CS 1674: Intro to Computer Vision

Introduction

Prof. Adriana Kovashka
University of Pittsburgh
August 28, 2018

Course Info

- **Course website:** http://people.cs.pitt.edu/~kovashka/cs1674_fa18
- **Instructor**: Adriana Kovashka (kovashka@cs.pitt.edu)
- **Office:** Sennott Square 5325
- **Class:** Tue/Thu, 4pm-5:15pm
- **Office hours**: Tue 2-3:55pm, Thu 1-3:55pm

About the Instructor

Born 1985 in Sofia, Bulgaria

Got BA in 2008 at Pomona College, CA (Computer Science & Media Studies)

Got PhD in 2014 at University of Texas at Austin (Computer Vision)

About the TA

- Narges Honarvar Nazari
- **Office:** Sennott Square 5501
- **Office hours:** TBD
  - Do this Doodle by the end of Friday: https://doodle.com/poll/gcmqgi7y6uymetu3

Course Goals

- To learn the basics of low-level image analysis
- To learn about some classic and modern approaches to high-level computer vision tasks (e.g. object recognition)
- To get experience with vision techniques
- To learn/apply basic machine learning (a key component of modern computer vision)
- To think critically about vision approaches, and to see connections between works

Textbooks

- Computer Vision: Algorithms and Applications by Richard Szeliski
- Visual Object Recognition by Kristen Grauman and Bastian Leibe
- More resources available on course webpage

- Your notes from class are your best study material, slides are *not* complete with notes

Programming Language

- We’ll use Matlab
- It can be downloaded for free from MyPitt
- We’ll do a short tutorial; ask TA if you need further help

Course Structure

- Lectures
- Weekly assignments
- Two exams
- Participation component

Policies and Schedule

http://people.cs.pitt.edu/~kovashka/cs1674_fa18

Types of computer vision

- Lower-level vision
  - Analyzing textures, edges and gradients in images, without concern for the semantics (e.g. objects) of the image
- Higher-level vision
  - Making predictions about the semantics or higher-level functions of content in images (e.g. objects, actions, etc.)
  - Involves machine learning

Warnings

Warning #1

- This class is **a lot of work**
- I've opted for shorter, more manageable HW assignments, but there is a lot of them
- I expect you'd be spending **6-8 hours** on homework each week

- … But you get to understand algorithms and concepts in detail!

Warning #2

- Some parts will be **hard** and require that you pay close attention!
- I will also pick on students randomly to answer questions
- **Use instructor's and TA's office hours!!!**

- … You will learn a lot!

Questions?

Plan for Today

- Blitz introductions
- What is computer vision?
  - Why do we care?
  - What are the challenges?
  - What is recent research like?
- Overview of topics
- Review and tutorial
  - Linear algebra
  - Matlab

Blitz introductions (10 sec)

- What is your name?
- What one thing outside of school are you passionate about?
- What do you hope to get out of this class?

- **Every time you speak, please remind me your name**

Computer Vision

What is computer vision?

Done?

"We see with our brains, not with our eyes“ (Oliver Sacks and others)

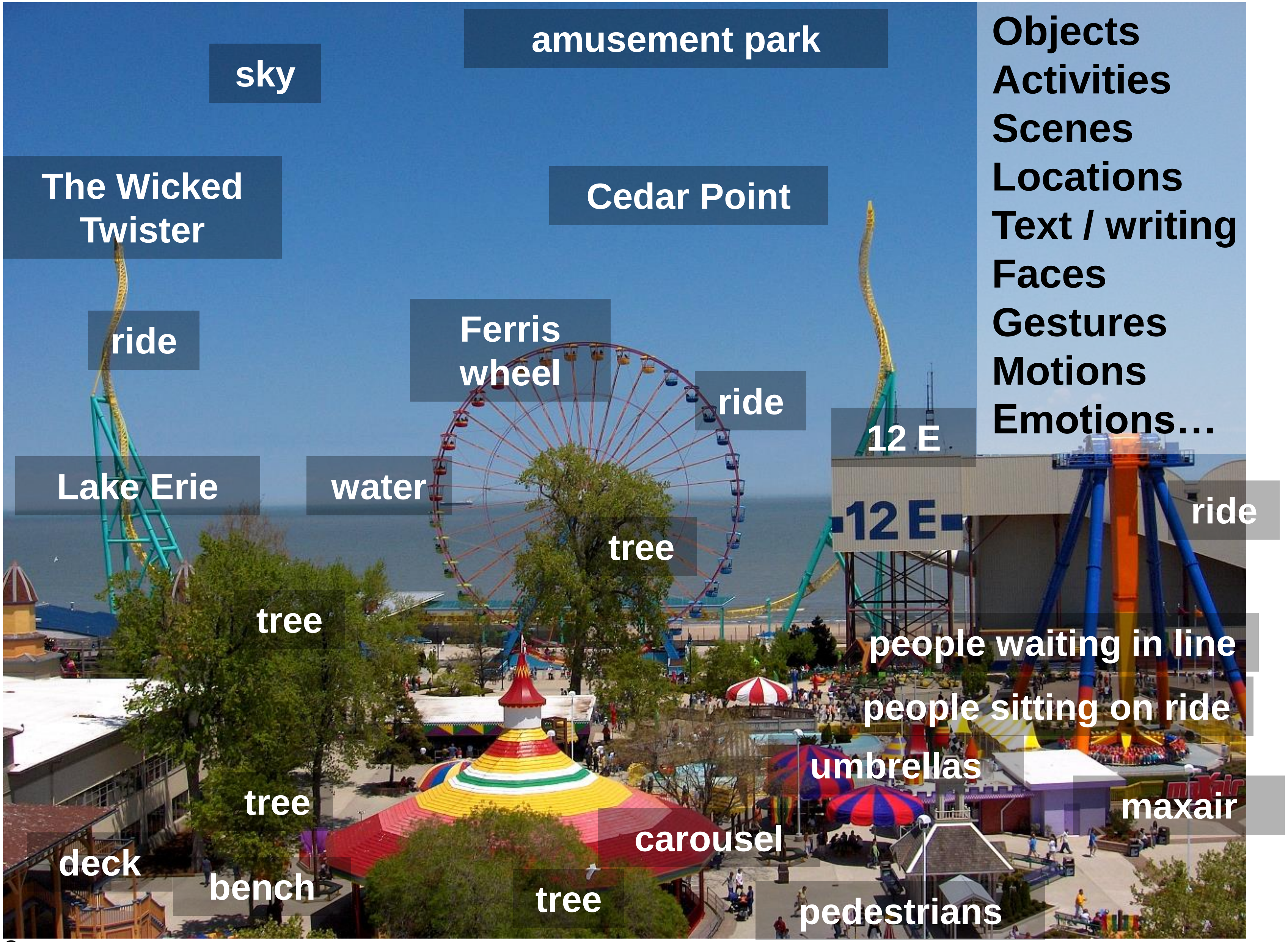
Kristen Grauman (adapted)

What is computer vision?

- Automatic understanding of images and video
  - Algorithms and representations to allow a machine to recognize objects, people, scenes, and activities
  - Algorithms to mine, search, and interact with visual data
  - Computing properties and navigating within the 3D world using visual data
  - Generating realistic synthetic visual data

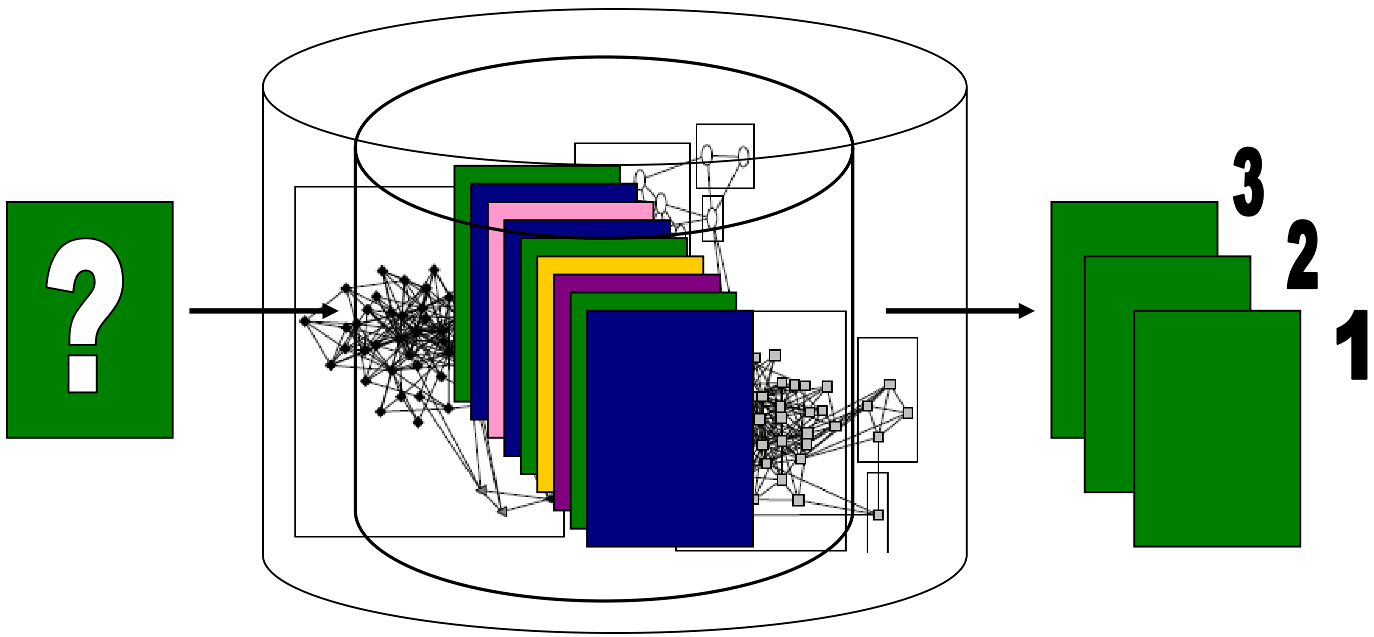
Adapted from Kristen Grauman

Perception and interpretation



Kristen Grauman

Visual search, organization



Query **Image or video archives** **Relevant content**

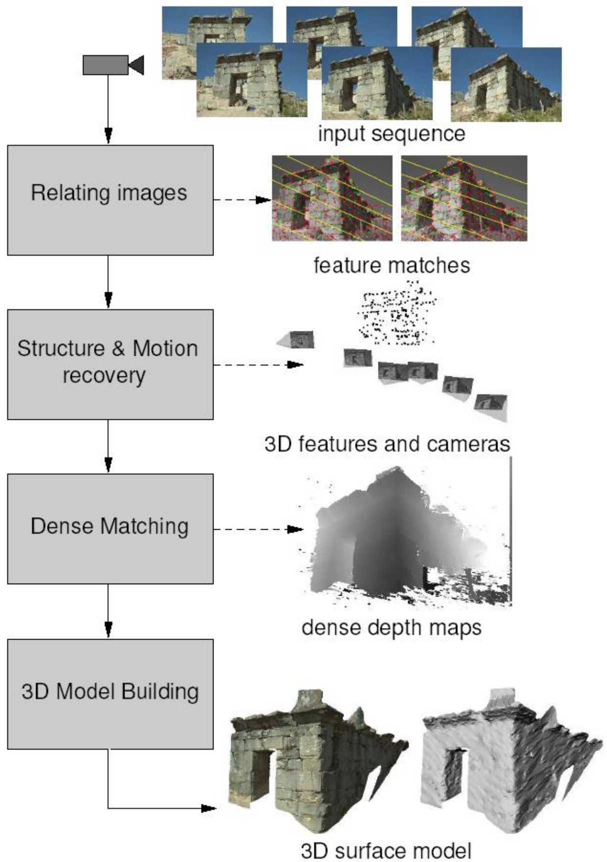
Kristen Grauman

Measurement

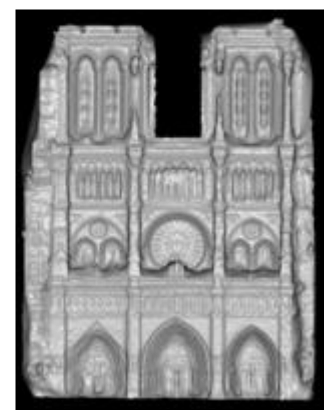
Real-time stereo

Pollefeys et al.

Structure from motion



Multi-view stereo for community photo collections



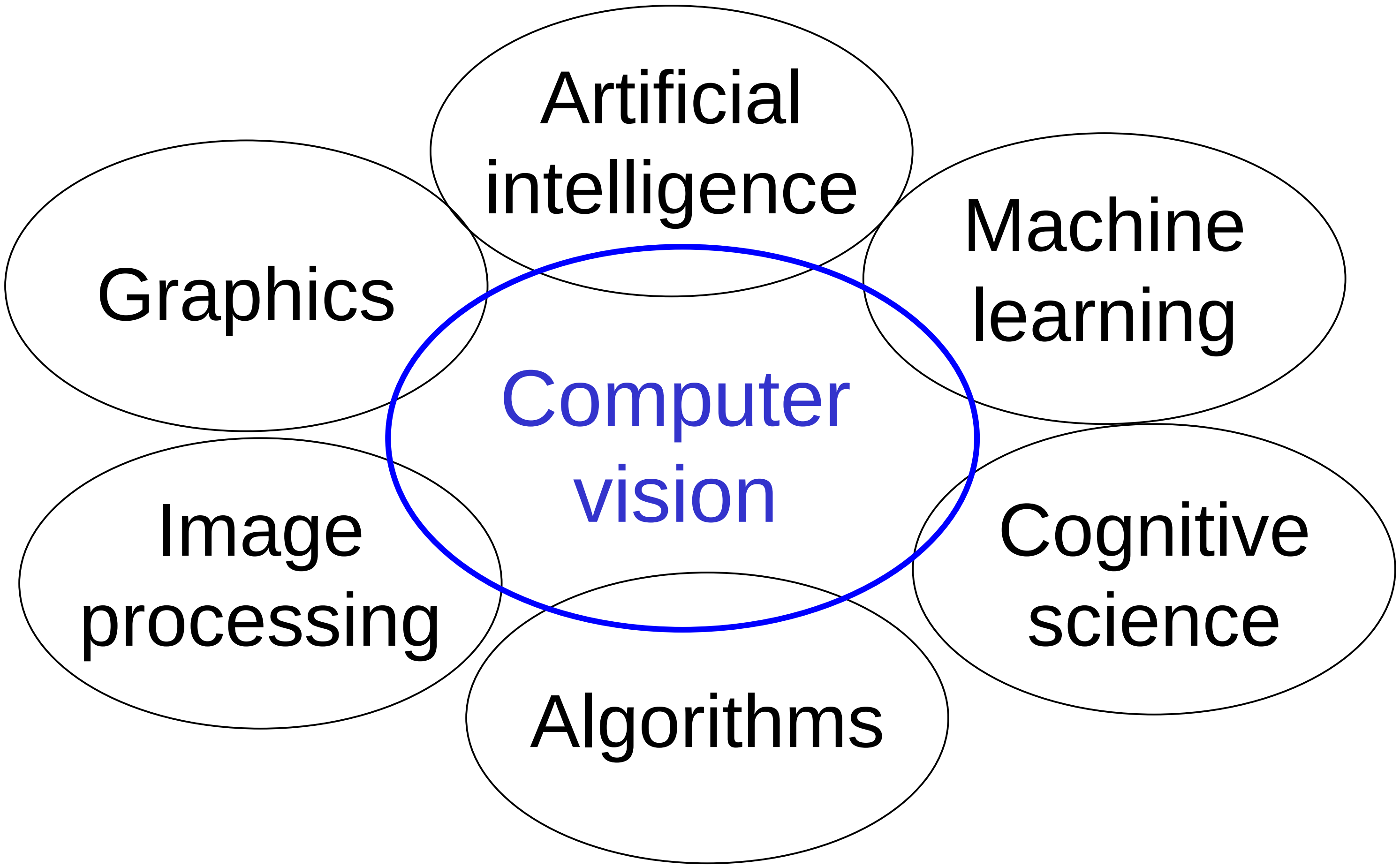
Goesele et al.

