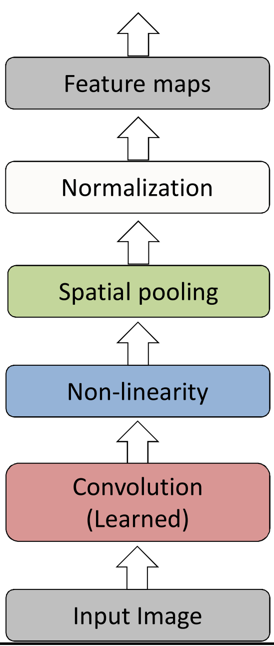


Input

Feature Activation Map

slide credit: S. Lazebnik

Convolutional Neural Networks



slide credit: S. Lazebnik

Convolutional Neural Networks

Feature maps

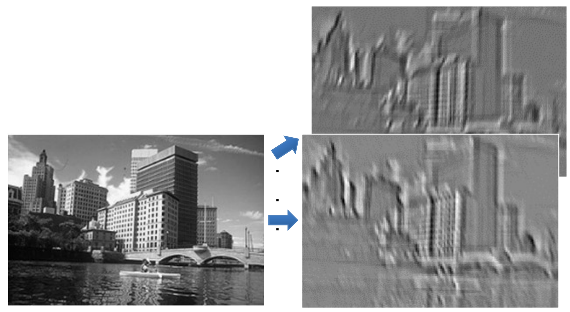
Normalization

Spatial pooling

Non-linearity

Convolution (Learned)

Input Image



Input

Feature Map

slide credit: S. Lazebnik

Convolutional Neural Networks

Feature maps

Normalization

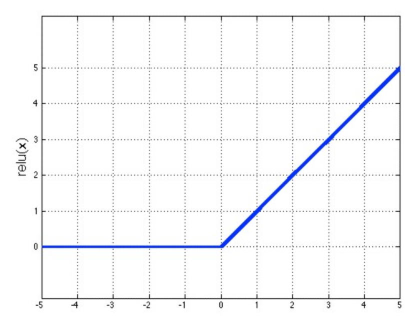
Spatial pooling

Non-linearity

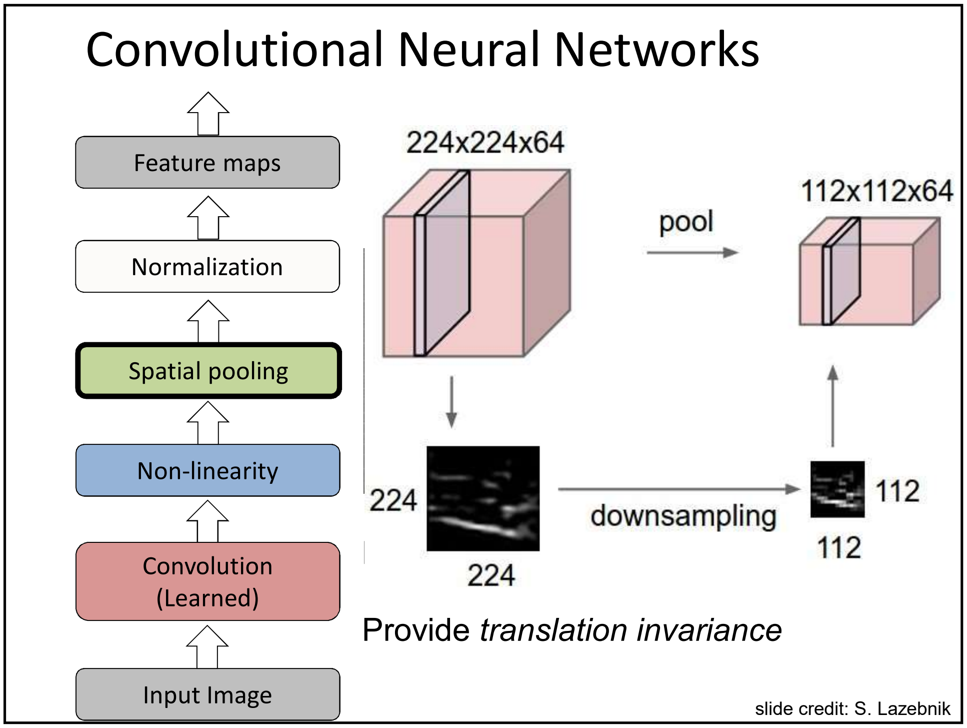
Convolution (Learned)

