Image and Video Processing

Mid & High Level Computer Vision

Computer Vision

Computer vision is the science and technology of building artificial systems that can interpret images in a similar way as humans can do.

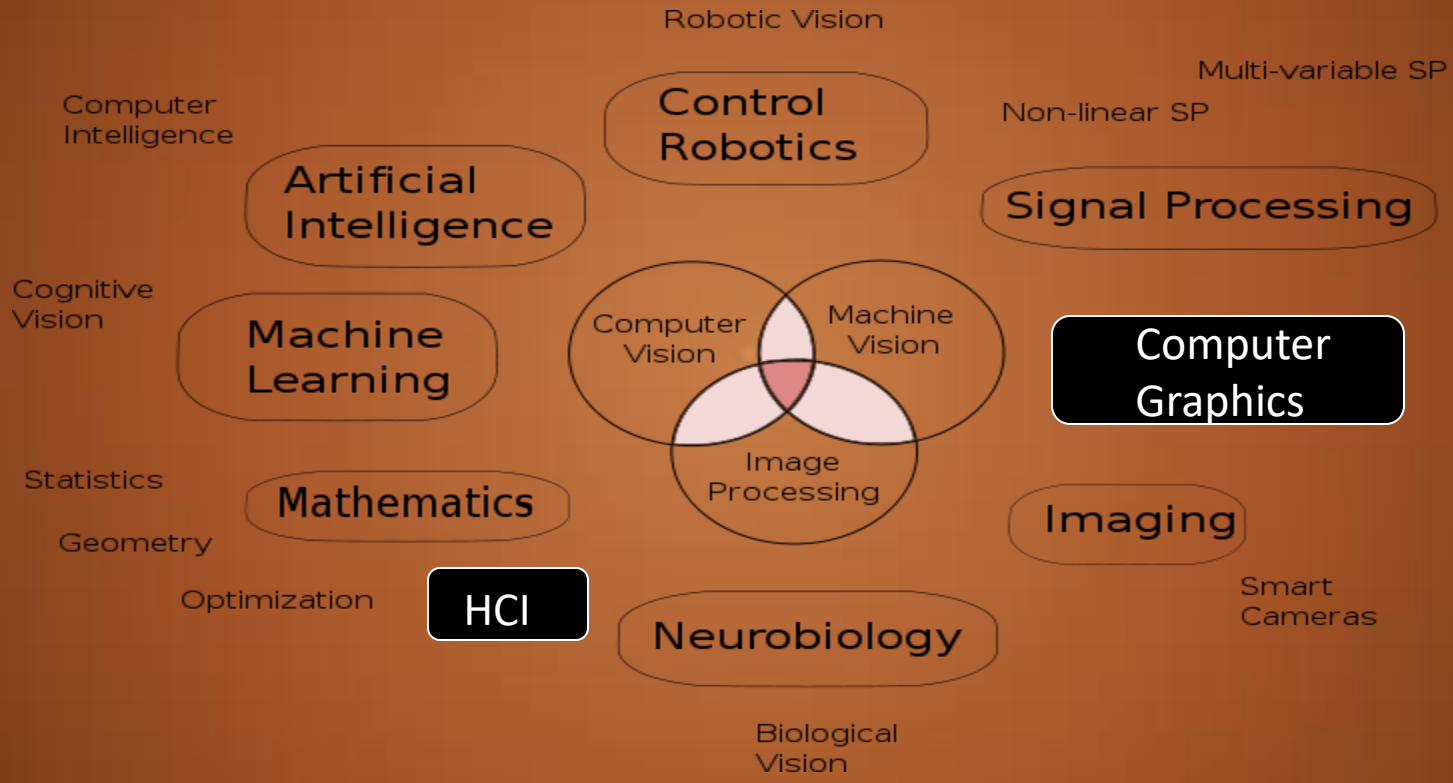
What kind of scene?

Where are the cars?

How far is the building?

…

Vision is multidisciplinary



From wiki

Computer vision vs human vision

What we see

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

What a computer sees

Vision is really hard

Vision is an amazing feat of natural intelligence

- Visual cortex occupies about 50% of Macaque brain
- More human brain devoted to vision than anything else

Is that a queen or a bishop?

Three levels of Computer Vision

Computer Vision tasks can be categorized at 3 levels –

1. Low Level Vision
2. Mid-Level Vision
3. High-Level Vision

Low-Level Vision

Image manipulation
- Resize, rotate …
- Enhancements
- Color, exposure

Low level Feature extraction
- Edges
- Region shape
- Descriptors

Mid-Level Vision

- Segmentation
- Panoramas
- Multi-view stereo
- 3D, Depth
- Motion estimation,
- Object tracking …