Slide credit: L. Lazebnik

Generation

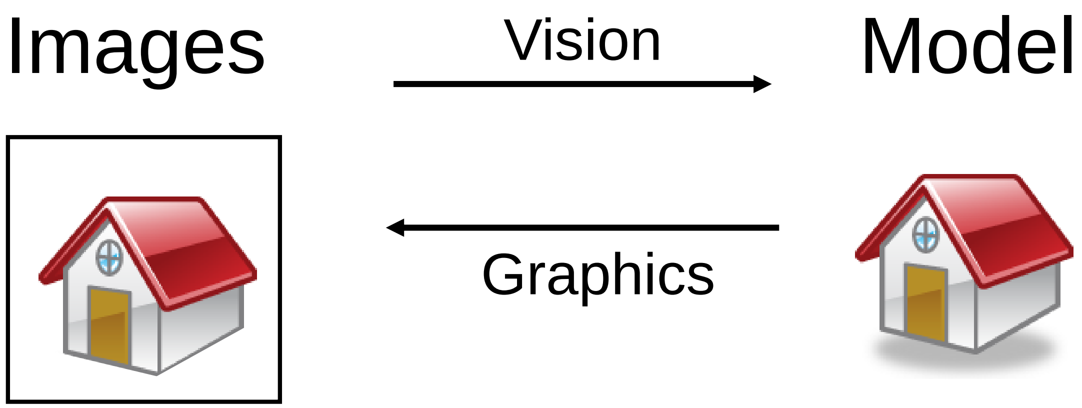
Karras et al., "Progressive Growing of GANs for Improved Quality, Stability, and Variation", ICLR 2018

Related disciplines



Kristen Grauman

Vision and graphics



Inverse problems: analysis and synthesis.

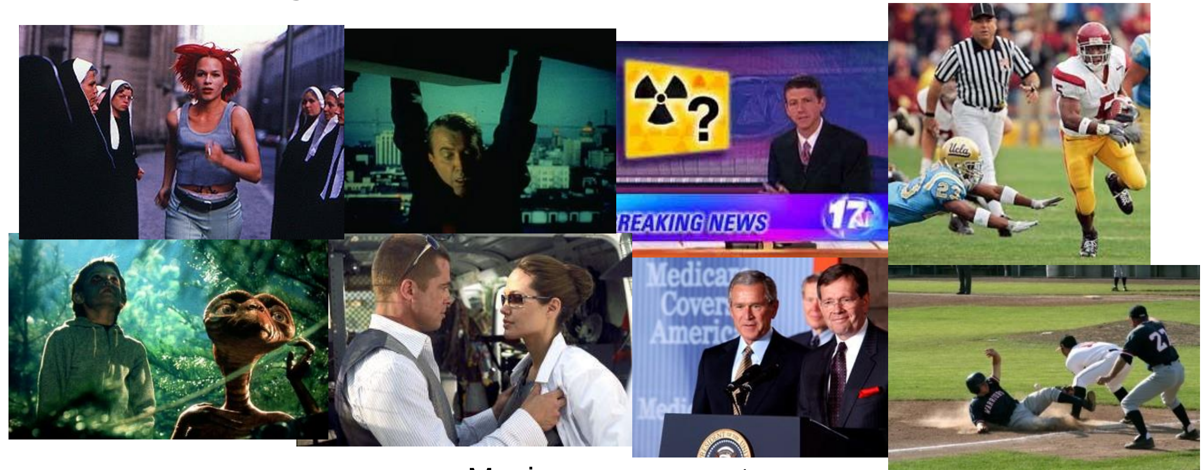
Kristen Grauman

Why vision?

144k hours uploaded to YouTube daily
4.5 mil photos uploaded to Flickr daily
10 bil images indexed by Google

- Images and video are everywhere!

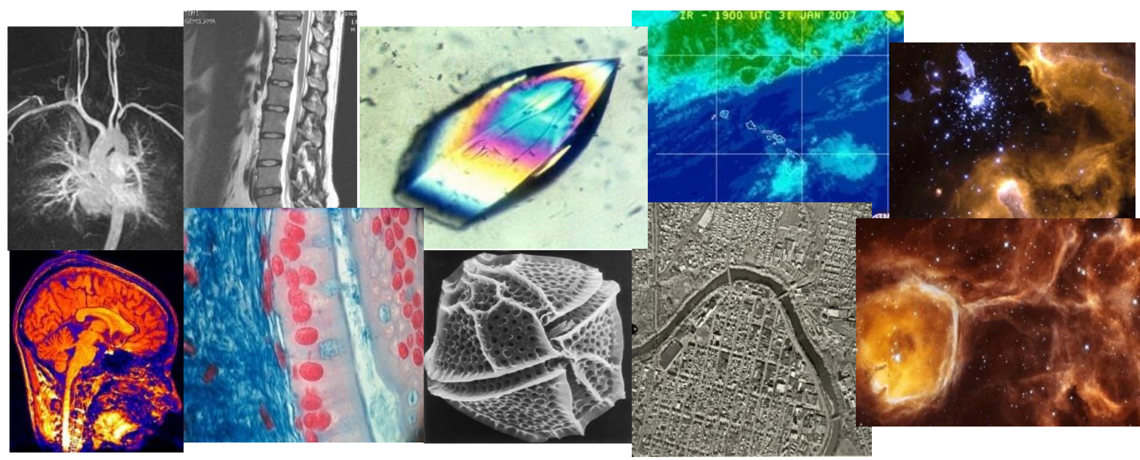
Personal photo albums



Movies, news, sports

Instagram flickr shutterstock YouTube gettyimages

Surveillance and security



Medical and scientific images

Adapted from Lana Lazebnik

Why vision?

- As image sources multiply, so do applications
  - Relieve humans of boring, easy tasks
  - Perception for robotics / autonomous agents
  - Organize and give access to visual content
  - Description of image content for the visually impaired
  - Human-computer interaction
  - Fun applications (e.g. transfer art styles to my photos)

Adapted from Kristen Grauman

Things that work well

Faces and digital cameras

Camera waits for everyone to smile to take a photo [Canon]

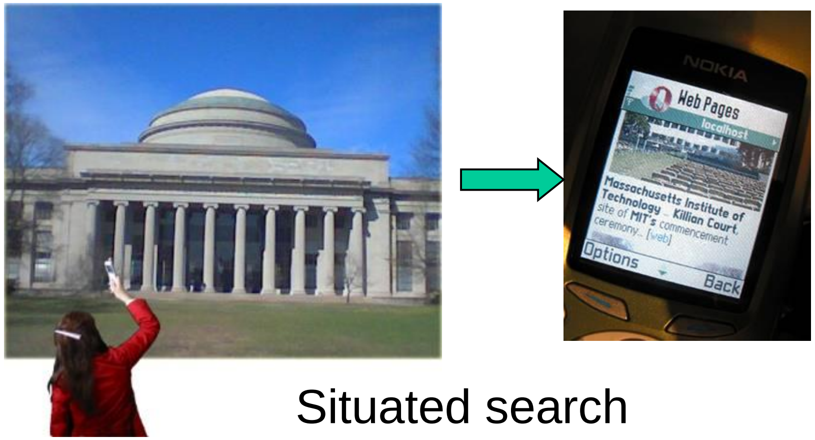
Setting camera focus via face detection

Kristen Grauman

Face recognition

Devi Parikh

Linking to info with a mobile device



Situated search
Yeh et al., MIT

MSR Lincoln

kooaba

Exploring photo collections

Snavely et al.

Interactive systems

Shotton et al.

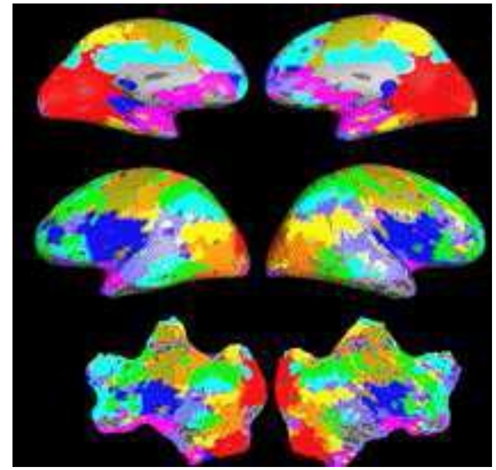
Video-based interfaces

YouTube Link

Human joystick
NewsBreaker Live

Assistive technology systems
Camera Mouse
Boston College

Vision for medical & neuroimages

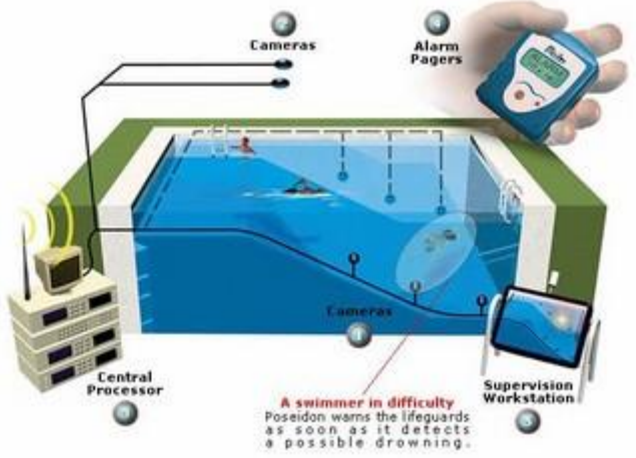


fMRI data
Golland et al.

Image guided surgery
MIT AI Vision Group

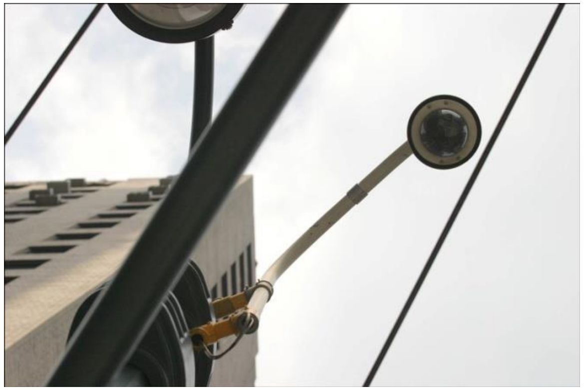
Safety & security

Navigation,
driver safety



Monitoring pool
(Poseidon)

Pedestrian detection
MERL, Viola et al.



Surveillance

Kristen Grauman

Things that need more work

The latest at CVPR, ICCV, ECCV

CVPR = IEEE/CVF Conference on Computer Vision and Pattern Recognition
ICCV = IEEE/CVF International Conference on Computer Vision
ECCV = European Conference on Computer Vision

Accurate object detection in real time

	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
Fast R-CNN	70.0	.5 FPS	2 s/img
Faster R-CNN	73.2	7 FPS	140 ms/img
YOLO	69.0	45 FPS	22 ms/img

2 feet

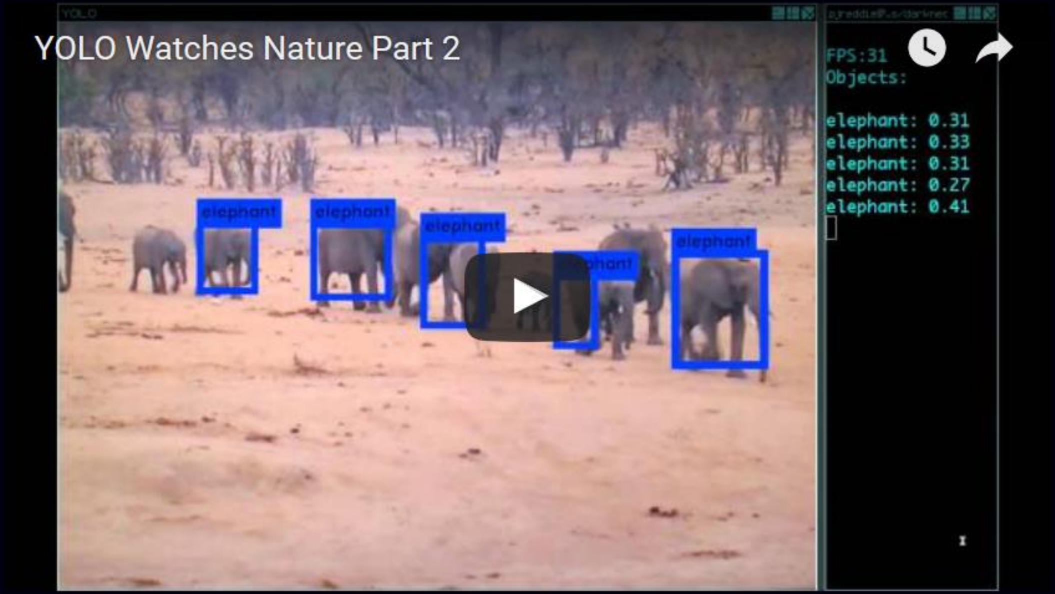
Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016

YOLO: Real-Time Object D...

pjreddie.com/darknet/yolo/

Our ability to detect objects has gone
from 34 mAP in 2008
to 73 mAP at 7 FPS (frames per second)
or 63 mAP at 45 FPS
in 2016

YOLO: Real-Time Object Detection



You only look once (YOLO) is a system for detecting objects on the Pascal VOC 2012 dataset. It can detect the 20 Pascal object classes:

- person
- bird, cat, cow, dog, horse, sheep
- aeroplane, bicycle, boat, bus, car, motorbike, train
- bottle, chair, dining table, potted plant, sofa, tv/monitor

Recognition in novel modalities

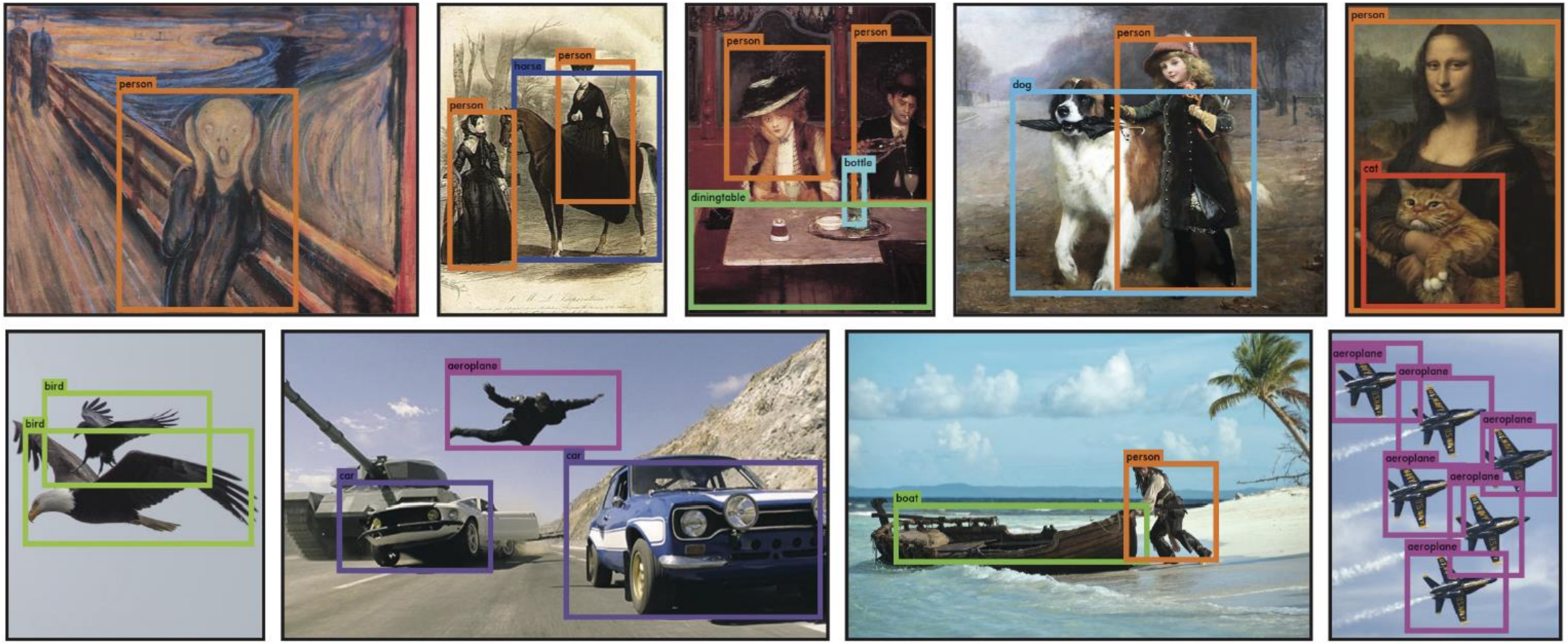
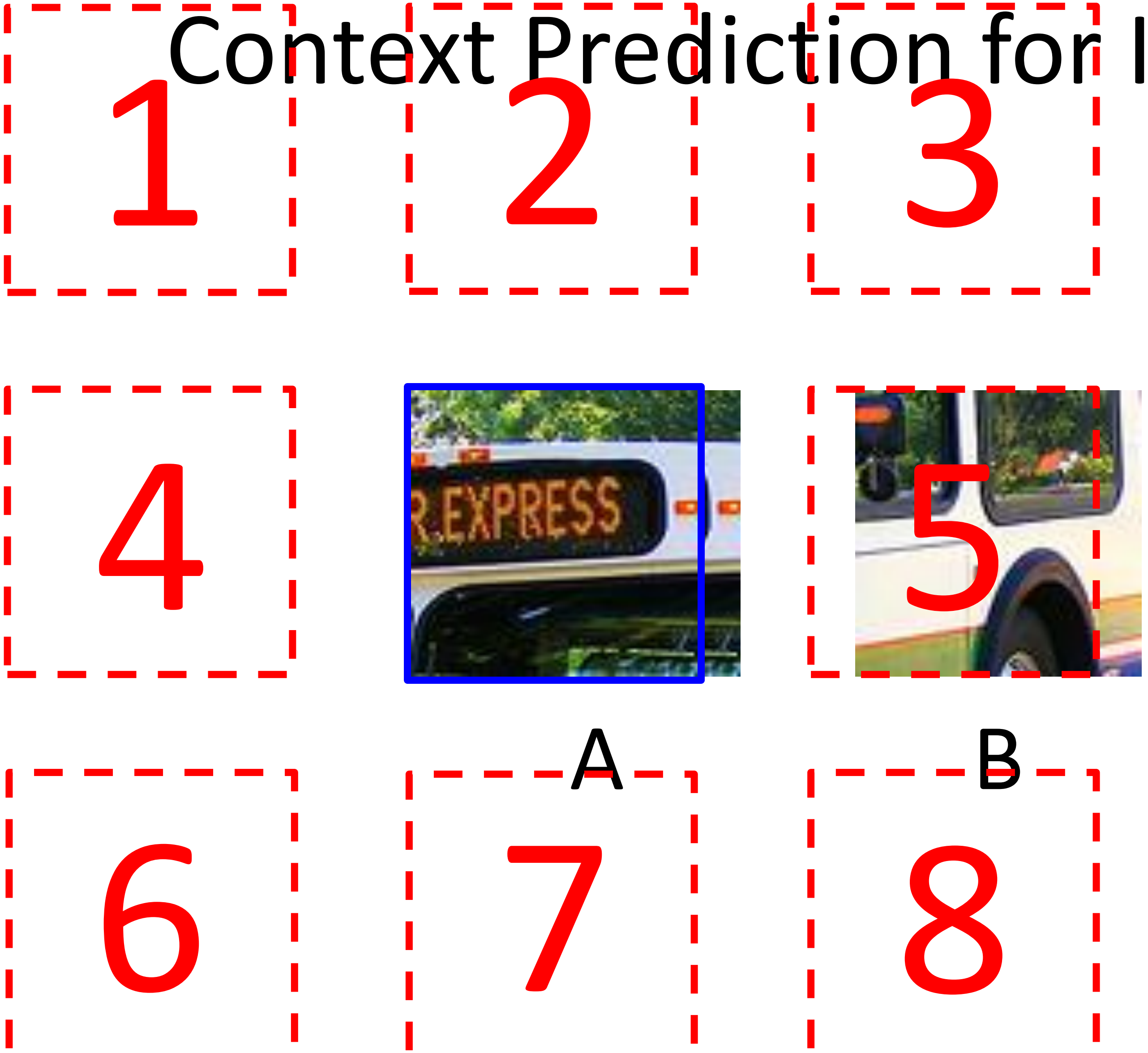