Input Image

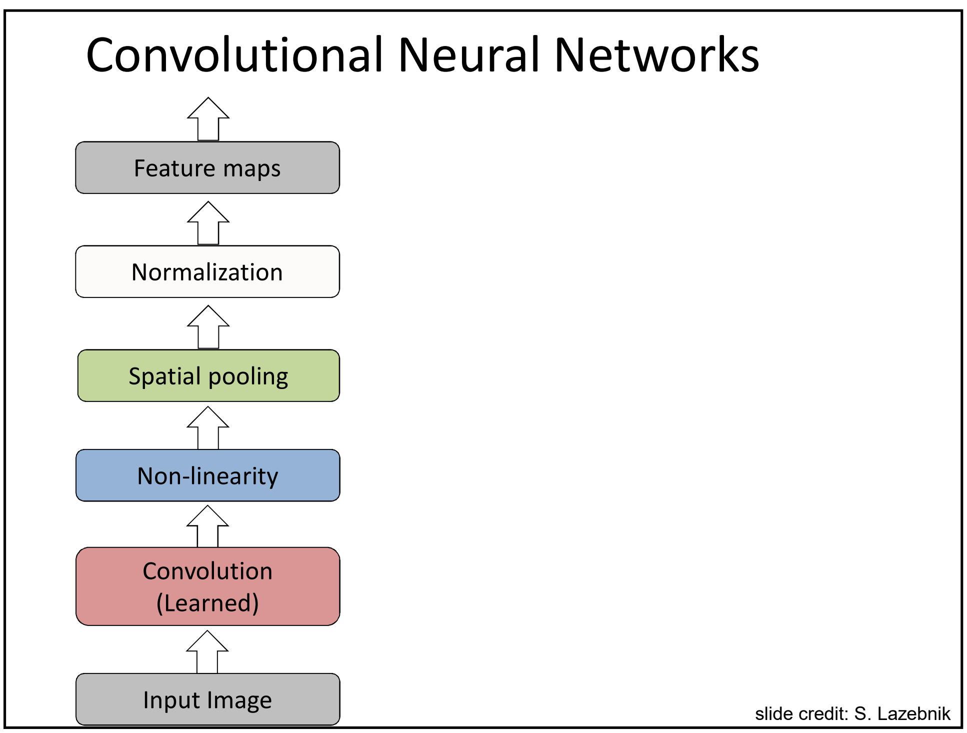
Rectified Linear Unit (ReLU)



slide credit: S. Lazebnik



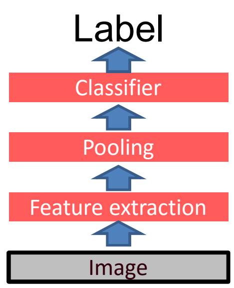
Convolutional Neural Networks
Feature maps
Normalization
Spatial pooling
Non-linearity
Convolution (Learned)
Input Image
224x224x64
pool
112x112x64
224
224
downsampling
112
112
Provide *translation invariance*
slide credit: S. Lazebnik

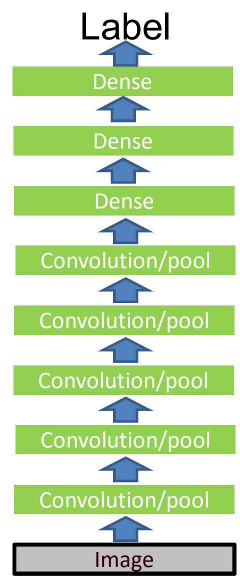


Convolutional Neural Networks
Feature maps
Normalization
Spatial pooling
Non-linearity
Convolution (Learned)
Input Image
slide credit: S. Lazebnik

Engineered vs. learned features

Convolutional filters are trained in a supervised manner by back-propagating classification error