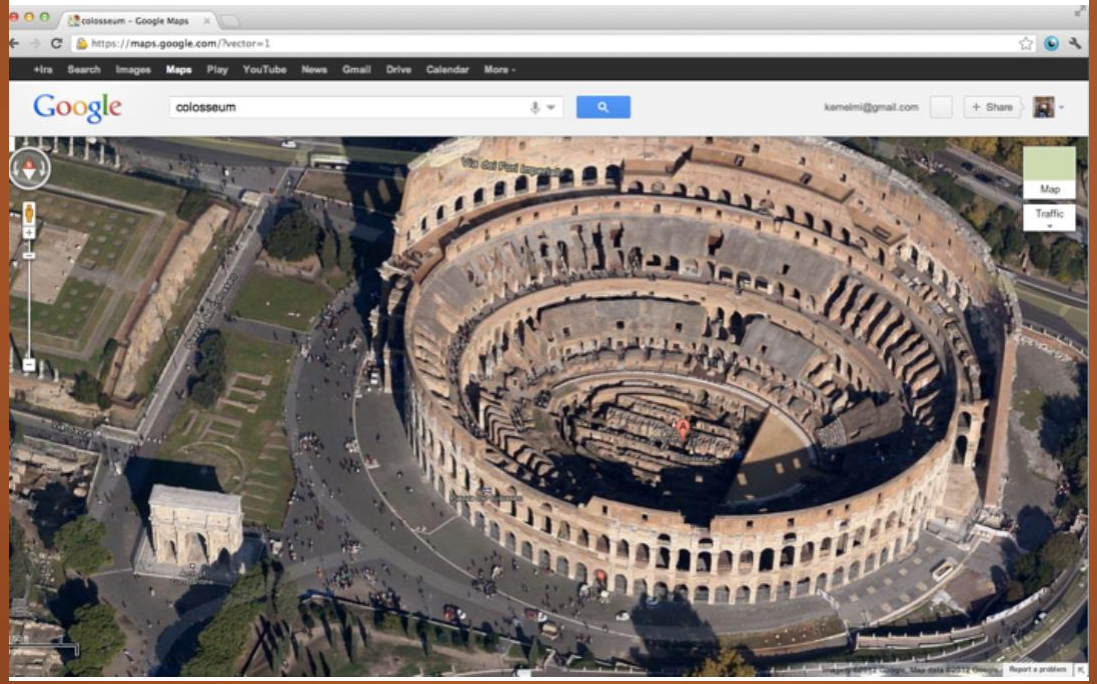
Mid-Level: Segmentation

Mid-Level: Panorama Stitching

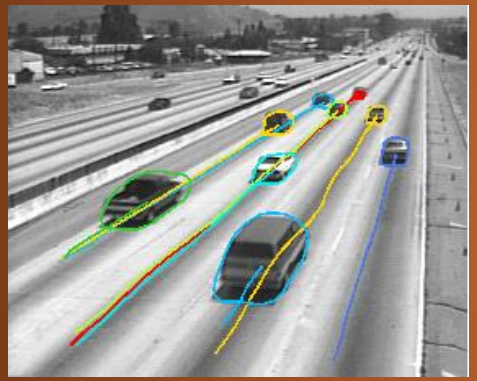
Mid-Level: Panorama Stitching

Mid-Level: 3D modeling

Google's 3D Maps
Structure estimation from tourist photos



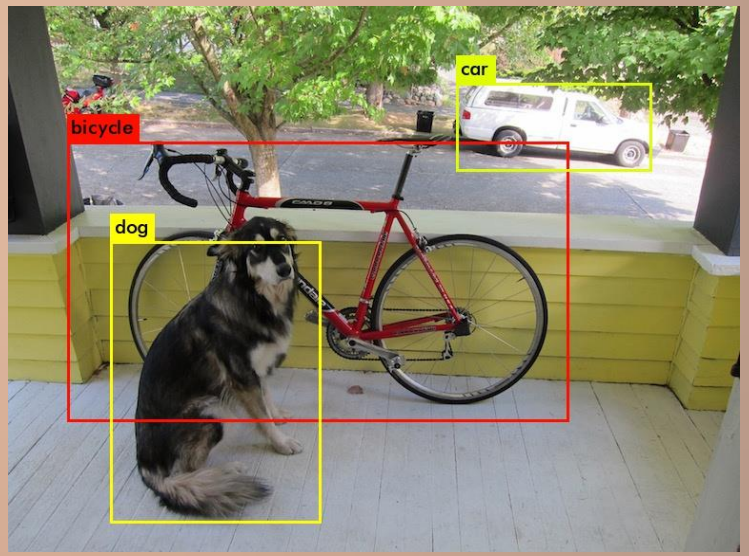
Mid-level: Vehicle/object tracking



High-Level Vision

Semantics! Hardest!

- Image classification, tagging
- Object recognition, detection
- Pose Estimation
- Activity Recognition …



High-Level: Classification

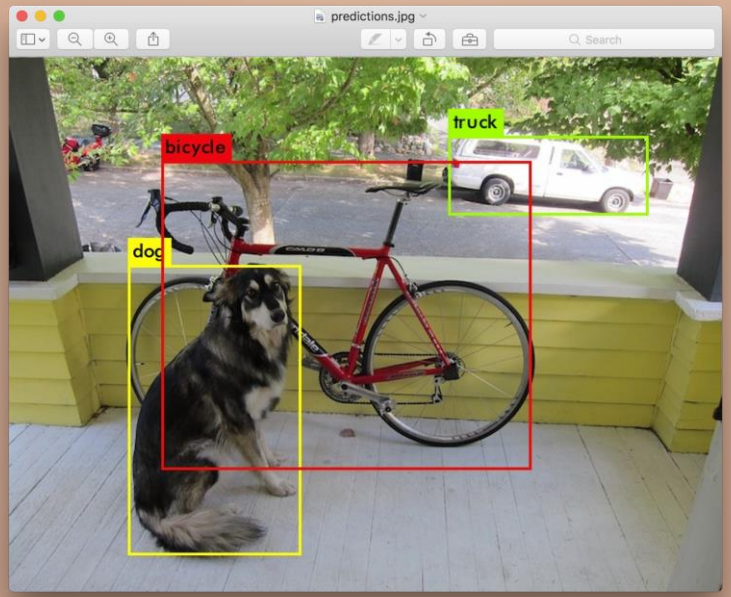
- What is in the image?

High-Level: Tagging

- What are ALL the things in the image?

High-Level: Detection

- What are ALL the things in the image?
- Where are they?



High-Level: Activity recognition/classification

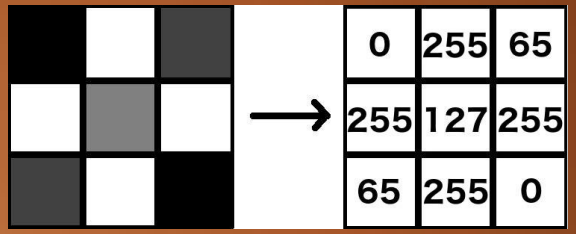
A little story about Computer Vision

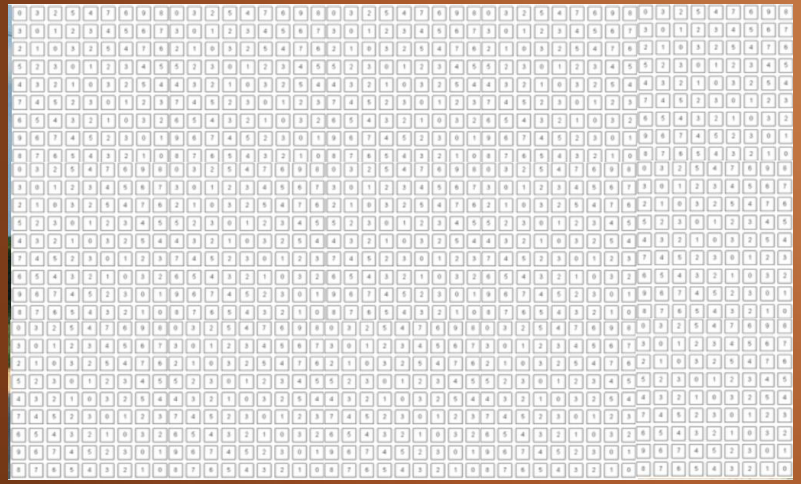
In 1966, Marvin Minsky at MIT asked his undergraduate student Gerald Jay Sussman to “spend the summer linking a camera to a computer and getting the computer to describe what it saw”. We now know that the problem is slightly more difficult than that.