Figure 6: Qualitative Results. YOLO running on sample artwork and natural images from the internet. It is mostly accurate although it does think one person is an airplane.

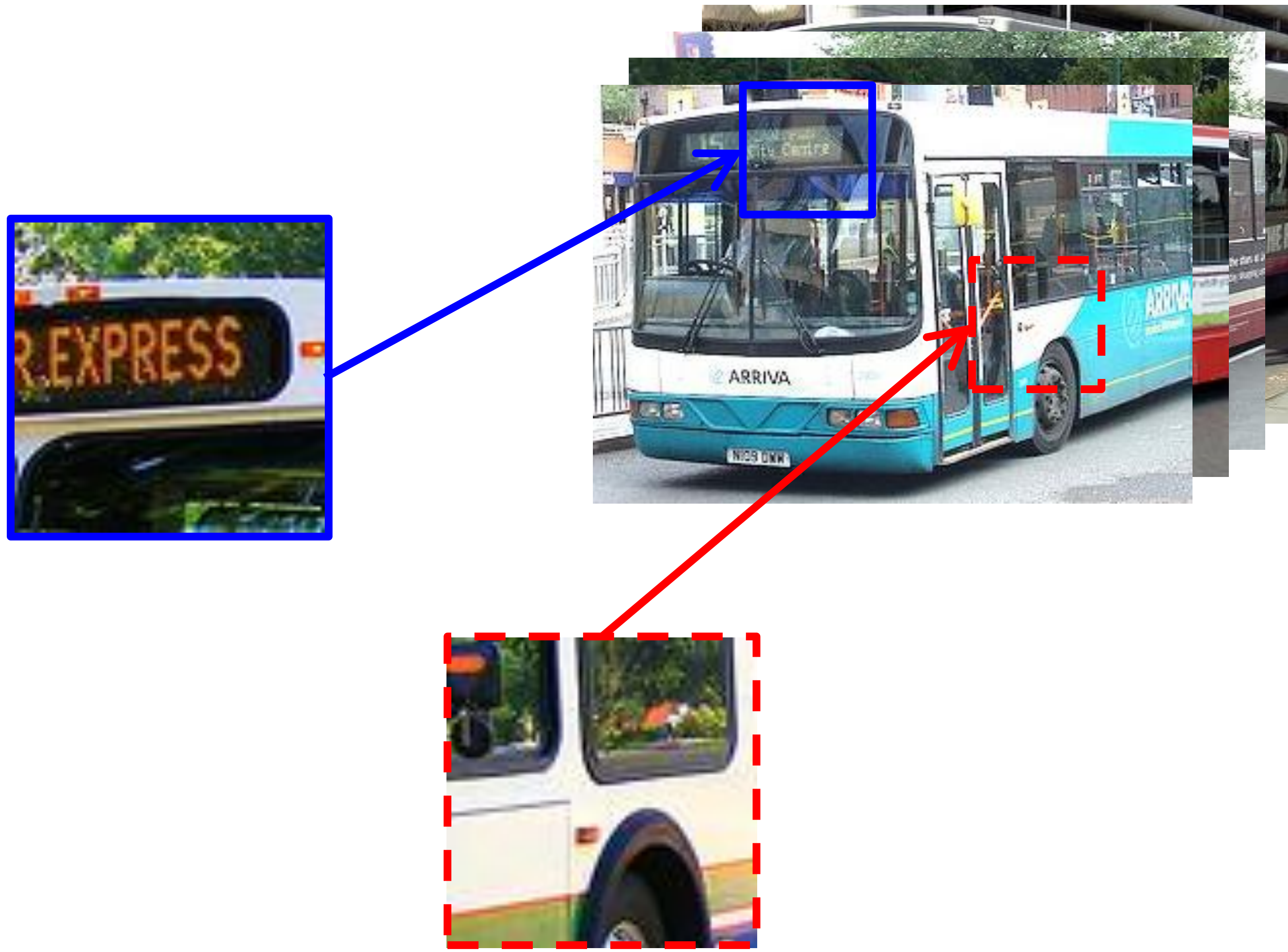
Redmon et al., CVPR 2016

Context Prediction for Images



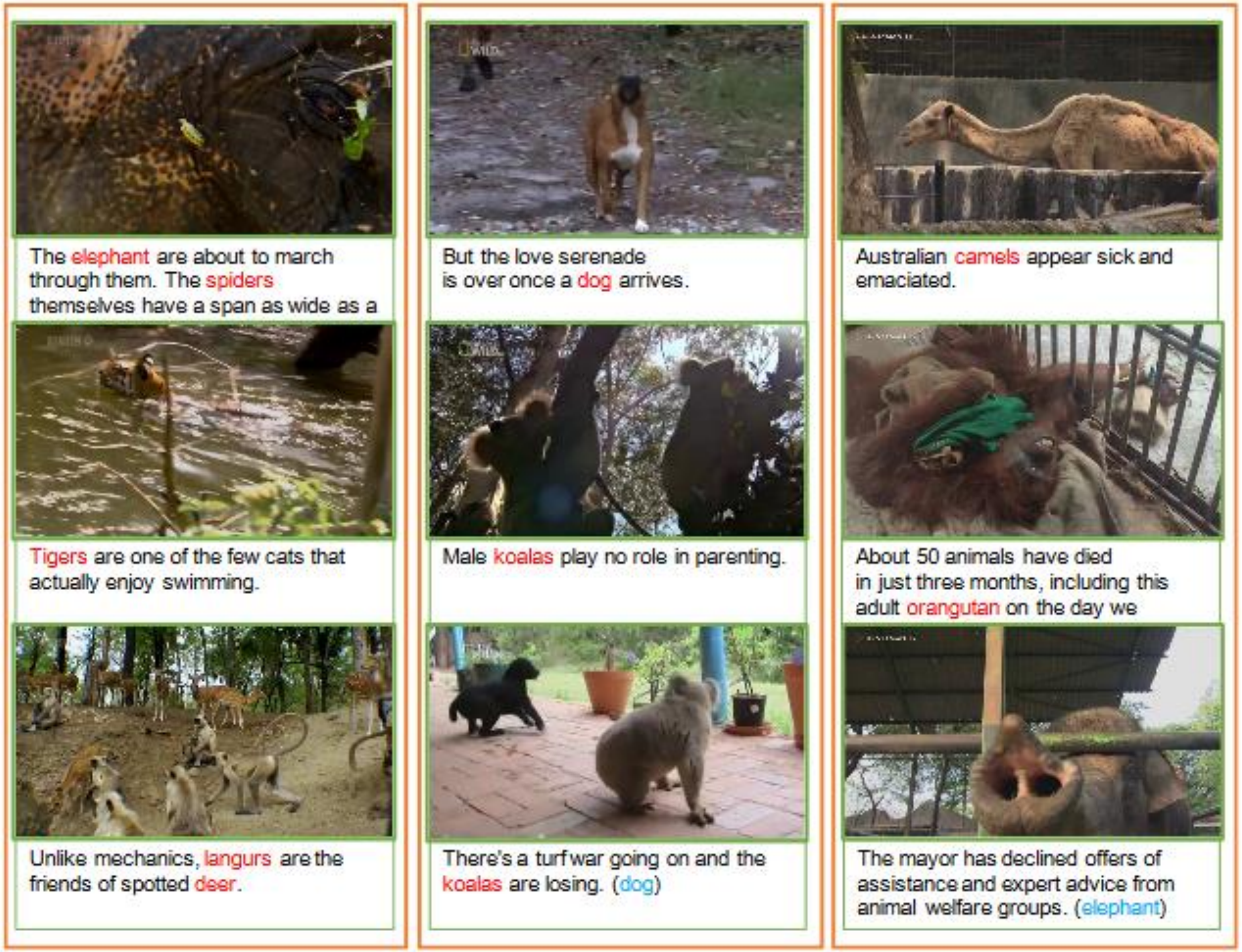
Doersch et al., ICCV 2015

Semantics from a non-semantic task



Doersch et al., ICCV 2015

Discover and Learn New Objects from Documentaries



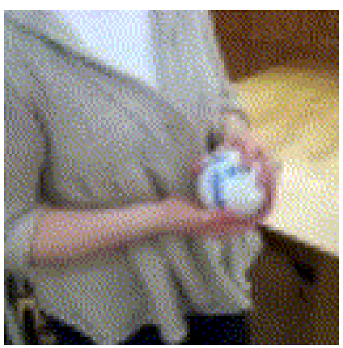
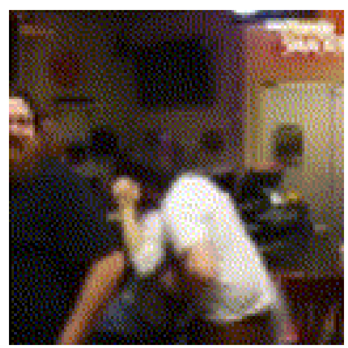
Chen et al., CVPR 2017

Social Scene Understanding: End-To-End Multi-Person Action Localization and Collective Activity Recognition

Bagautdinov et al., CVPR 2017

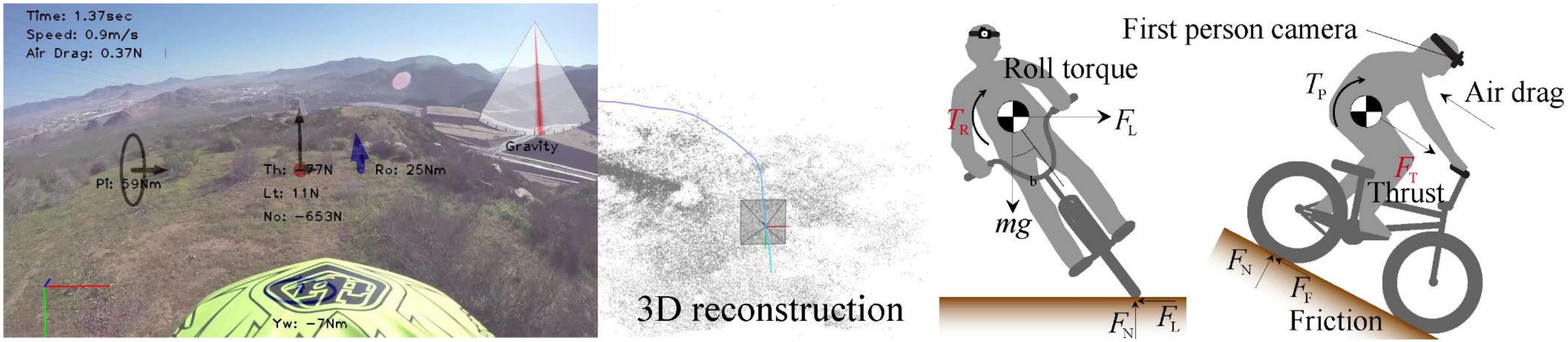
Generating the Future with Adversarial Transformers