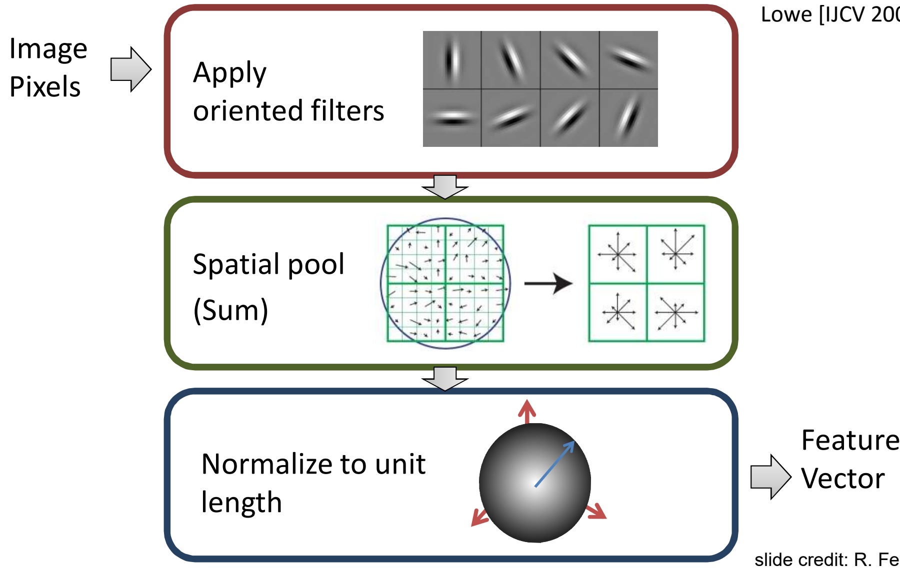




Jia-Bin Huang and Derek Hoiem, UIUC

SIFT Descriptor

Lowe [IJCV 2004]



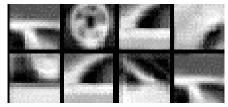
slide credit: R. Fergus

Spatial Pyramid Matching

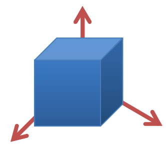
Lazebnik, Schmid, Ponce [CVPR 2006]

SIFT Features

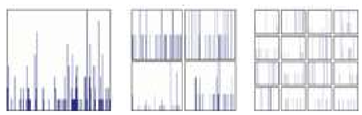
Filter with Visual Words



Max



Multi-scale spatial pool (Sum)

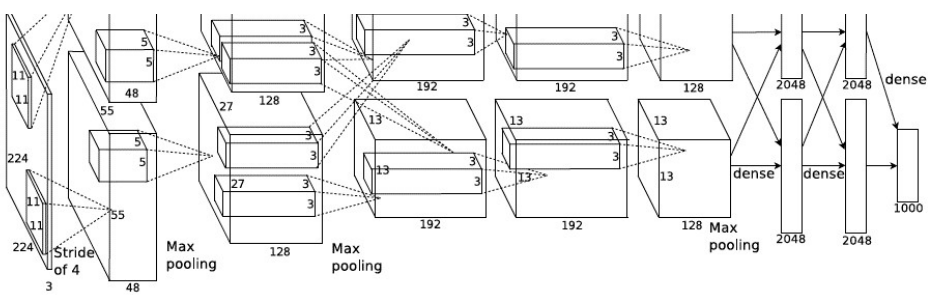


Classifier

slide credit: R. Fergus

AlexNet

- Similar framework to LeCun'98 but:
  - Bigger model (7 hidden layers, 650,000 units, 60,000,000 params)
  - More data (10^6 vs. 10^3 images)
  - GPU implementation (50x speedup over CPU)
    - Trained on two GPUs for a week



A. Krizhevsky, I. Sutskever, and G. Hinton,
ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012

Jia-Bin Huang and Derek Hoiem, UIUC

Applications

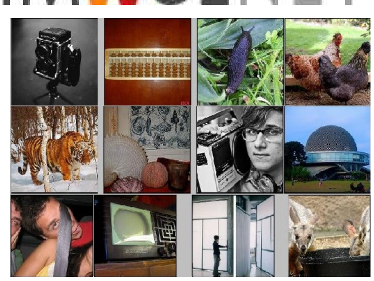
- Handwritten text/digits
  - MNIST (0.17% error [Ciresan et al. 2011])
  - Arabic & Chinese [Ciresan et al. 2012]

- Simpler recognition benchmarks
  - CIFAR-10 (9.3% error [Wan et al. 2013])
  - Traffic sign recognition
    - 0.56% error vs 1.16% for humans [Ciresan et al. 2011]