(Szeliski 2009, Computer Vision)

Why is CV hard?

When machines “view” images, all they see are numbers that represent individual pixels.





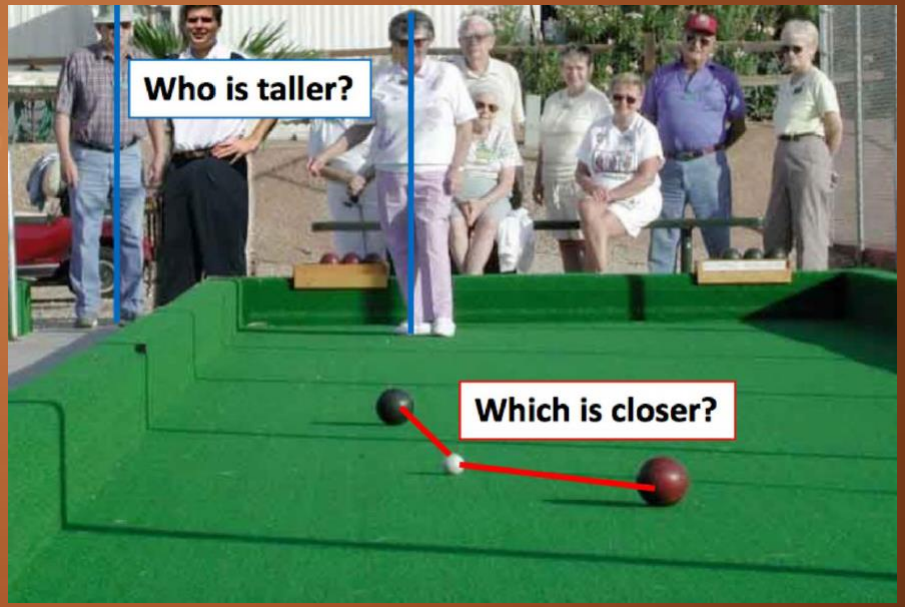
animal or not?

1. Dealing with a lot of data

- Lot of data that needs to be processed to be made sense of
- Things start to get really difficult for computer vision the higher the resolution of images. Ex. HD video at 30 fps = 3 *1920 * 1080 * 30 ~= 180 million pixels/sec
- Real time processing of all that no.’s!

2. Loss of Information

- Digitising process, noise etc.
- Compression
- 3D (or 4D) to 2D (or 3D)

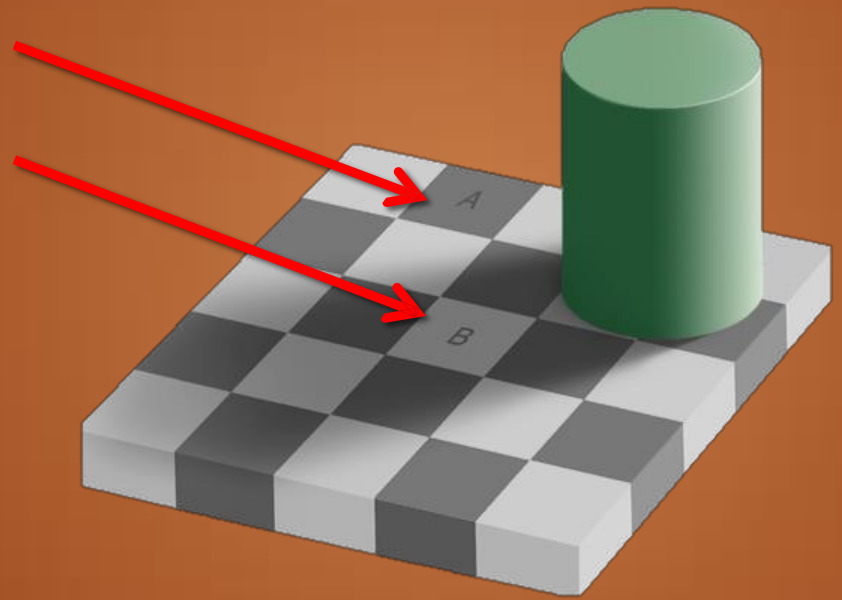


3. Illusory perceptions

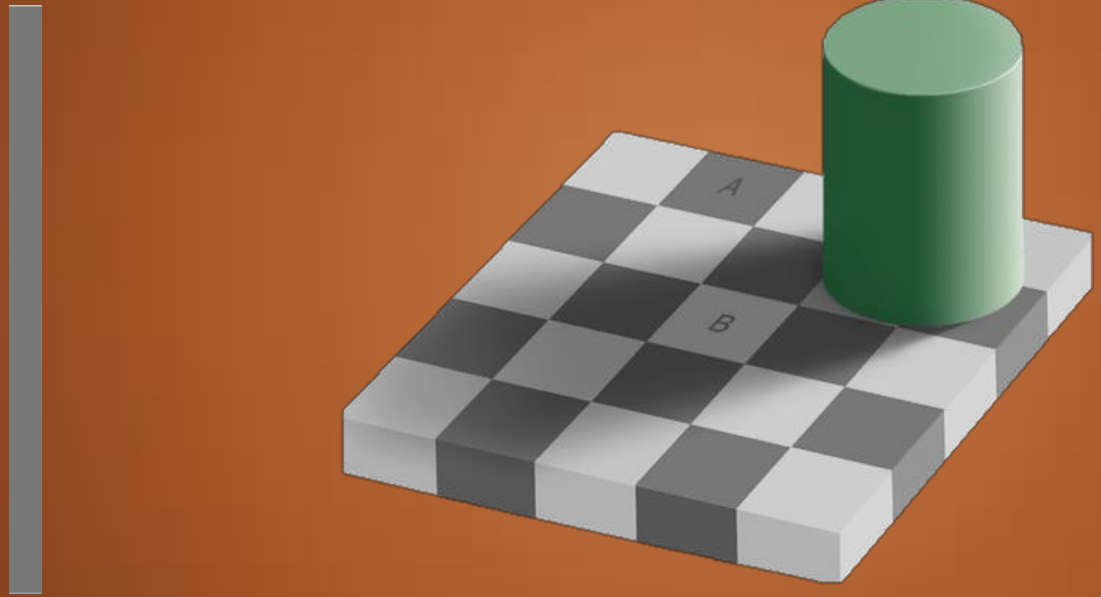
Simple scene right?