Vondrick and Torralba, CVPR 2017

Force from Motion:
Decoding Physical Sensation from a First Person Video

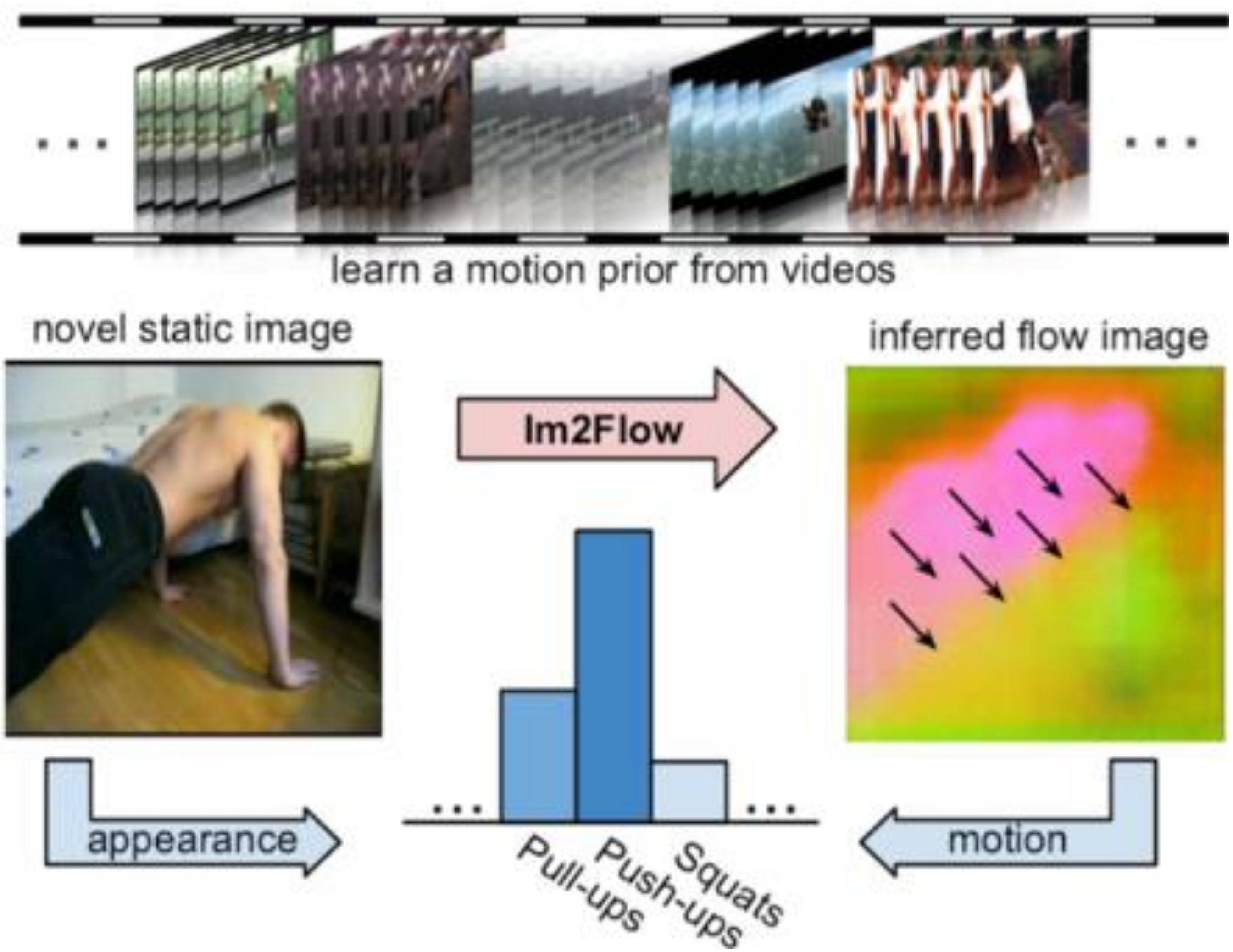


Park et al., CVPR 2016

Assessing the Quality of Actions

Pirsiavash et al., ECCV 2014

Im2Flow: Motion Hallucination from Static Images for Action Recognition



Gao et al., CVPR 2018

Image generation

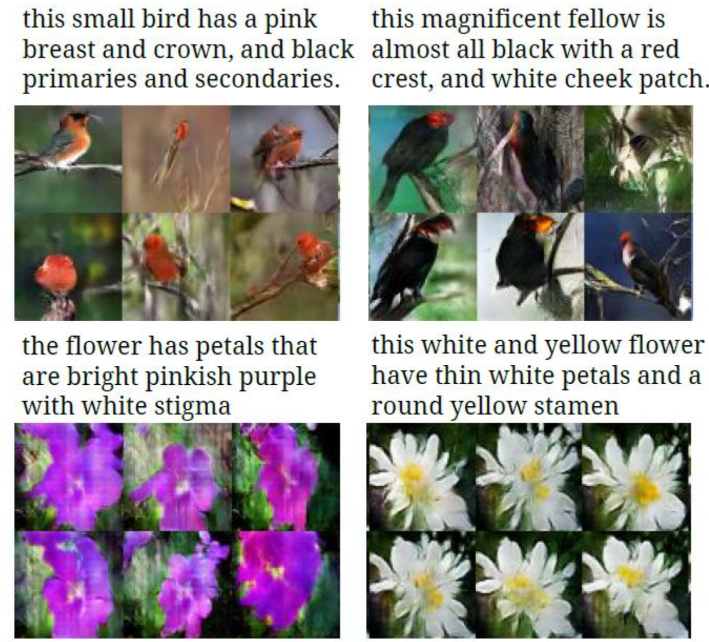


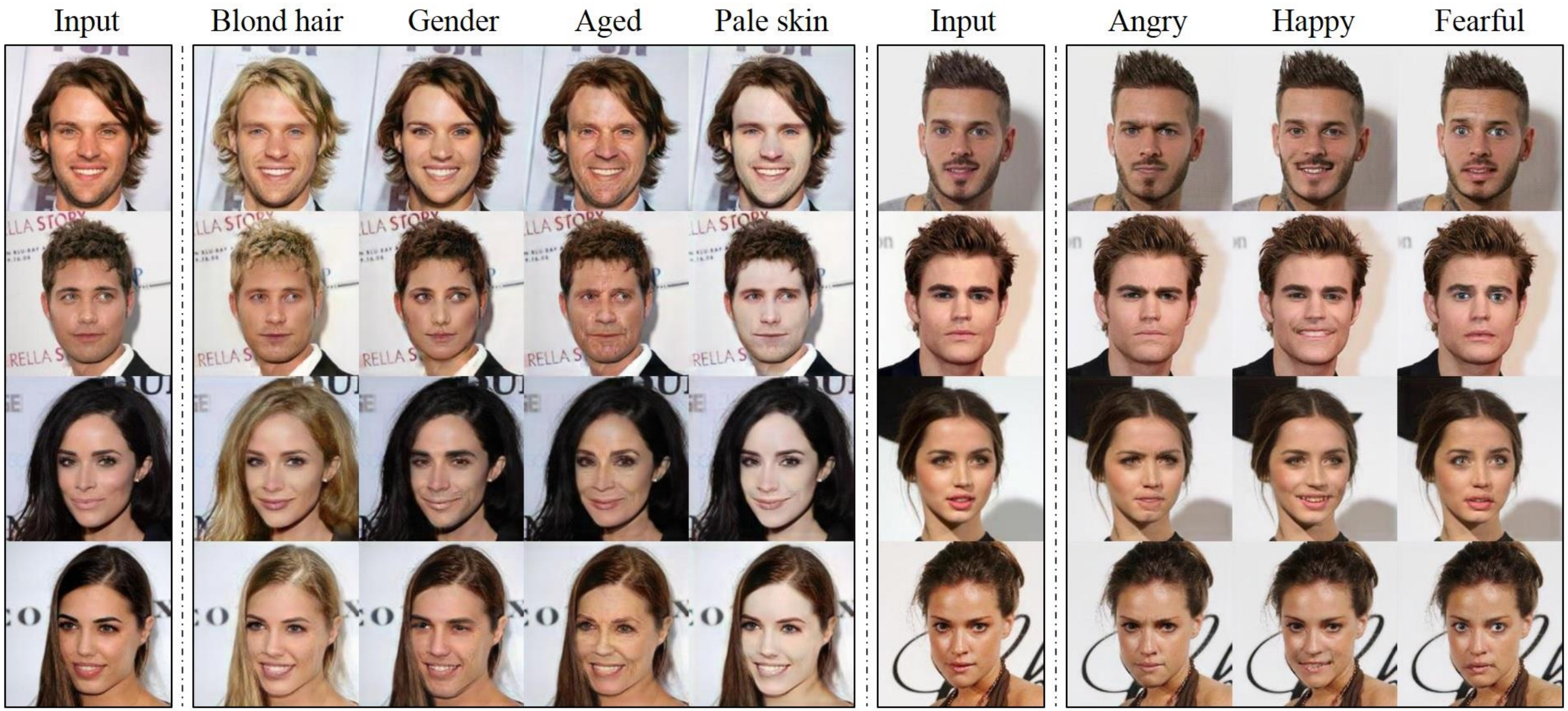
Figure 1. Examples of generated images from text descriptions. Left: captions are from zero-shot (held out) categories. Right: captions are from training set categories.

Reed et al., ICML 2016

Figure 3: Generated bedrooms after five epochs of training. There appears to be evidence of visual under-fitting via repeated noise textures across multiple samples such as the base boards of some of the beds.

Radford et al., ICLR 2016

StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation



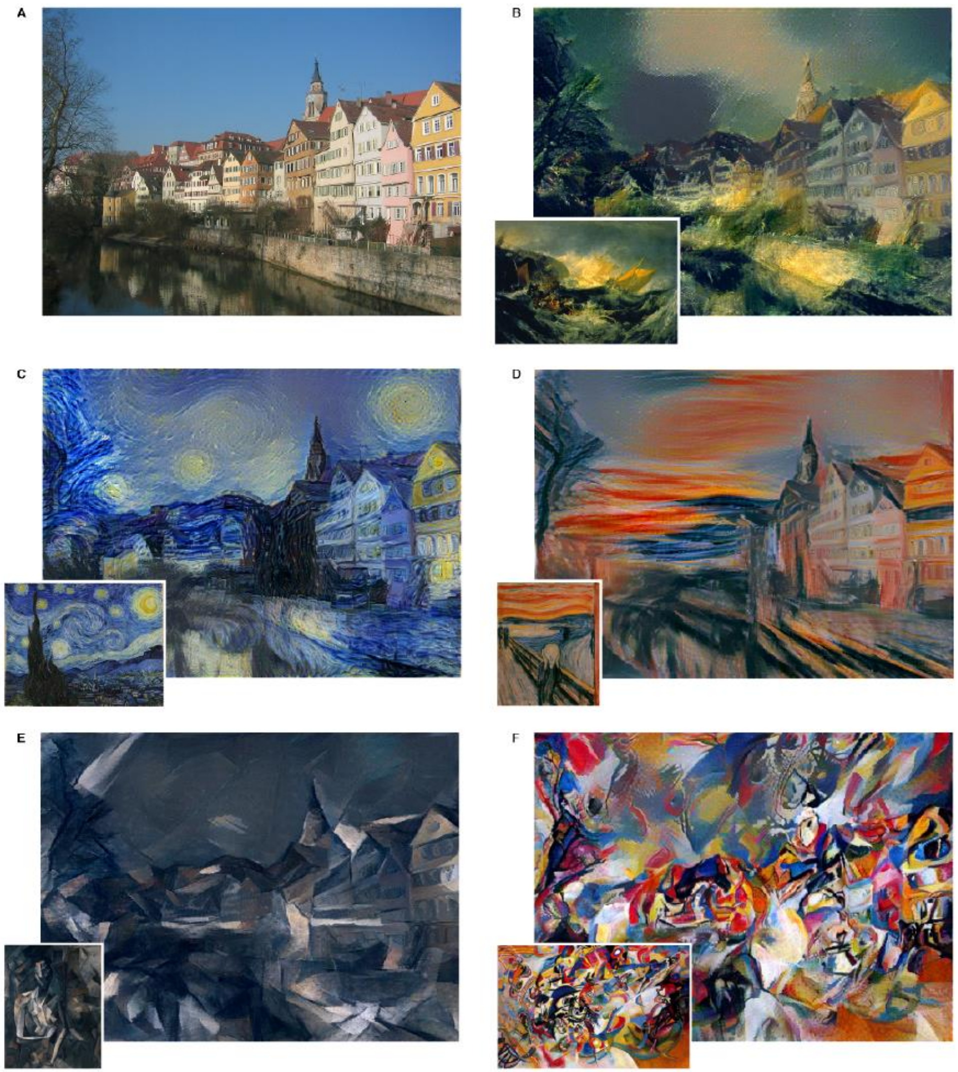
Choi et al., CVPR 2018

Scribbler: Controlling Deep Image Synthesis with Sketch and Color

Figure 1. A user can sketch and scribble colors to control deep image synthesis. On the left is an image generated from a hand drawn sketch. On the right several objects have been deleted from the sketch, a vase has been added, and the color of various scene elements has been constrained by sparse color strokes. For best resolution and additional results, see scribbler.eye.gatech.edu

Sangkloy et al., CVPR 2017

Image Style Transfer Using Convolutional Neural Networks



DeepArt.io – try it for yourself!

Gatys et al., CVPR 2016

MovieQA: Understanding Stories in Movies through Question-Answering

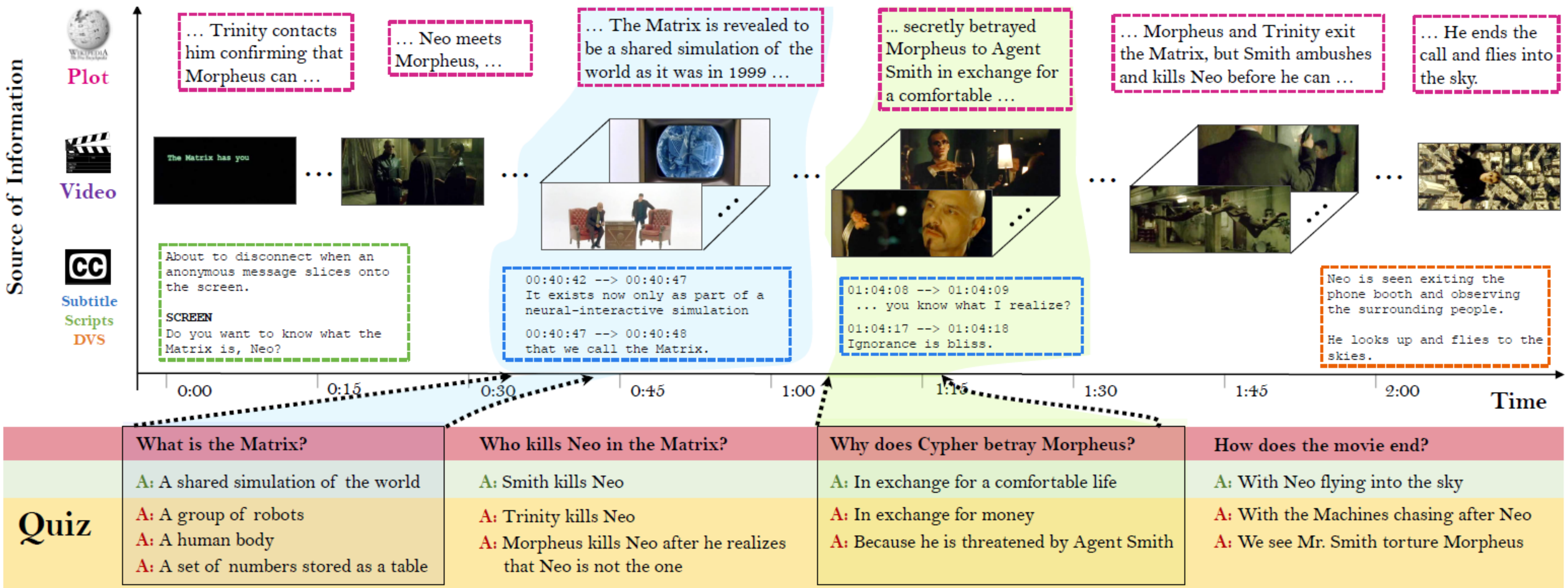
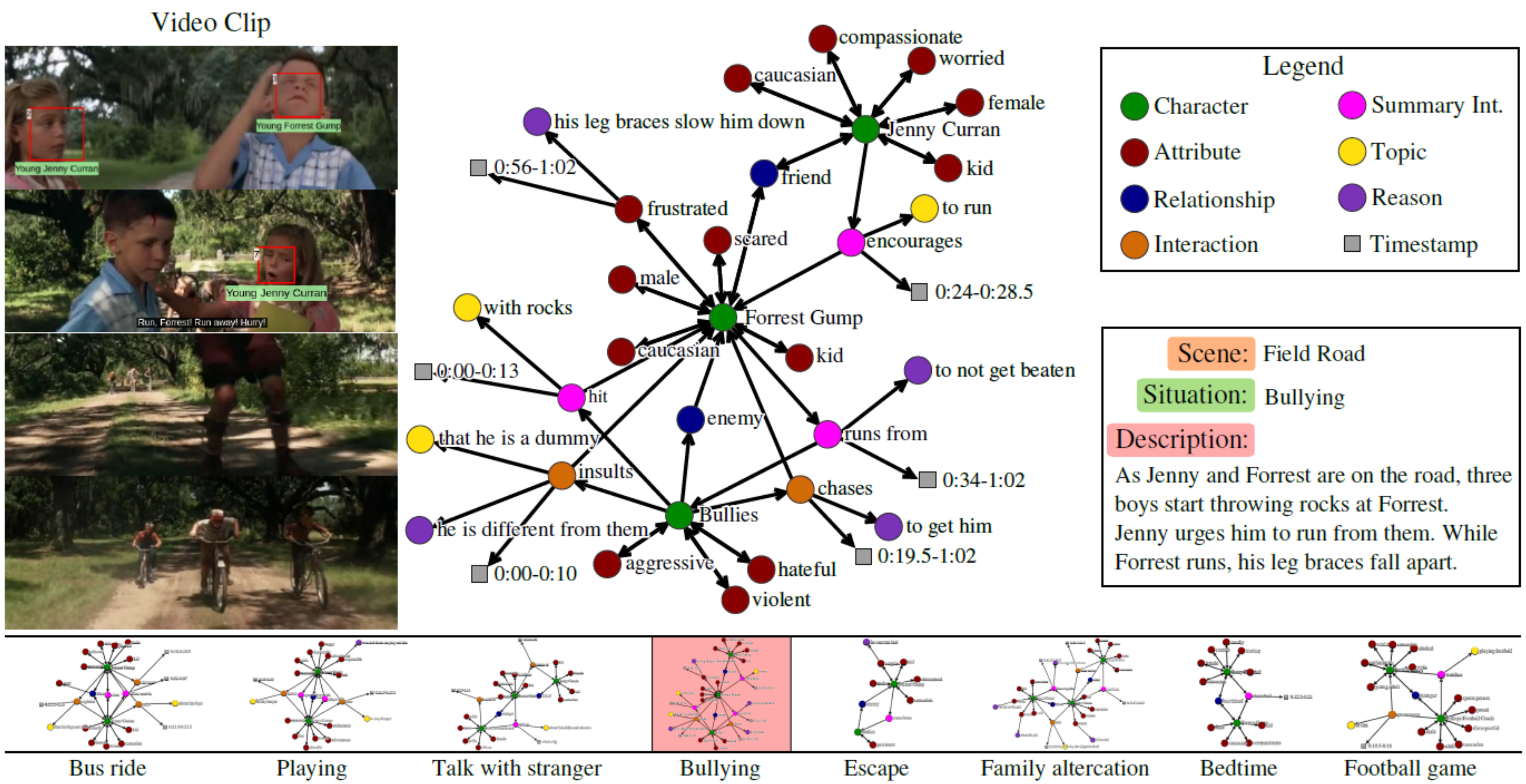


Figure 1: Our MovieQA dataset contains 14,944 questions about 408 movies. It contains multiple sources of information: plots, subtitles, video clips, scripts, and DVS transcriptions. In this figure we show example QAs from *The Matrix* and localize them in the timeline.