Slide: R. Fergus

Application: ImageNet

IMAGENET



[Deng et al. CVPR 2009]

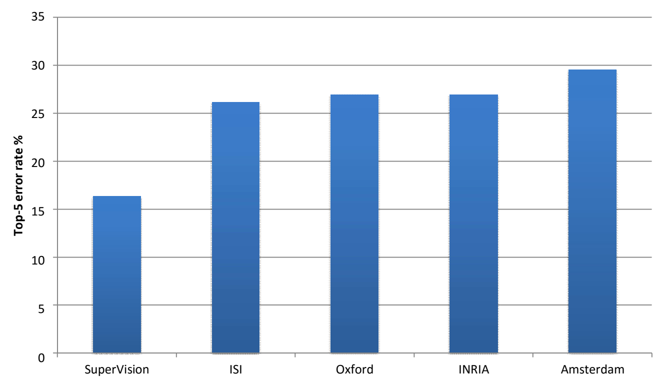
- ~14 million labeled images, 20k classes
- Images gathered from Internet
- Human labels via Amazon Turk

https://sites.google.com/site/deeplearningcvpr2014

Slide: R. Fergus

ImageNet Classification 2012

- Krizhevsky et al. -- 16.4% error (top-5)
- Next best (non-convnet) – 26.2% error

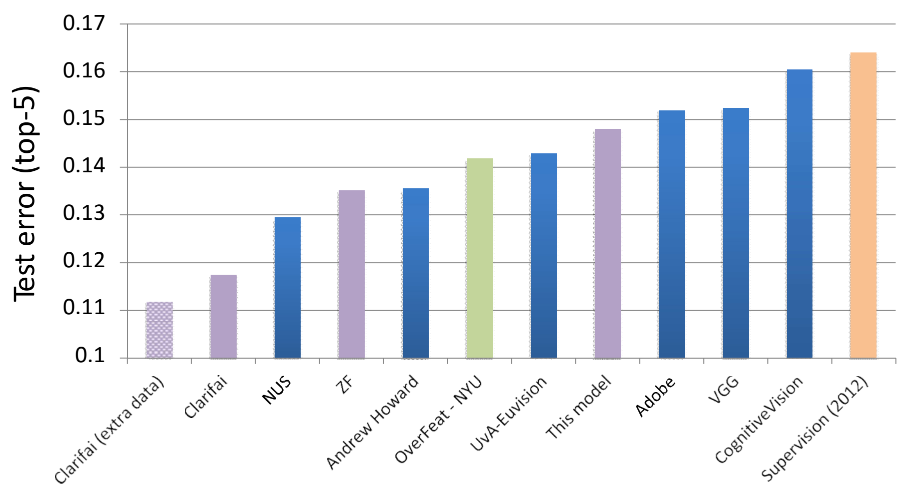


https://sites.google.com/site/deeplearningcvpr2014

Slide: R. Fergus

ImageNet Classification 2013 Results

- http://www.image-net.org/challenges/LSVRC/2013/results.php



https://sites.google.com/site/deeplearningcvpr2014

ImageNet Challenge 2012-2014

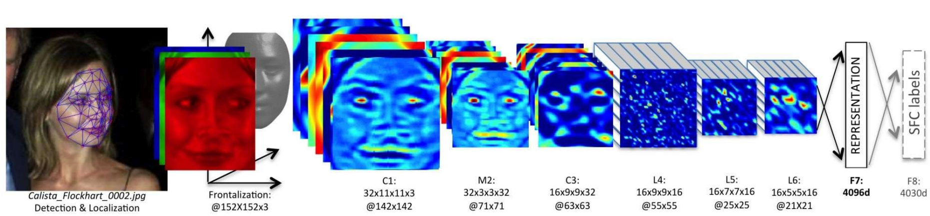
Best non-convnet in 2012: 26.2%

Team	Year	Place	Error (top-5)	External data
SuperVision – Toronto (7 layers)	2012	-	16.4%	no
SuperVision	2012	1st	15.3%	ImageNet 22k
Clarifai – NYU (7 layers)	2013	-	11.7%	no
Clarifai	2013	1st	11.2%	ImageNet 22k
VGG – Oxford (16 layers)	2014	2nd	7.32%	no
GoogLeNet (19 layers)	2014	1st	6.67%	no
Human expert*			5.1%	