Dark square

Light square

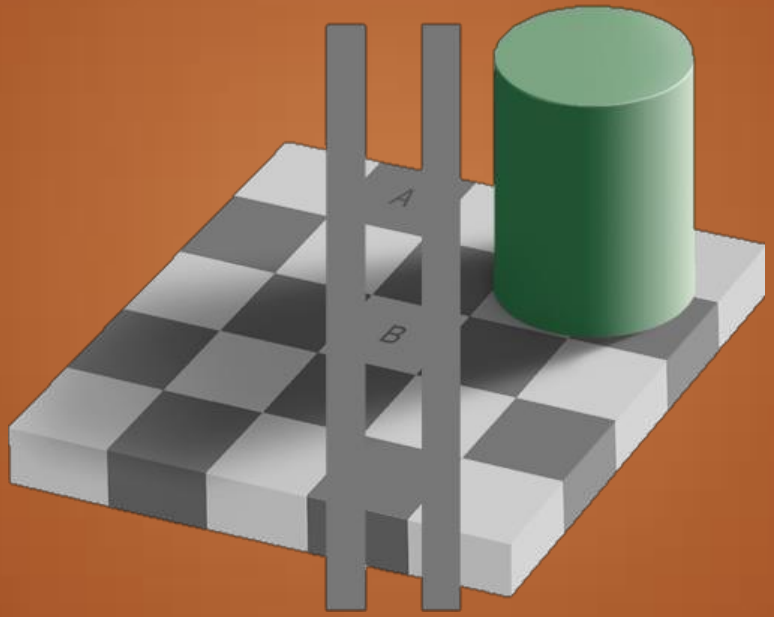


Edward Adelson



Edward Adelson

Really!



Edward Adelson

4. Interpretation is needed

- Most difficult thing for machines to deal with.
- We use accumulated learning and memory (called *a priori* knowledge
- Therefore with more machine learning capability, there has been great advancement in CV also.
- …but really we don't understand the recognition process

Visual Recognition Challenges: illumination

Visual Recognition Challenges: Deformation



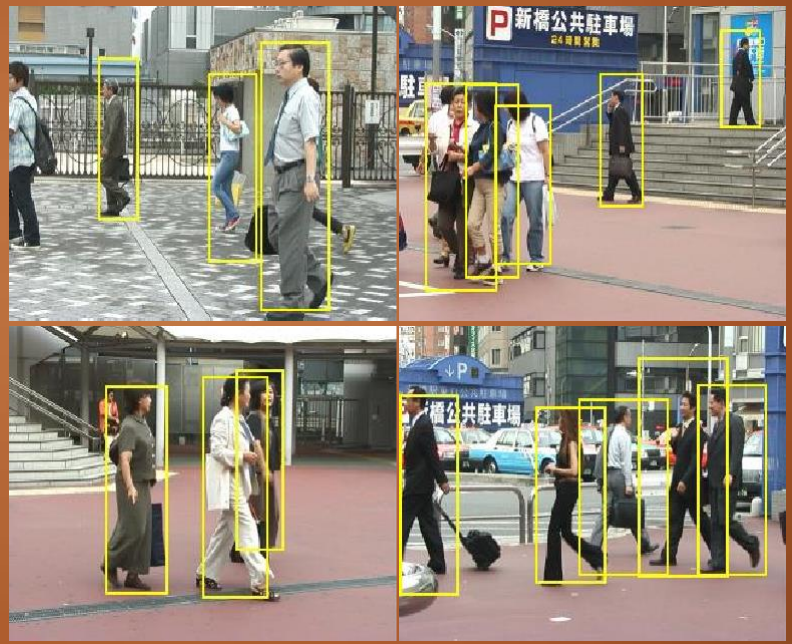

Visual Recognition Challenges: Occlusion

Visual Recognition Challenges: Intraclass variation

Challenges: Background clutter

Visual Recognition Challenges

Realistic scenes are crowded, cluttered, have overlapping objects.



An image classifier

```
def predict(image):
    # ????
    return class_label
```

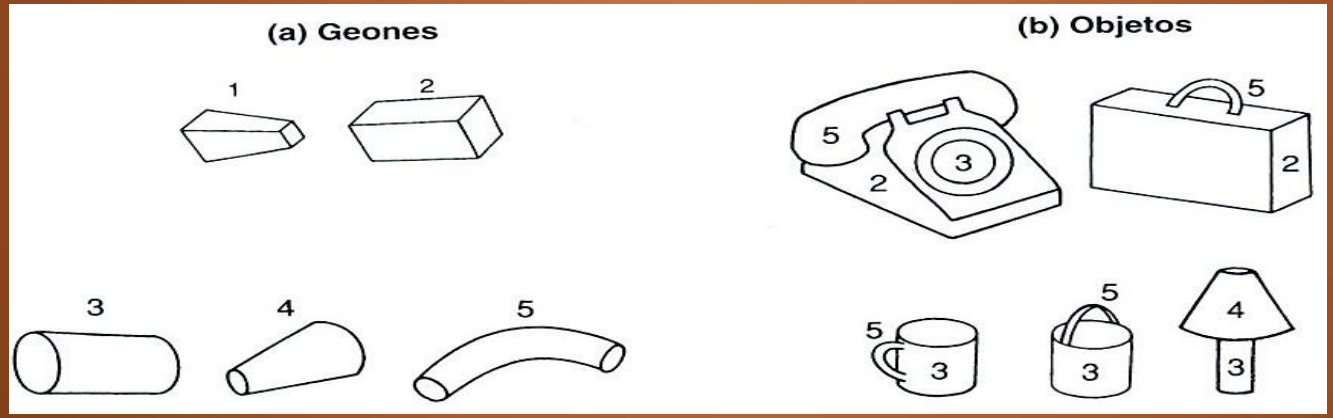
Unlike e.g. sorting a list of numbers,

no obvious way to hard-code the algorithm for recognizing a cat, or other classes.