Tapaswi et al., CVPR 2016

MovieGraphs: Towards Understanding Human-Centric Situations From Videos



Vicol et al., CVPR 2018

Automatic Understanding of Image and Video Advertisements

Zaeem Hussain, Mingda Zhang, Xiaozhong Zhang, Keren Ye, Christopher Thomas, Zuha Agha, Nathan Ong, Adriana Kovashka
University of Pittsburgh

Understanding advertisements is more challenging than simply recognizing physical content from images, as ads employ a variety of strategies to persuade viewers.

Symbolism | Atypical Objects | Culture/Memes

We collect an advertisement dataset containing 64,832 images and 3,477 videos, each annotated by 3-5 human workers from Amazon Mechanical Turk.

Image	Topic	204,340	Strategy	20,000
	Sentiment	102,340	Symbol	64,131
	Q+A Pair	202,090	Slogan	11,130
Video	Topic	17,345	Fun/Exciting	15,380
	Sentiment	17,345	English?	17,374
	Q+A Pair	17,345	Effective	16,721

Here are some sample annotations in our dataset.

What's being advertised in this image?

Cars, automobiles

What sentiments are provoked in the viewer?

Amused, Creative, Impressed, Youthful, Conscious

What strategies are used to persuade viewer?

Symbolism, Contrast, Straightforward, Transferred qualities

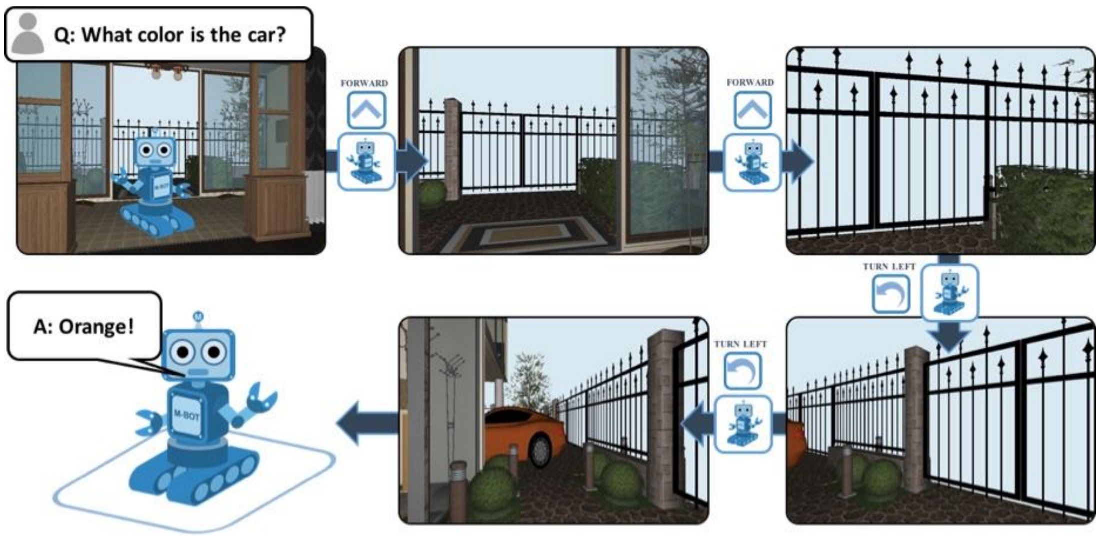
What should the viewer do, and why should they do this?

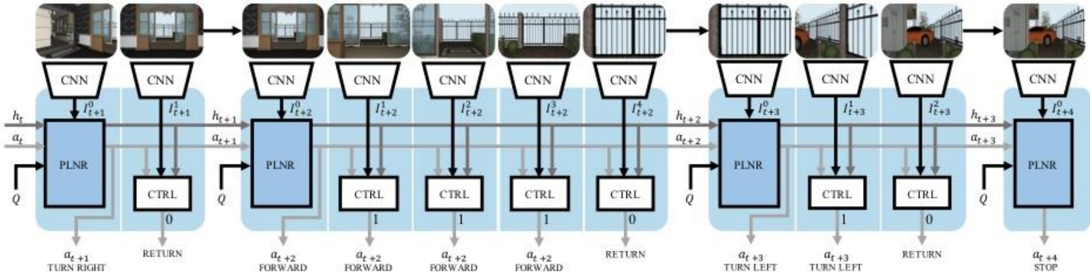
- I should buy Volkswagen because it can hold a big bear.
- I should buy VW SUV because it can fit anything and everything in it.
- I should buy this car because it can hold everything I need.

More information available at http://cs.pitt.edu/~kovashka/ads

Hussein et al., CVPR 2017

Embodied Question Answering





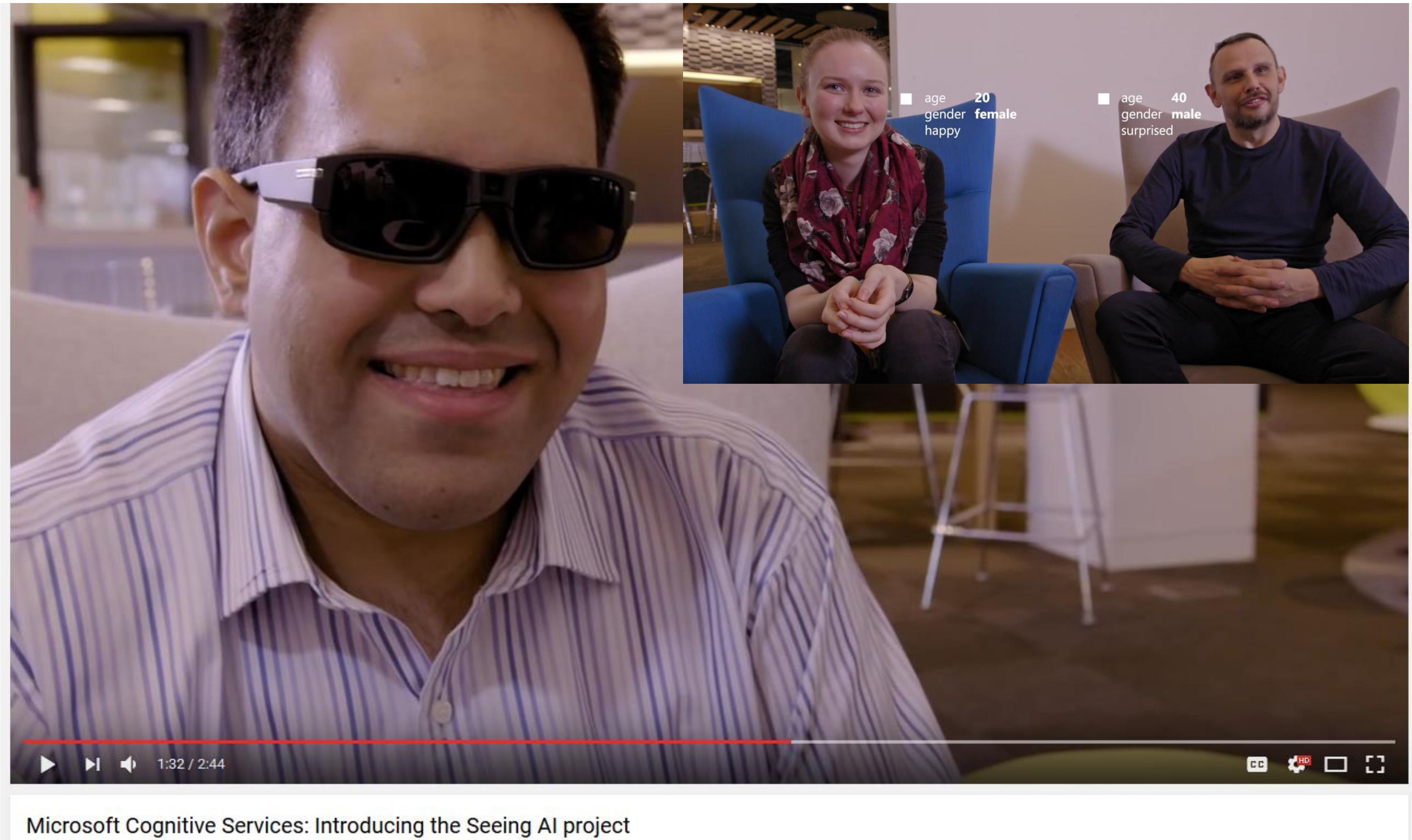
Das et al., CVPR 2018

Computer vision is not solved

- Deep learning makes excellent use of massive data (labeled for the task of interest?)
  - But it’s hard to understand *how* it does so
  - It doesn’t work well when massive data is not available and your task is different than tasks for which data is available
  - We can recognize objects with 97% accuracy but reasoning about relationships and intent is harder

Seeing AI

YouTube link



Obstacles?

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence Group July 7, 1966
Vision Memo. No. 100.

THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

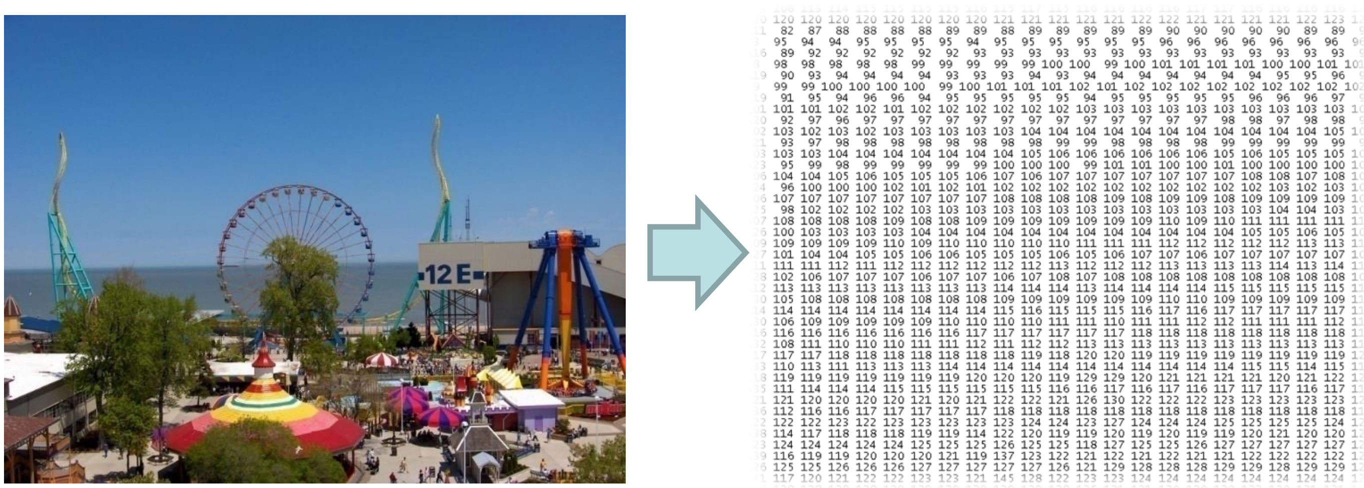
Read more about the history: Szeliski Sec. 1.2

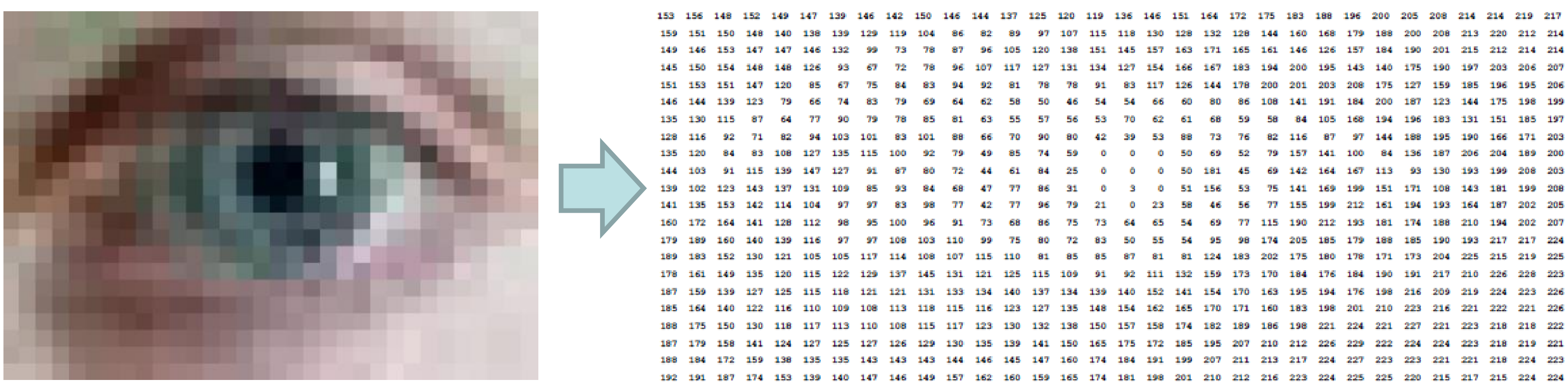
Why is vision difficult?

- Ill-posed problem: real world much more complex than what we can measure in images
  - 3D → 2D
- Impossible to literally “invert” image formation process with limited information
  - Need information outside of this particular image to generalize what image portrays (e.g. to resolve occlusion)

Adapted from Kristen Grauman

What the computer gets





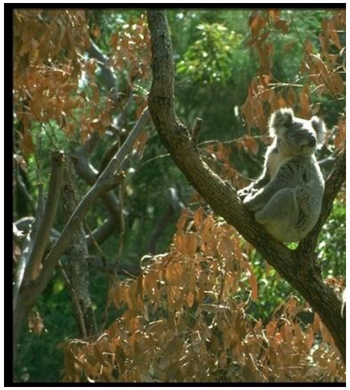
Why is this problematic?