Jia-Bin Huang and Derek Hoiem, UIUC

Industry Deployment

- Used in Facebook, Google, Microsoft
- Image Recognition, Speech Recognition, ….
- Fast at test time



Taigman et al. DeepFace: Closing the Gap to Human-Level Performance in Face Verification, CVPR'14

Slide: R. Fergus

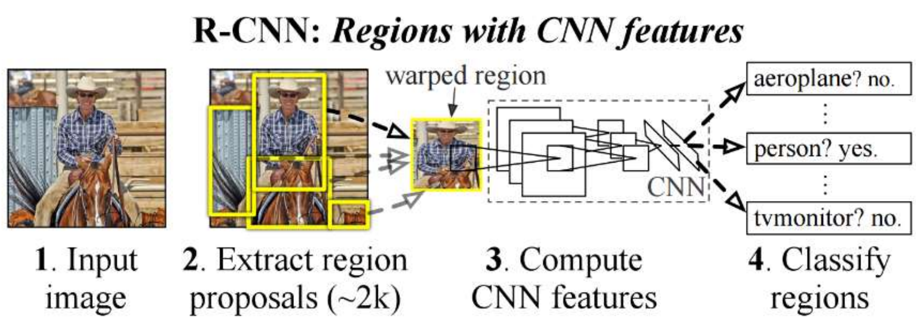
Beyond classification

- Detection
- Segmentation
- Regression
- Pose estimation
- Matching patches
- Synthesis

and many more…

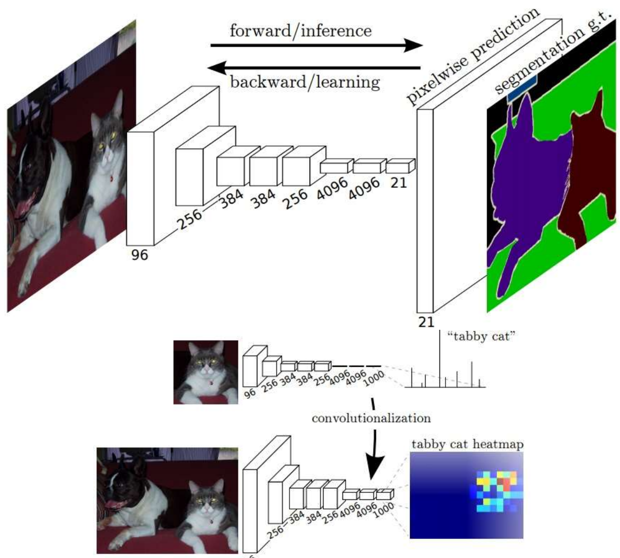
R-CNN: Regions with CNN features

- Trained on ImageNet classification
- Finetune CNN on PASCAL



RCNN [Girshick et al. CVPR 2014]

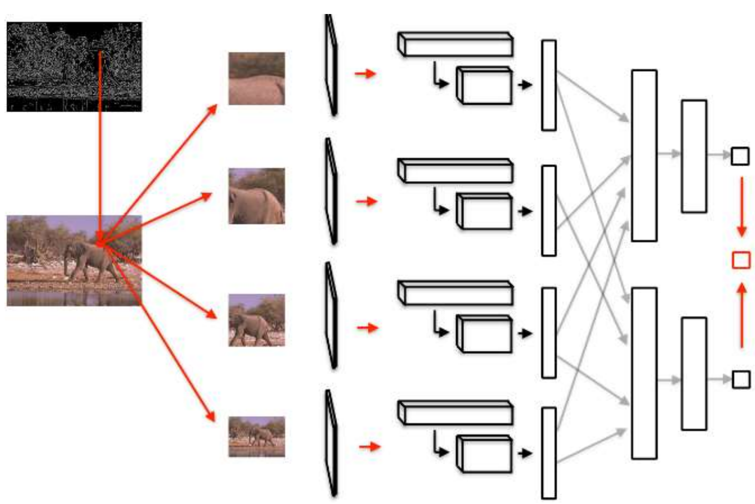
Labeling Pixels: Semantic Labels



Fully Convolutional Networks for Semantic Segmentation [Long et al. CVPR 2015]

Jia-Bin Huang and Derek Hoiem, UIUC

Labeling Pixels: Edge Detection



DeepEdge: A Multi-Scale Bifurcated Deep Network for Top-Down Contour Detection [Bertasius et al. CVPR 2015]