Early Computer Vision

- Early computer vision methods tried to model the world, without using training data

(RBC – Recognition by Components)

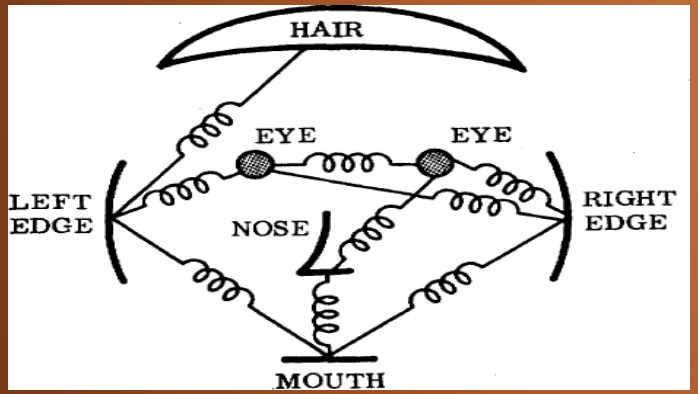


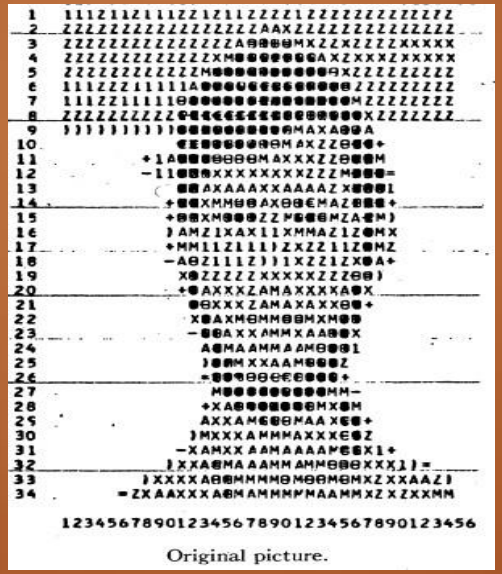
Biederman, I. (1987) Recognition-by-components: a theory of human image understanding. Psychol Rev. 1987;94(2):115-47.

Computer Vision

- Early computer vision methods tried to model the world, without using training data

(Part-Based Models)





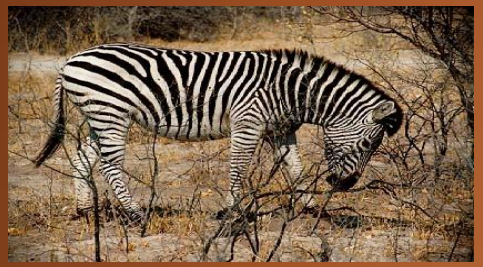
HAIR WAS LOCATED AT (7, 23)
L/EDGE WAS LOCATED AT (17, 13)
R/EDGE WAS LOCATED AT (17, 26)
L/EYE WAS LOCATED AT (14, 17)
R/EYE WAS LOCATED AT (14, 23)
NOSE WAS LOCATED AT (20, 20)
MOUTH WAS LOCATED AT (22, 20)

Fischler, M.A.; Elschlager, R.A. (1973). "The Representation and Matching of Pictorial Structures". IEEE Transactions on Computers: 67.

The need for machine learning

- Methods that use training data quickly outperformed modelling approaches (1990+).
- Machine learning is now a core part of computer vision.
- Nearly every machine learning algorithm has been used in one way or another in computer vision.
- Visual data (images and videos) is a new source for machine learning scientists.

Object categorization: statistical viewpoint



$$p(zebra \mid image)$$

vs.

$$p(no\ zebra \mid image)$$

Bayes rule:

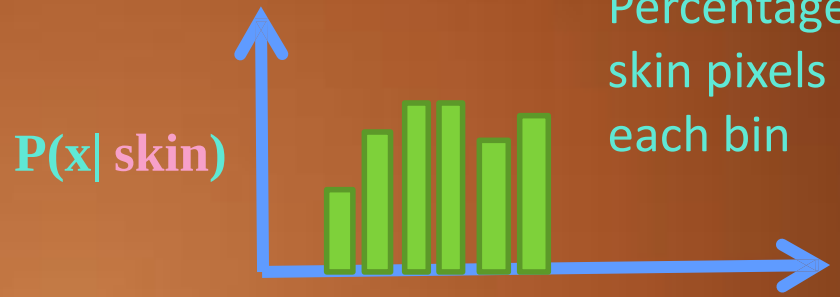
$$\underbrace{\frac{p(zebra \mid image)}{p(no\ zebra \mid image)}}_{\text{posterior ratio}} = \underbrace{\frac{p(image \mid zebra)}{p(image \mid no\ zebra)}}_{\text{likelihood ratio}} \cdot \underbrace{\frac{p(zebra)}{p(no\ zebra)}}_{\text{prior ratio}}$$

Object categorization: statistical viewpoint

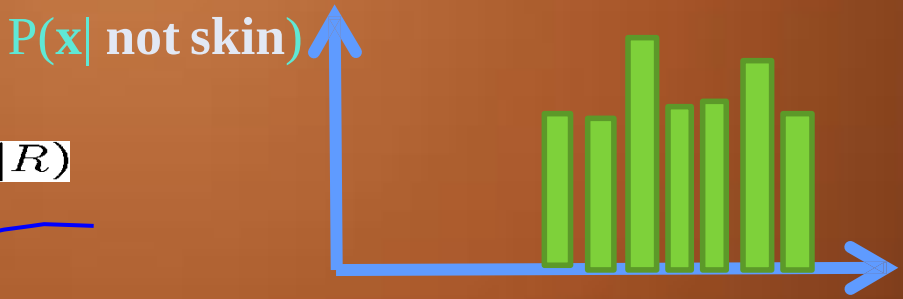
$$\underbrace{\frac{p(zebra \mid image)}{p(no\ zebra \mid image)}}_{\text{posterior ratio}} = \underbrace{\frac{p(image \mid zebra)}{p(image \mid no\ zebra)}}_{\text{likelihood ratio}} \cdot \underbrace{\frac{p(zebra)}{p(no\ zebra)}}_{\text{prior ratio}}$$

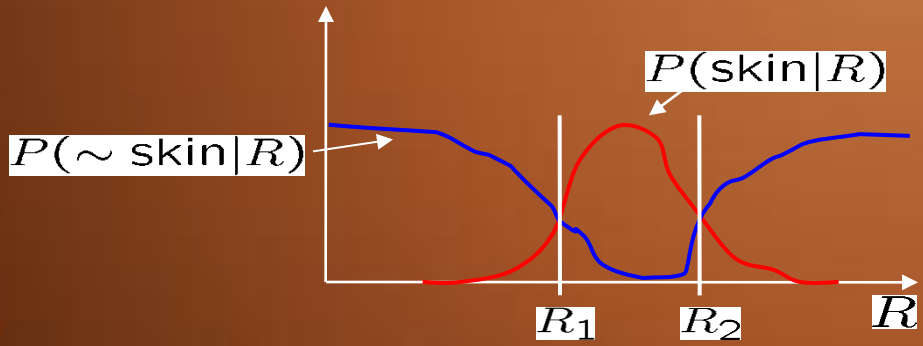
- **Discriminative methods model posterior**
- **Generative methods model likelihood and prior**