Adapted from Kristen Grauman and Lana Lazebnik

Challenges: many nuisance parameters

Illumination

Object pose



Clutter

Occlusions

Intra-class appearance

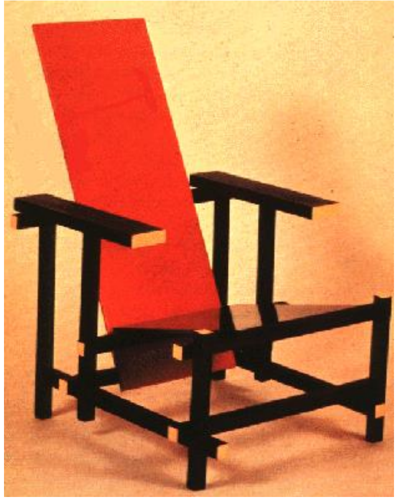
Viewpoint

Think again about the pixels…

Kristen Grauman

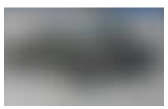
Challenges: intra-class variation

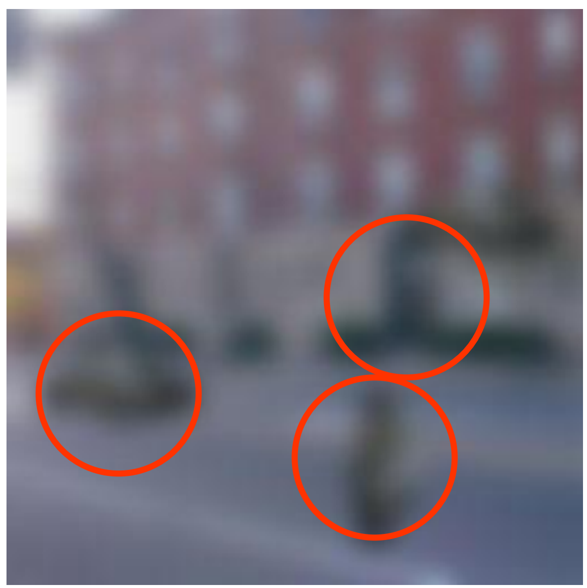
CMOA Pittsburgh



slide credit: Fei-Fei, Fergus & Torralba

Challenges: importance of context



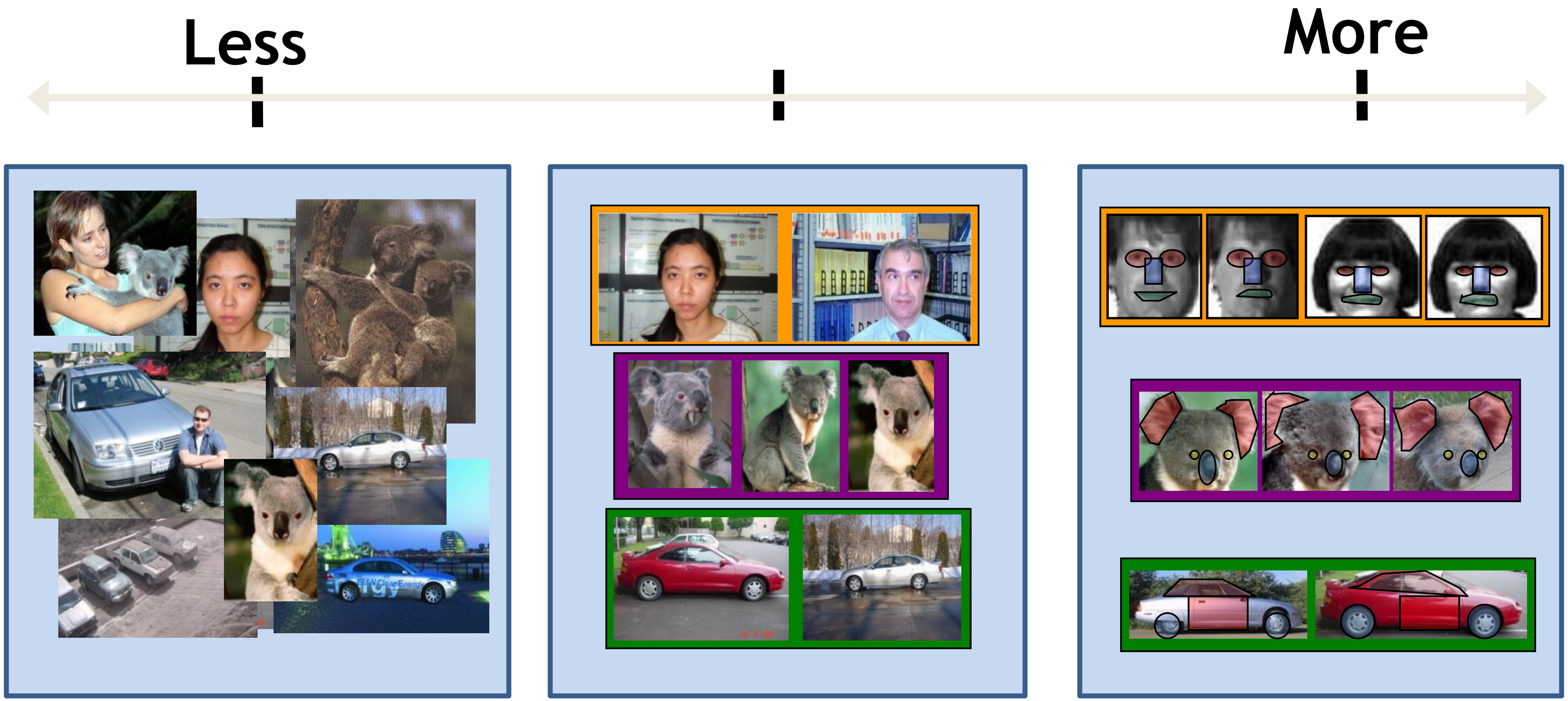


Challenges: Complexity

- Thousands to millions of pixels in an image
- 3,000-30,000 human recognizable object categories
- 30+ degrees of freedom in the pose of articulated objects (humans)
- Billions of images indexed by Google Image Search
- 1.424 billion smart camera phones sold in 2015
- About half of the cerebral cortex in primates is devoted to processing visual information [Felleman and van Essen 1991]

Kristen Grauman

Challenges: Limited supervision



Unlabeled, multiple objects

Classes labeled, some clutter

Cropped to object, parts and classes labeled

Kristen Grauman

Challenges: Vision requires reasoning

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

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

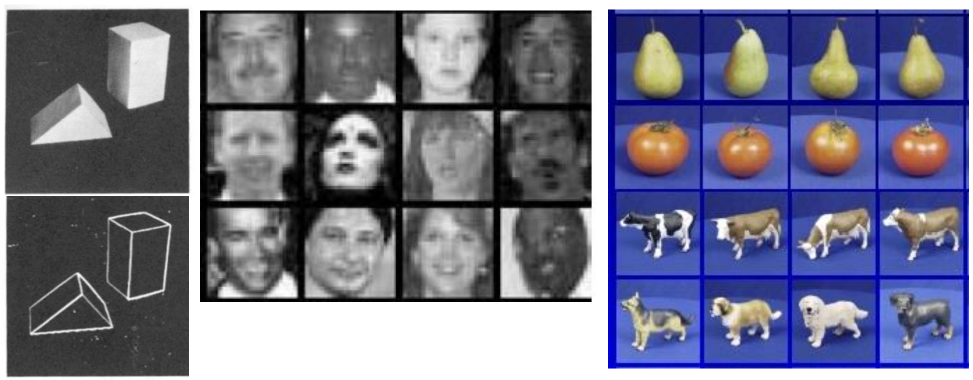
Is this person expecting company?
What is just under the tree?

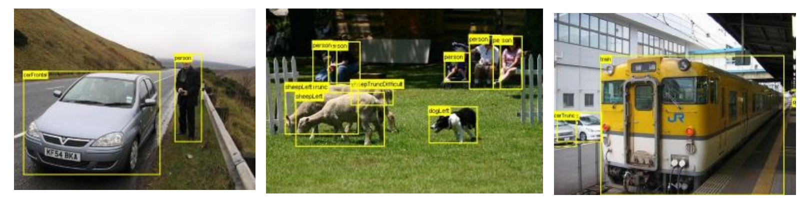
Does it appear to be rainy?
Does this person have 20/20 vision?

Antol et al., "VQA: Visual Question Answering", ICCV 2015

Evolution of datasets

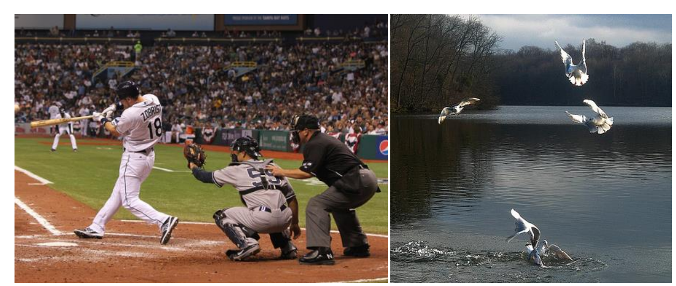
- Challenging problem → active research area





PASCAL:
20 categories, 12k images

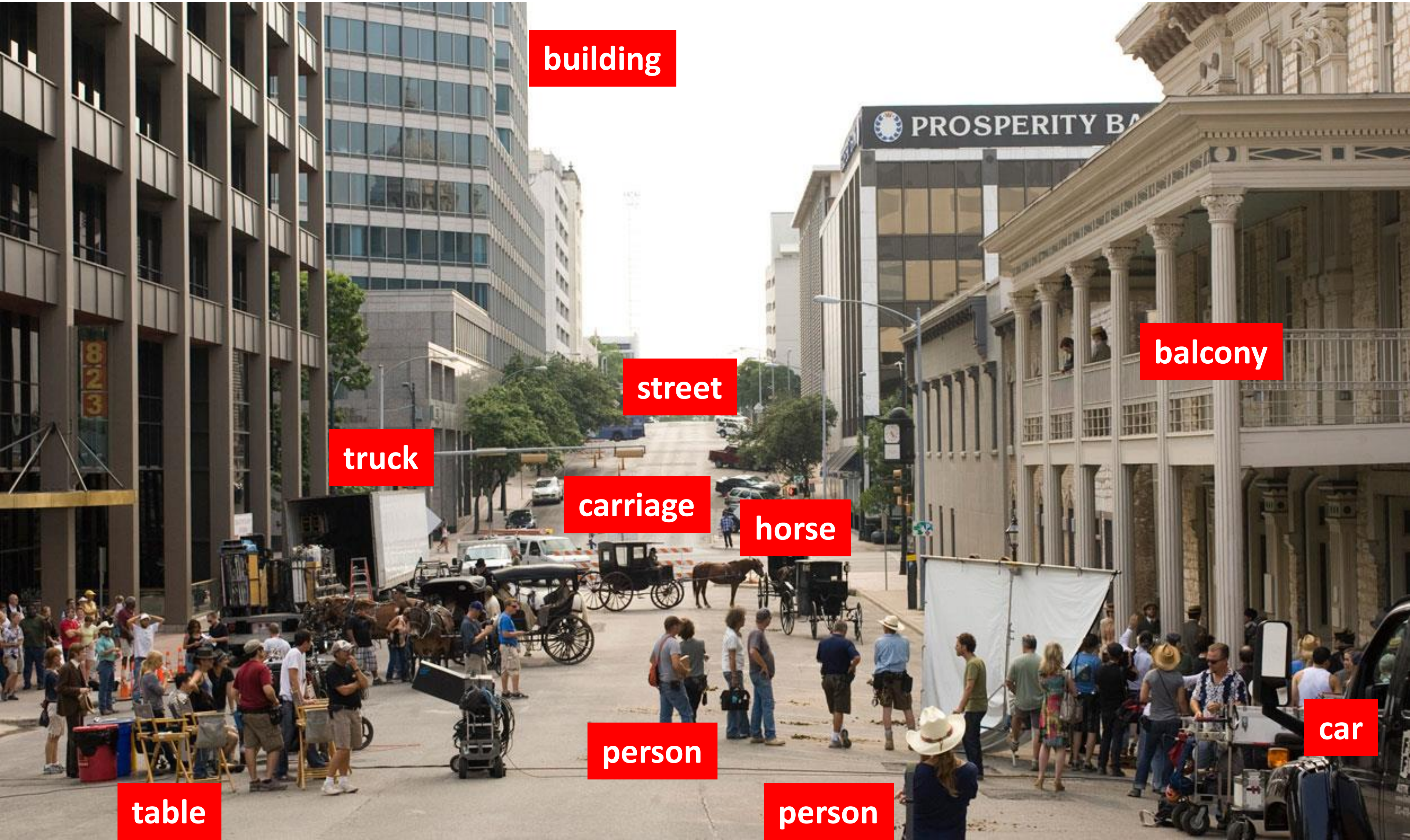
ImageNet:
22k categories, 14mil images



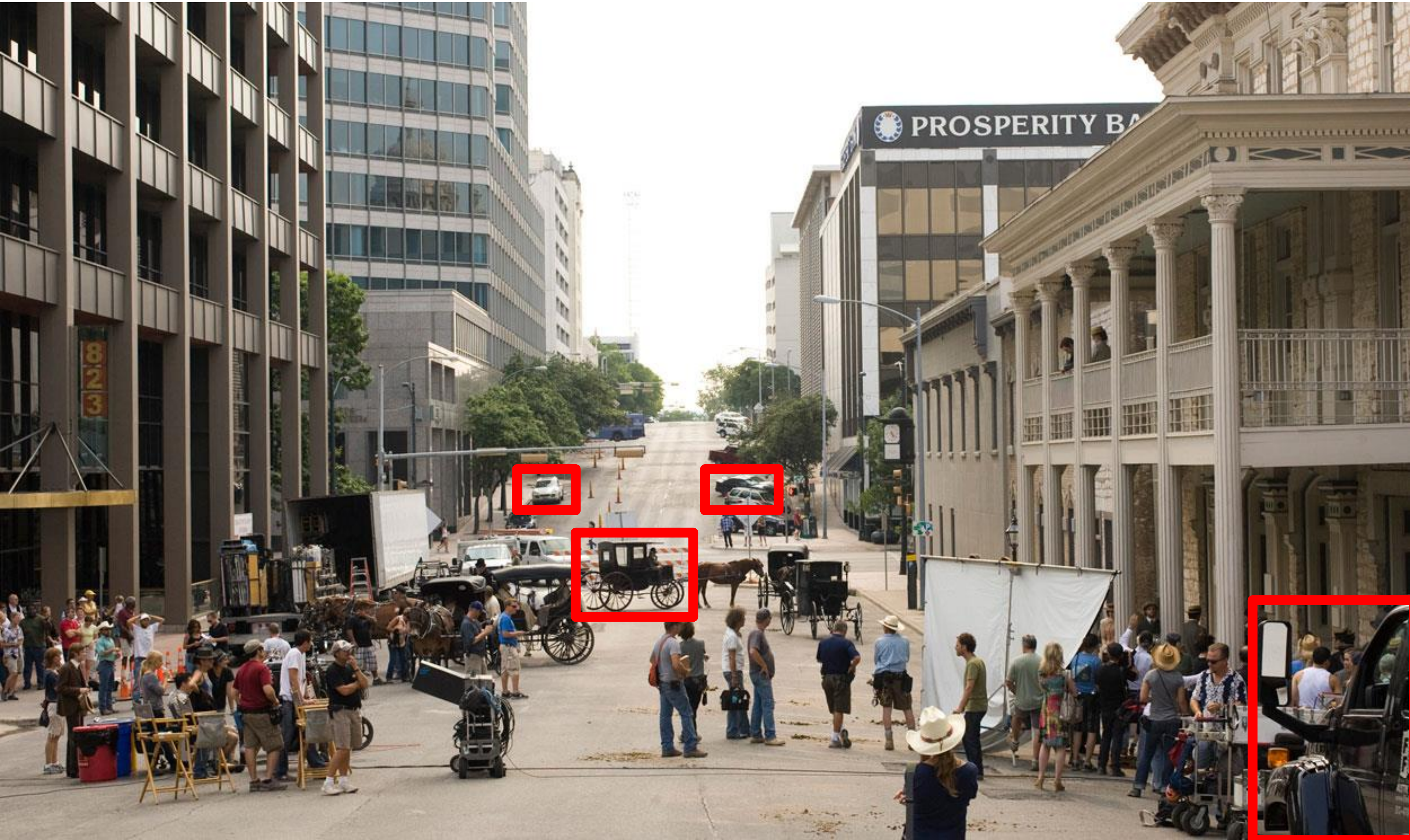
Microsoft COCO:
80 categories, 300k images

Some Visual Recognition Problems: Why are they challenging?

Recognition: What objects do you see?



Detection: Where are the cars?



Activity: What is this person doing?

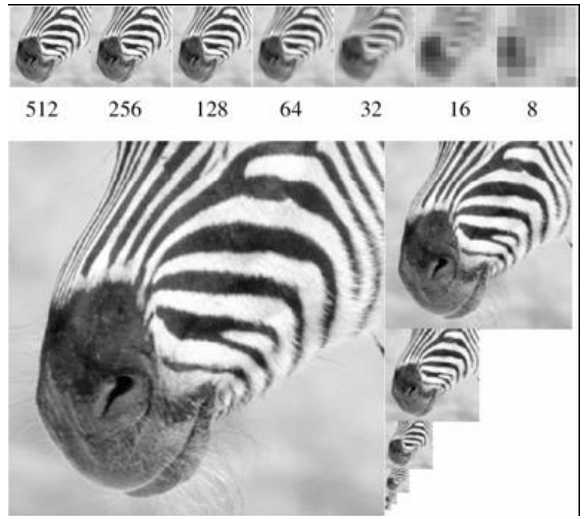
Scene: Is this an indoor scene?

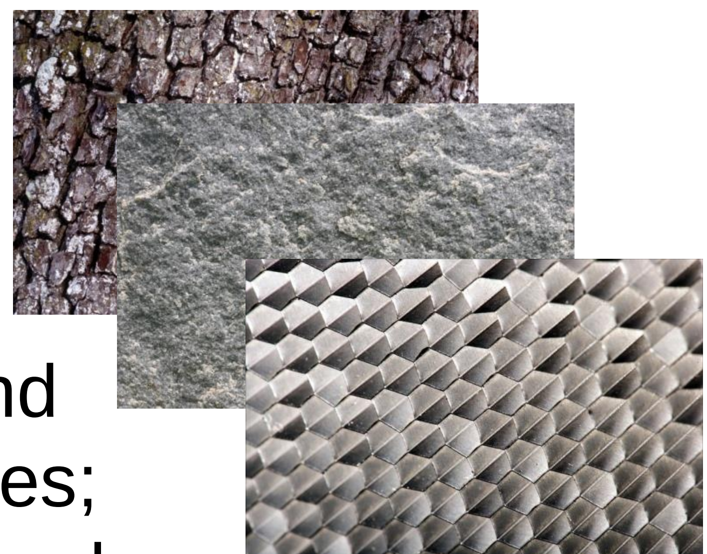
Instance: Which city? Which building?