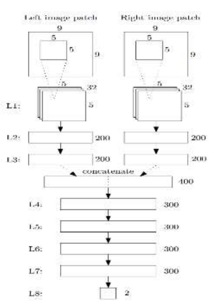
Jia-Bin Huang and Derek Hoiem, UIUC

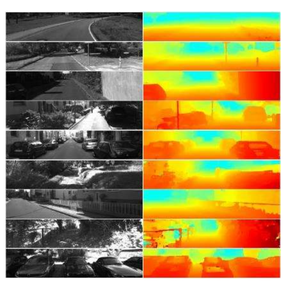
CNN for Regression

DeepPose [Toshev and Szegedy CVPR 2014]

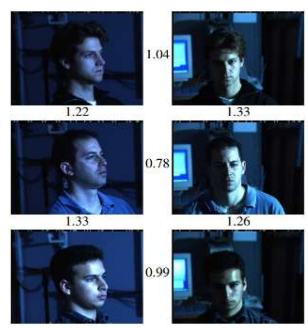
Jia-Bin Huang and Derek Hoiem, UIUC

CNN as a Similarity Measure for Matching

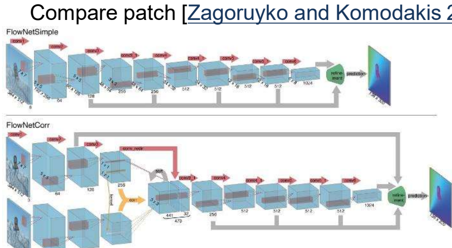




Stereo matching [Zbontar and LeCun CVPR 2015]
Compare patch [Zagoruyko and Komodakis 2015]



FaceNet [Schroff et al. 2015]

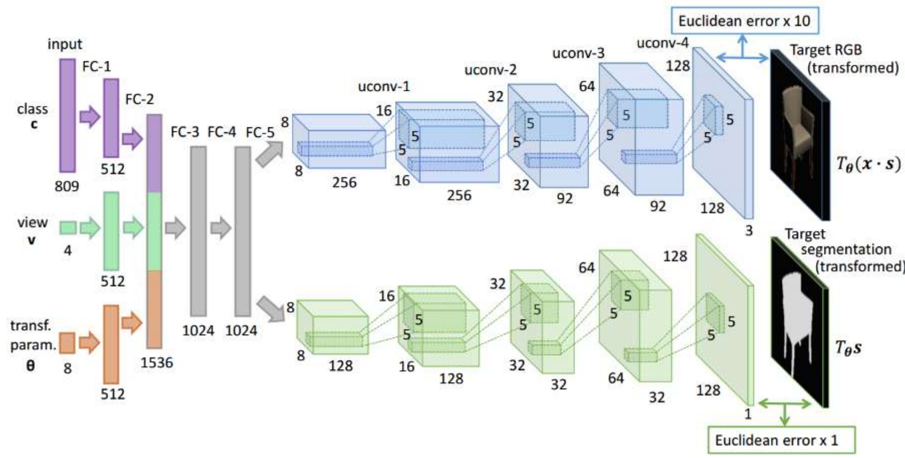


FlowNet [Fischer et al 2015]

Match ground and aerial images
[Lin et al. CVPR 2015]

Jia-Bin Huang and Derek Hoiem, UIUC

CNN for Image Generation



Learning to Generate Chairs with Convolutional Neural Networks [Dosovitskiy et al. CVPR 2015]
Jia-Bin Huang and Derek Hoiem, UIUC

Chair Morphing

Learning to Generate Chairs with Convolutional Neural Networks [Dosovitskiy et al. CVPR 2015]
Jia-Bin Huang and Derek Hoiem, UIUC

Recap

- Neural networks / multi-layer perceptrons
  - View of neural networks as learning hierarchy of features
- Convolutional neural networks
  - Architecture of network accounts for image structure
  - "End-to-end" recognition from pixels
  - Together with big (labeled) data and lots of computation → major success on benchmarks, image classification and beyond

Announcements

- Reminder: Assignment 1 due Sept 16 11:59 pm on Canvas
- Reminder: Optional CNN/Caffe tutorial on Monday Sept 12, 5-7 pm (here)