Generative



Feature x = Hue



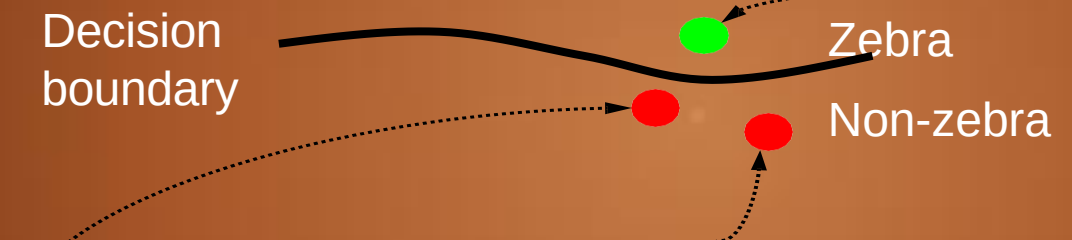


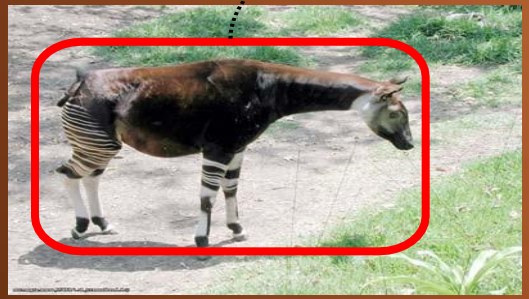
Discriminative

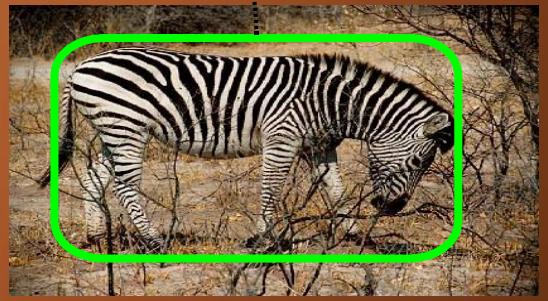
Modelling of

$$\frac{p(zebra \mid image)}{p(no\ zebra \mid image)}$$

- Direct construct a good decision boundary







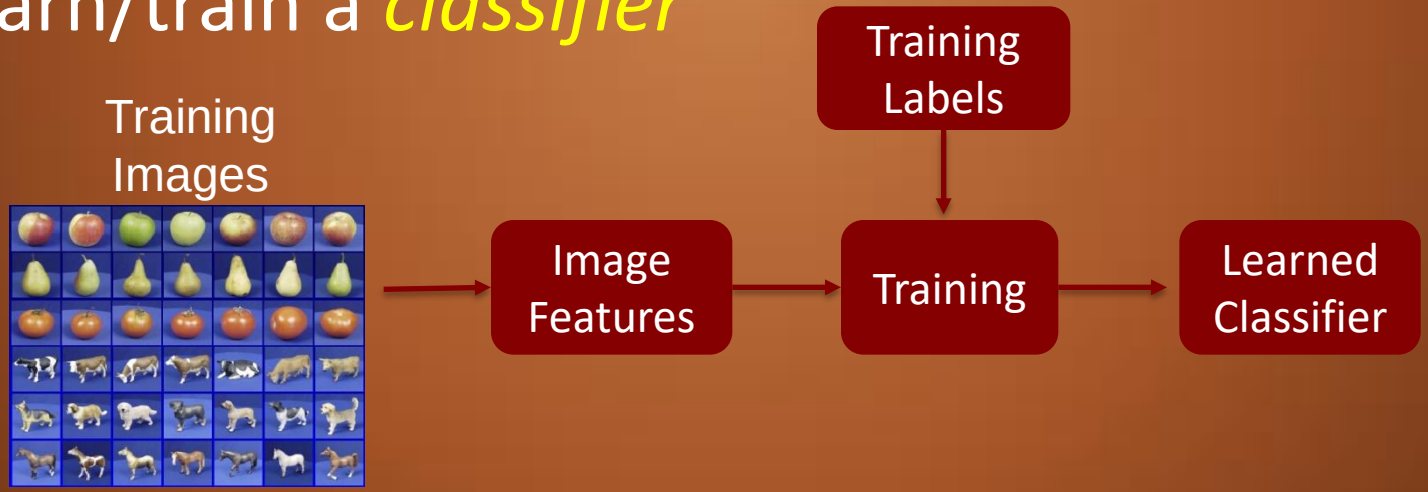
Three main issues

- Representation/Features
  - How to represent an object category
- Learning/Training
  - How to form the classifier, given training data
- Recognition/Testing
  - How the classifier is to be used on novel data

Basic Recognition framework

Train

- Feature *representation - to describe training instances/objects (here images)*
- Learn/train a *classifier*



Basic Recognition framework

Test

- Generate candidates in new image
- *Score* the candidates

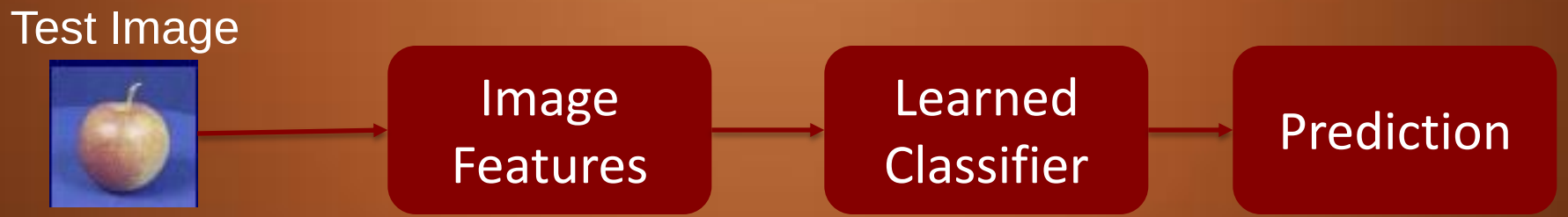
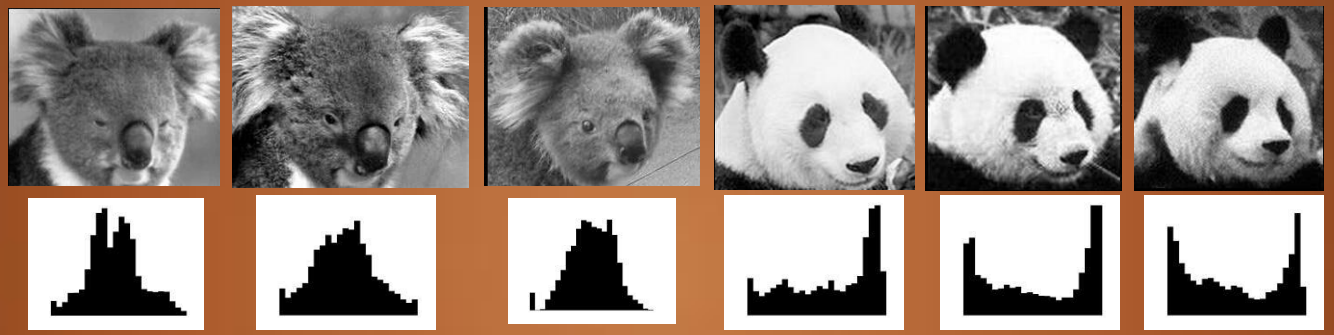