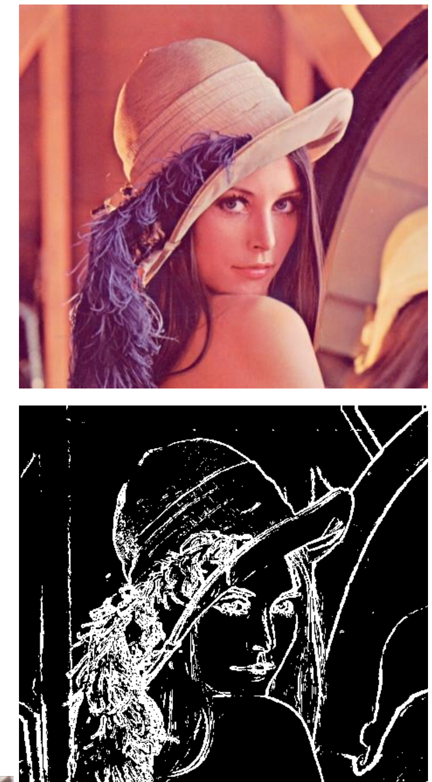
Visual question answering:
Why is there a carriage in the street?

Overview of topics

Features and filters





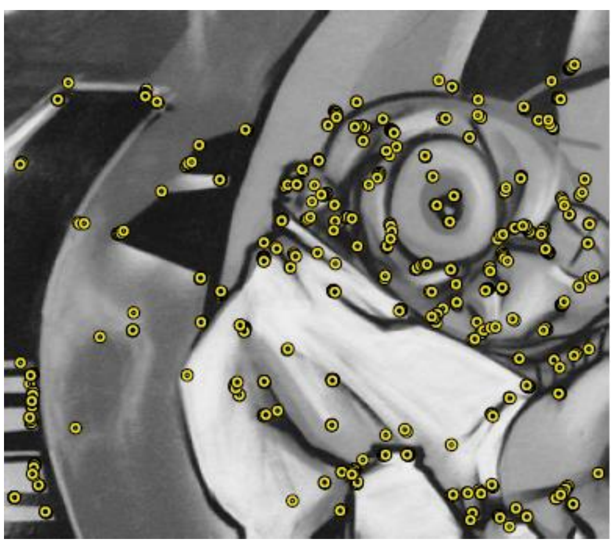


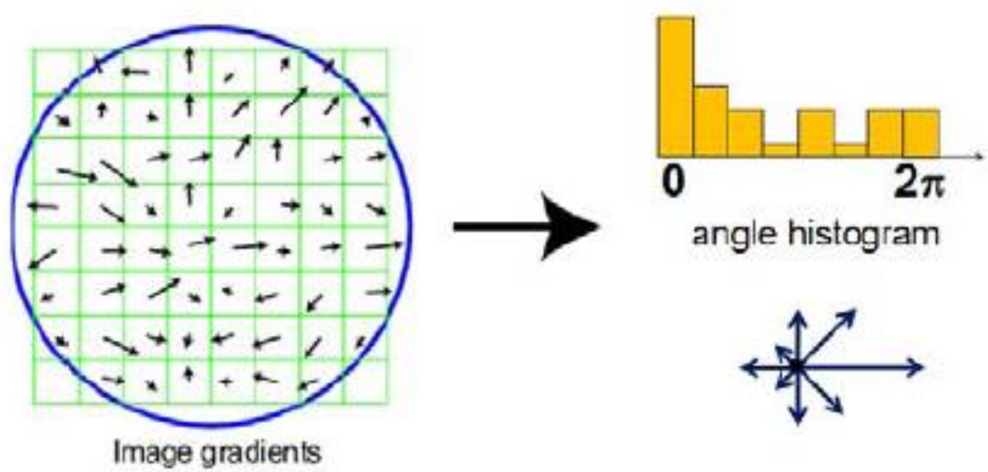
- Transforming and describing images; textures, colors, edges

Kristen Grauman

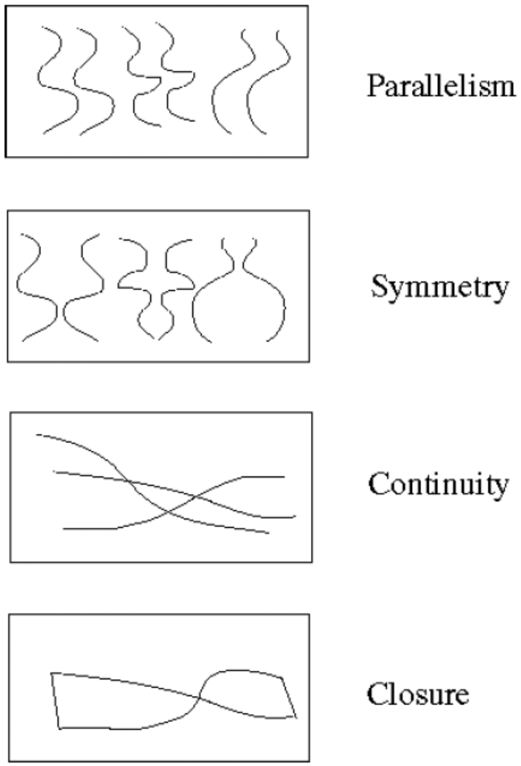
Features and filters

- Detecting distinctive + repeatable features
- Describing images with local statistics



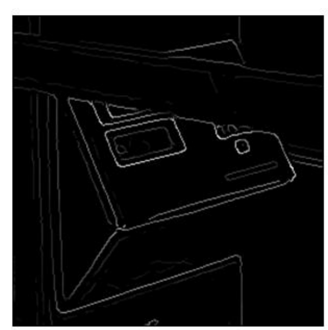


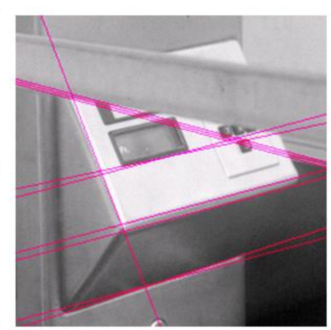
Grouping and fitting

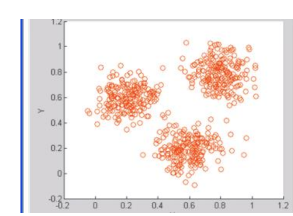


[fig from Shi et al]

- Clustering, segmentation, fitting; what parts belong together?



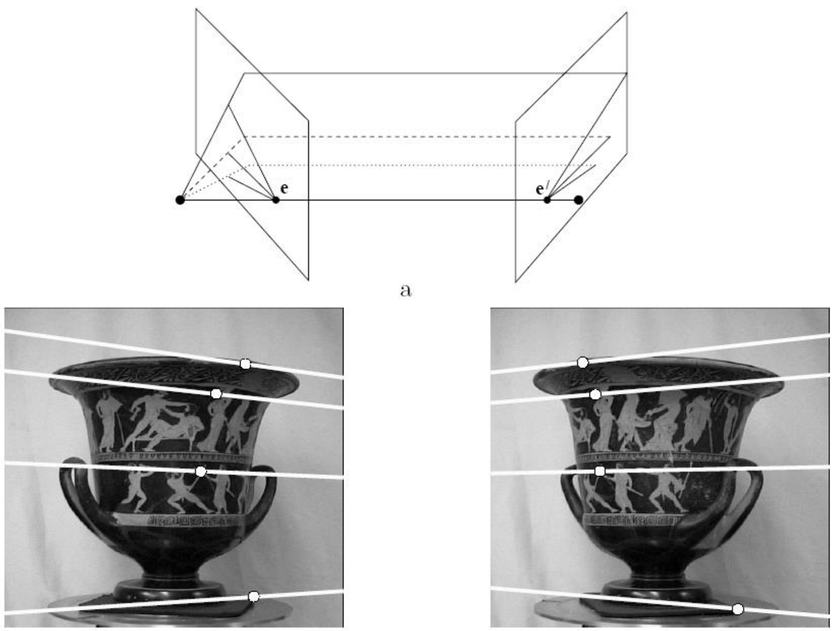




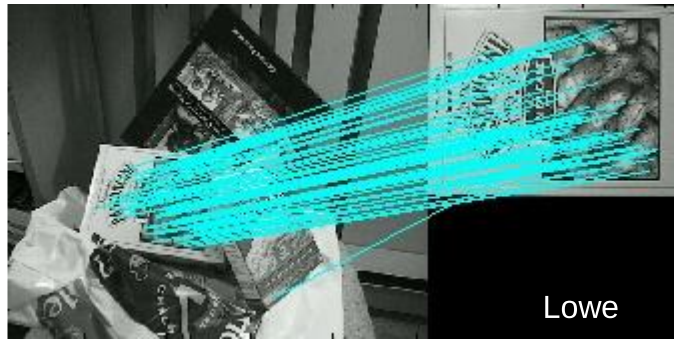
Kristen Grauman

Multiple views

- Multi-view geometry, matching, invariant features, stereo vision



Hartley and Zisserman



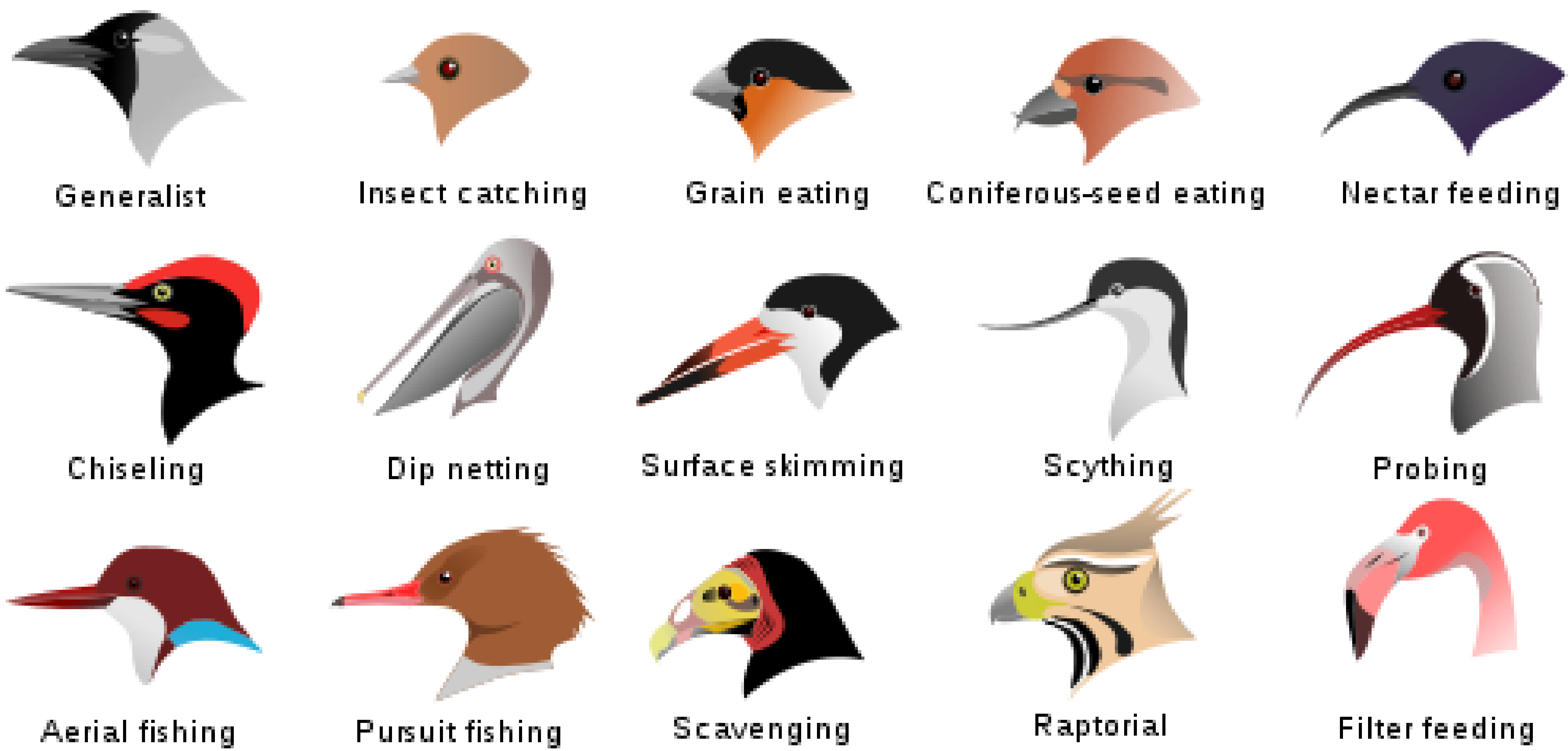
Lowe

Fei-Fei Li

Kristen Grauman

Image categorization

- Fine-grained recognition

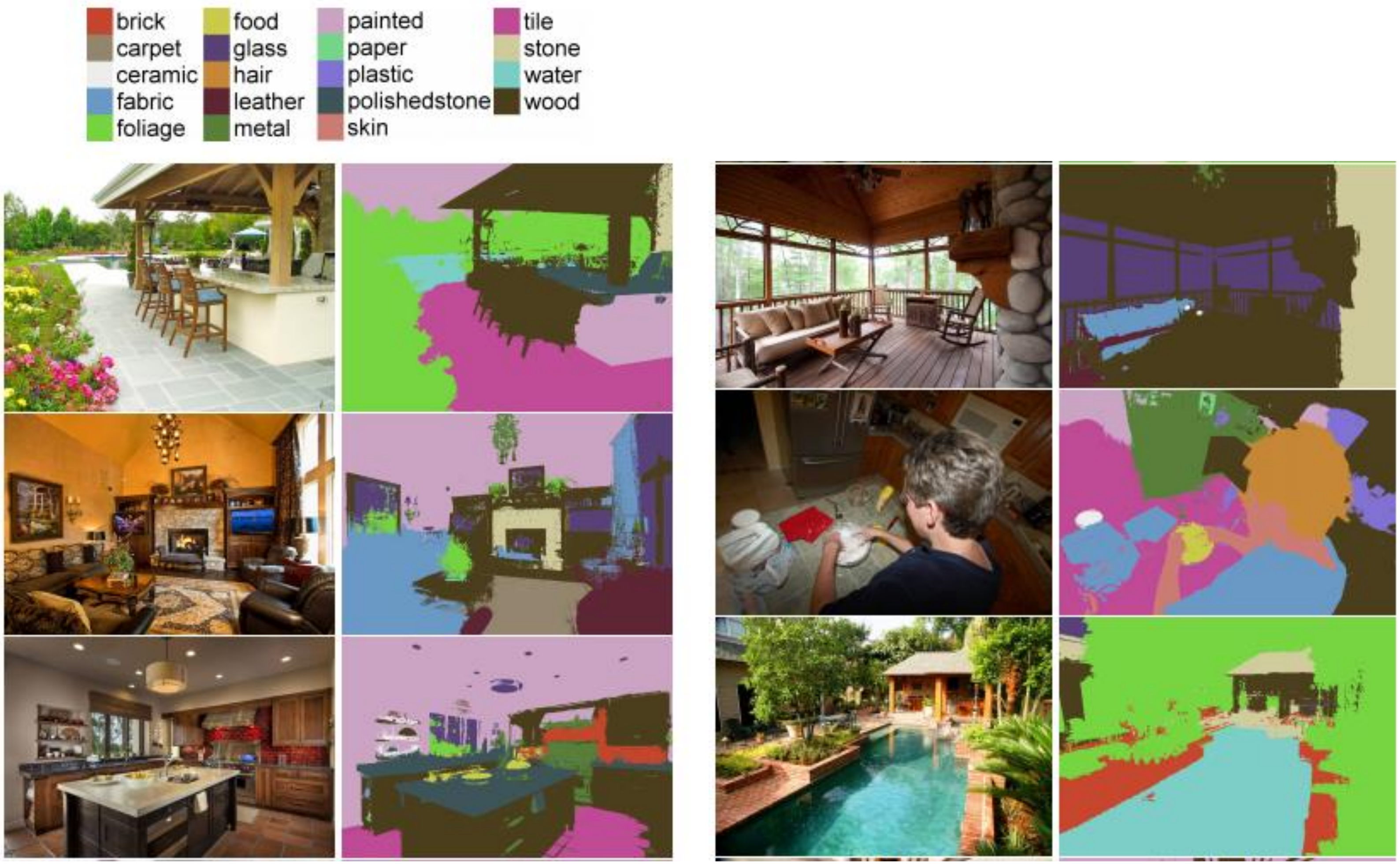


[Visipedia Project]