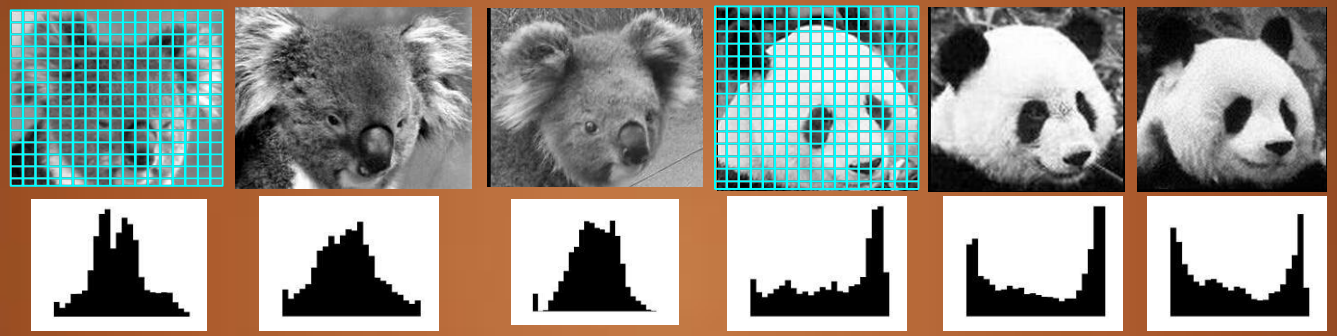
Image Features: Window based



Simple holistic descriptions of image content

- grayscale / color histogram

Image Features: Window based



Simple holistic descriptions of image content

- grayscale / color histogram
- vector of pixel intensities

Image Features: Window based

- Pixel-based representations sensitive to small shifts

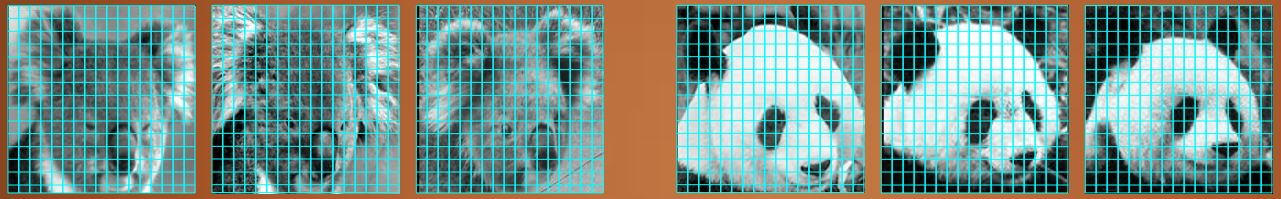
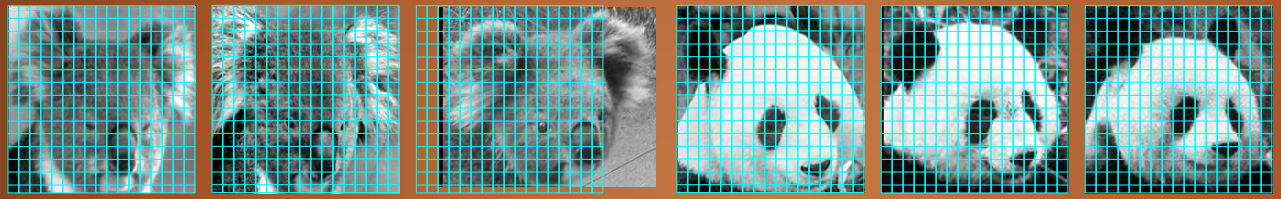


Image Features: Window based

- Pixel-based representations sensitive to small shifts



- Color/grayscale histogram based description can be sensitive to illumination and intra-class appearance variation

Image Features: Window based

- Consider edges, contours, and (oriented) intensity gradients

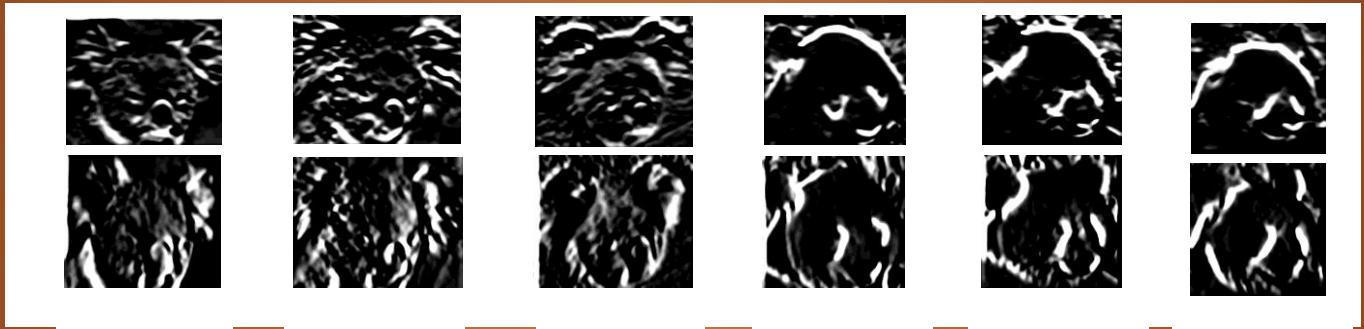
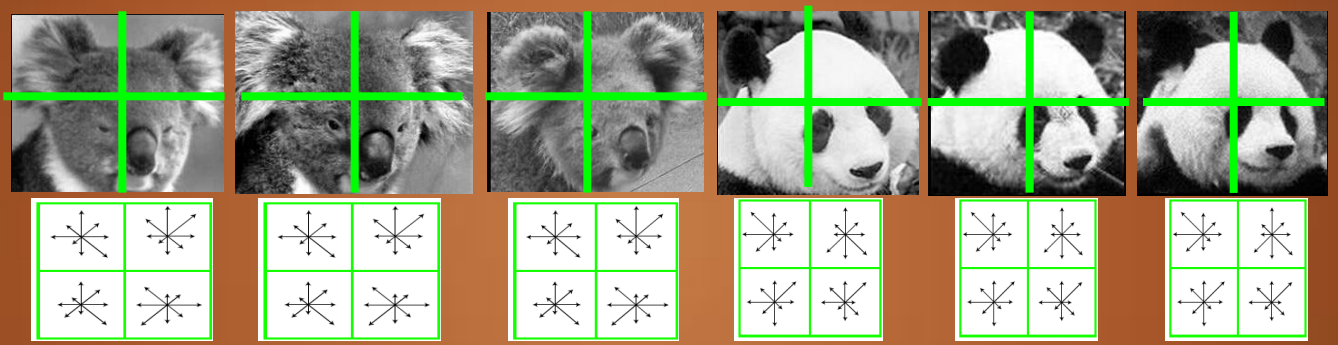


Image Features: Window based

- Consider edges, contours, and (oriented) intensity gradients



- Summarize local distribution of gradients with histogram
  - Locally orderless: offers invariance to small shifts and rotations
  - Contrast-normalization: try to correct for variable illumination

Generic category recognition: basic framework

Train

- Feature *representation - to describe training instances/objects (here images)*
- Learn/train a *classifier*

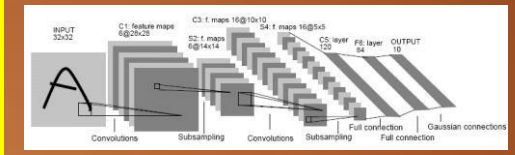
Discriminative classifier construction

Nearest neighbor

10^6 examples

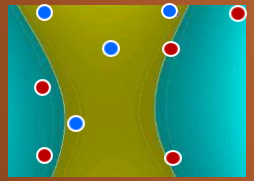
Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005 ...

Neural networks



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

SVMs

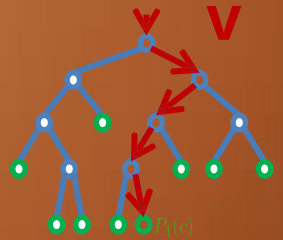


Guyon, Vapnik, Heisele, Serre, Poggio, 2001, …

Boosting

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

Random Forests



Breiman, 1984
Shotton, et al CVPR 2008

Slide adapted from Antonio Torralba

Generic category recognition: basic framework

Test

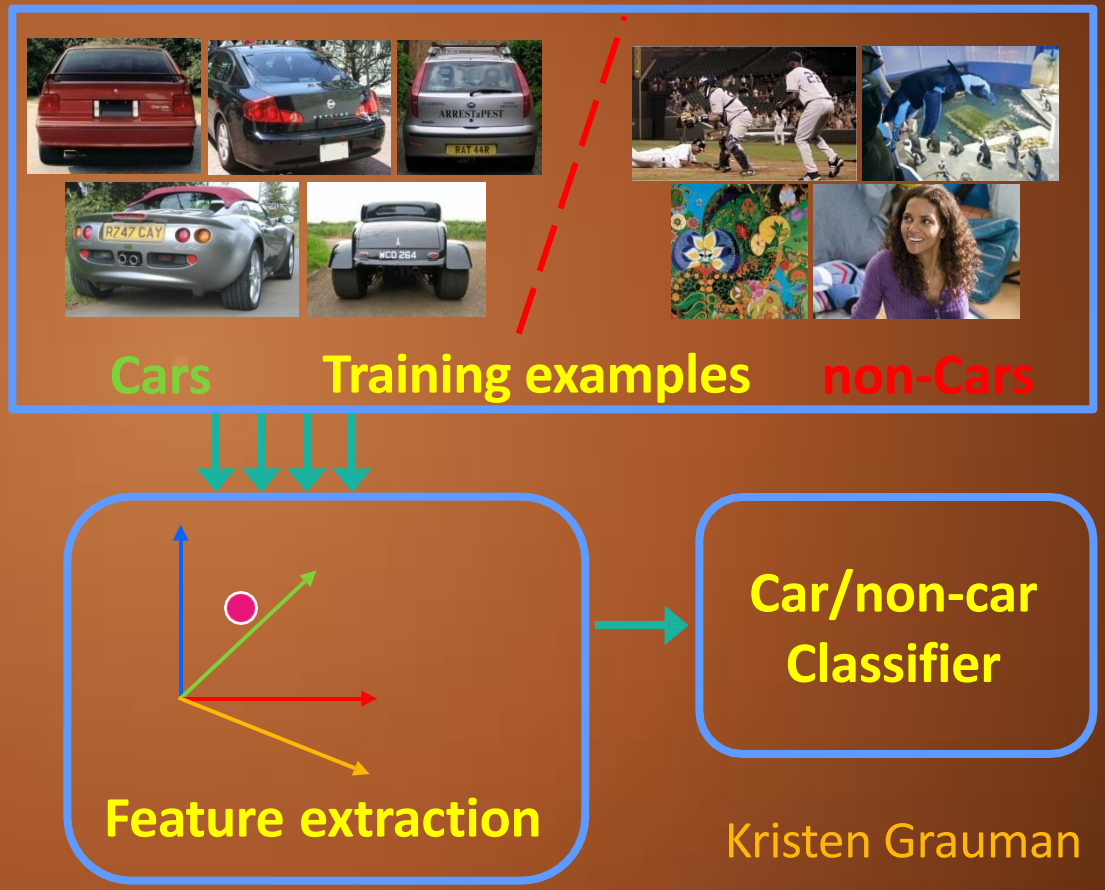
- Generate candidates in new image
- *Score* the candidates

Window-based models: Generating object proposals and scoring candidates

Window-based object detection: Recap

Training:

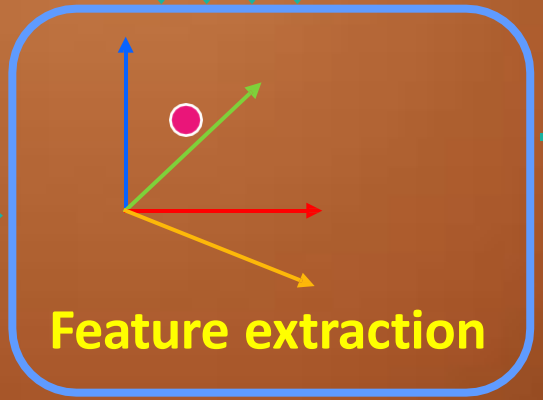
1. Obtain training data
2. Define features
3. Train classifier



Window-based object detection: Recap

Given new image:

1. Slide window
2. Score by classifier



Car/non-car Classifier