Slide credit: D. Hoiem

Image categorization

- Material recognition

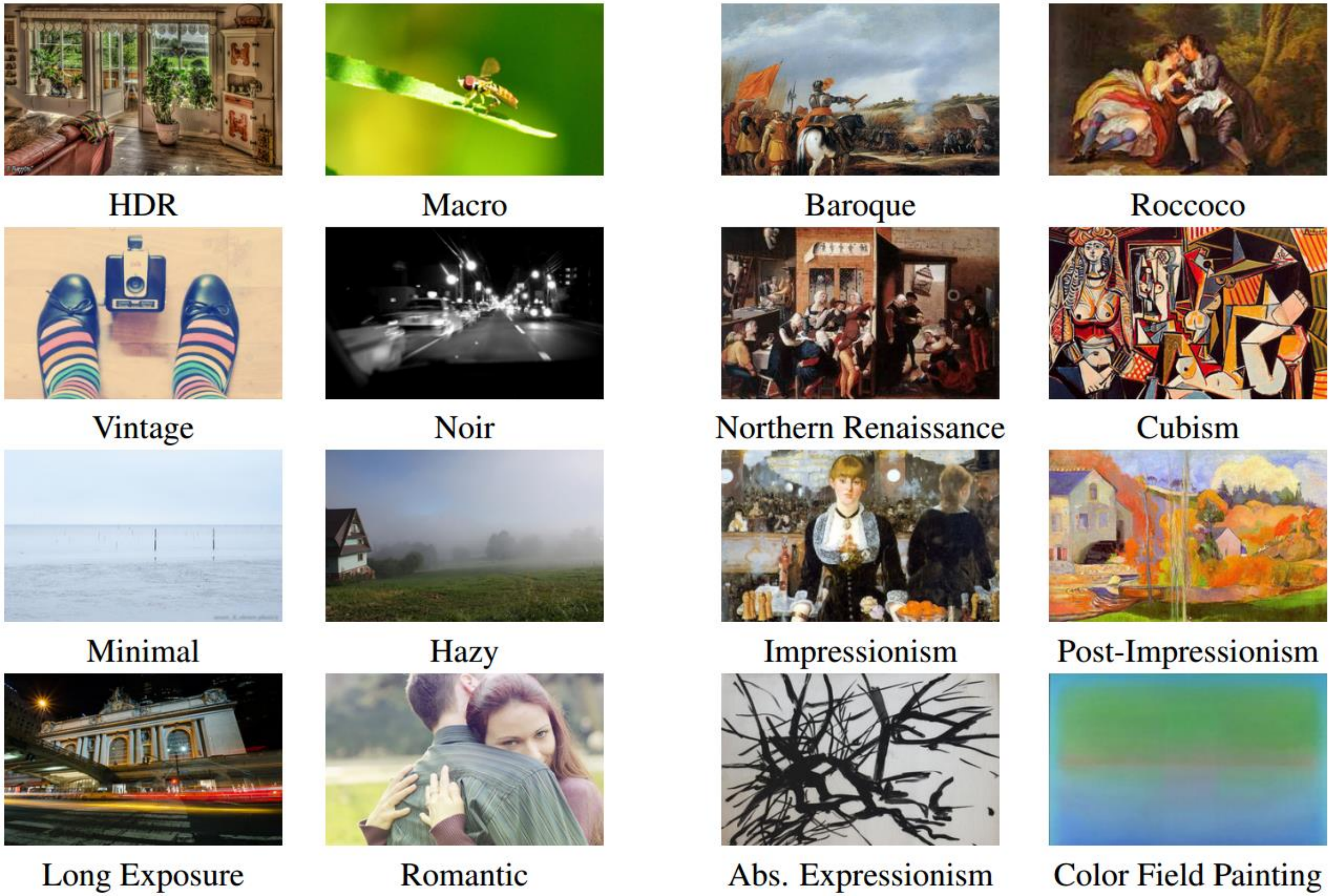


[Bell et al. CVPR 2015]

Slide credit: D. Hoiem

Image categorization

- Image style recognition

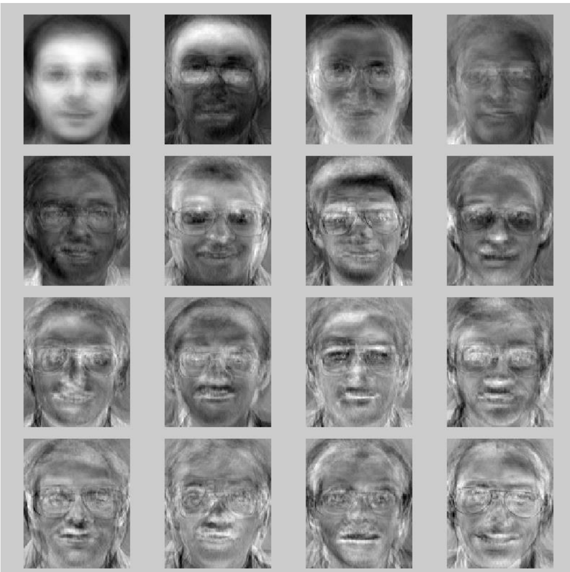


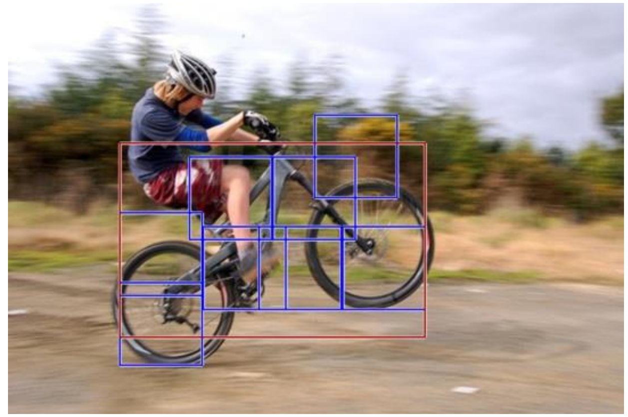
Flickr Style: 80K images covering 20 styles. Wikipaintings: 85K images for 25 art genres.

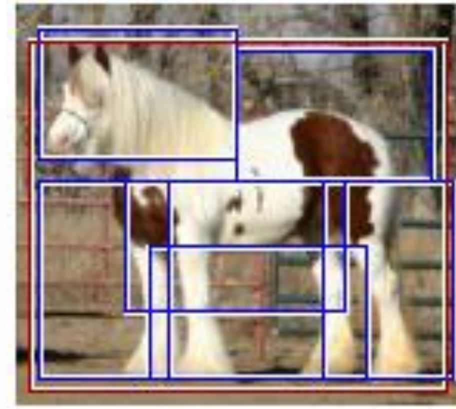
[Karayev et al. BMVC 2014]

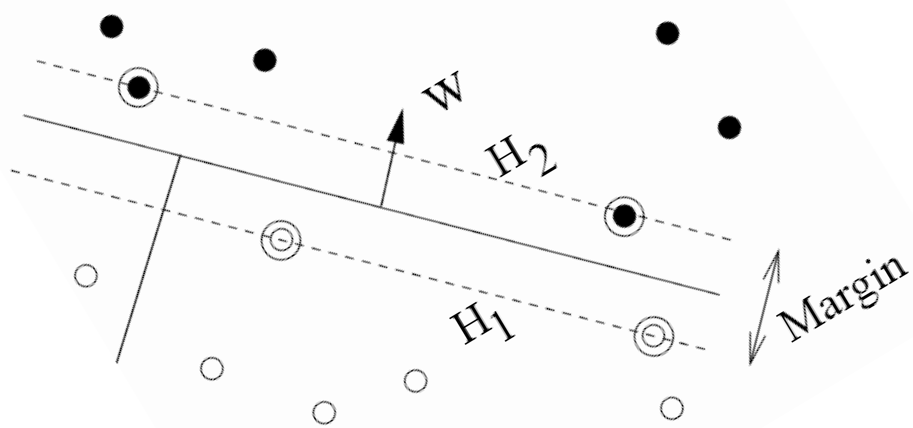
Slide credit: D. Hoiem

Visual recognition and SVMs









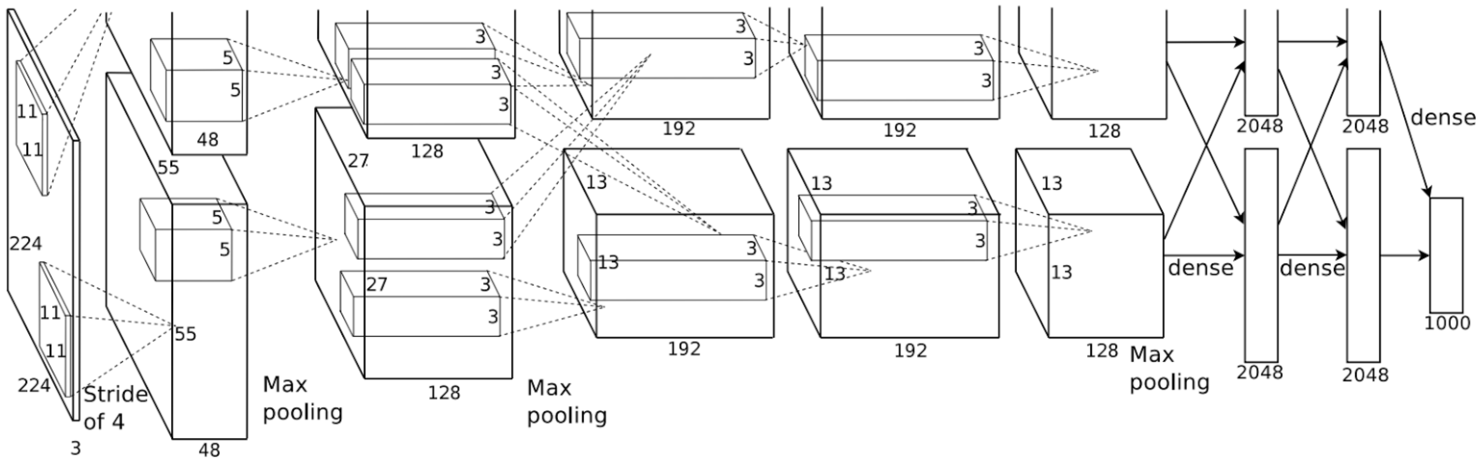
- Recognizing objects and categories, learning techniques

Adapted from Kristen Grauman

Convolutional neural networks (CNNs)

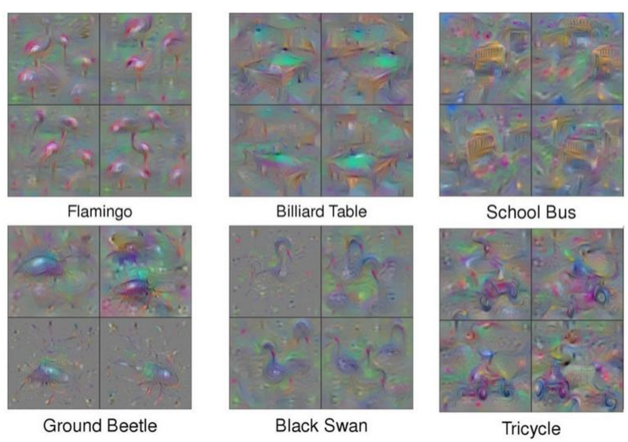
- State-of-the-art on many recognition tasks

Image



Prediction

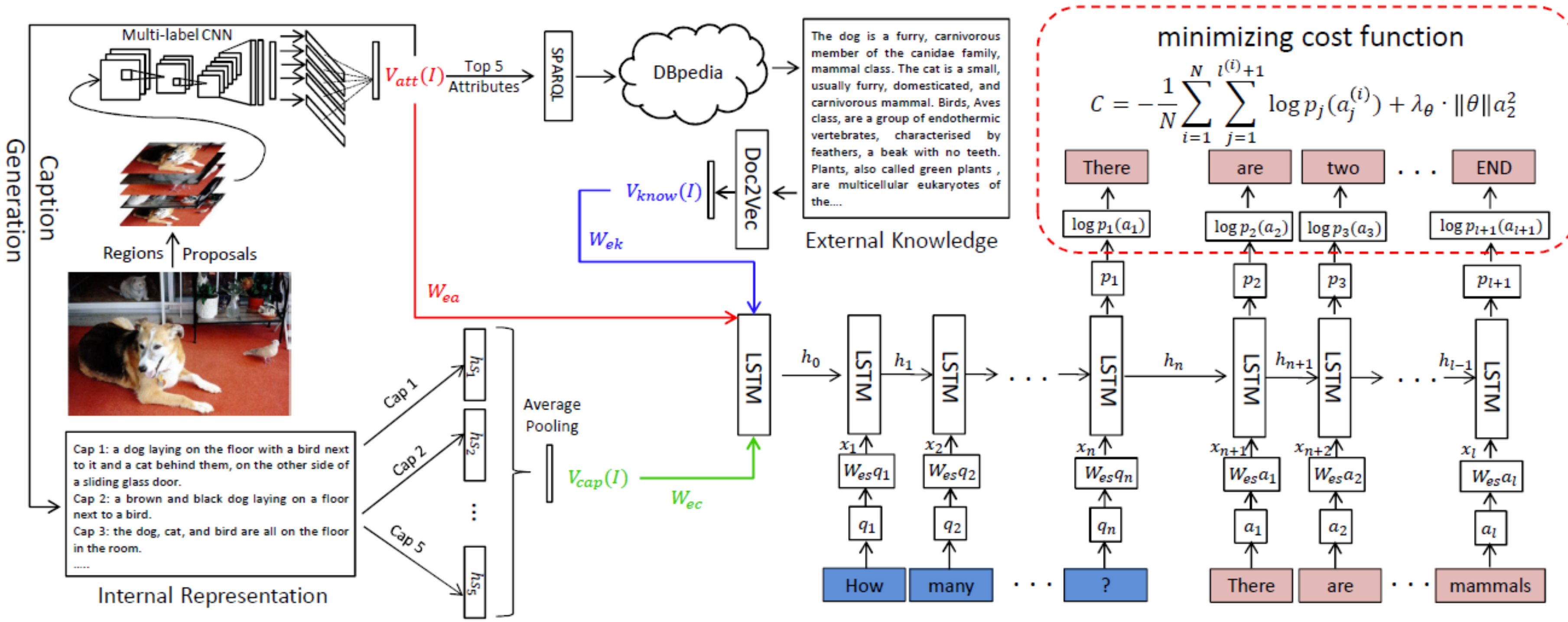
Krizhevsky et al., NIPS 2012



Yosinski et al., ICML DL workshop 2015

Recurrent neural networks

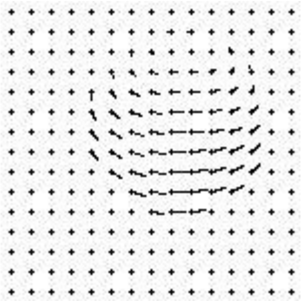
- Sequence processing, e.g. question answering

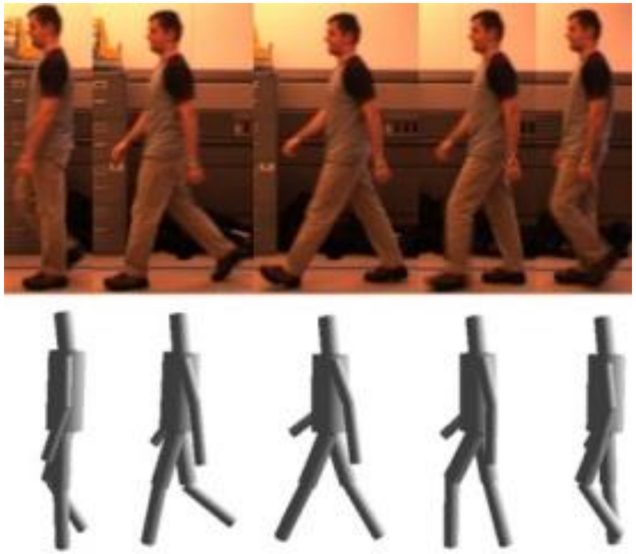


Wu et al., CVPR 2016

Motion and tracking

- Tracking objects, video analysis



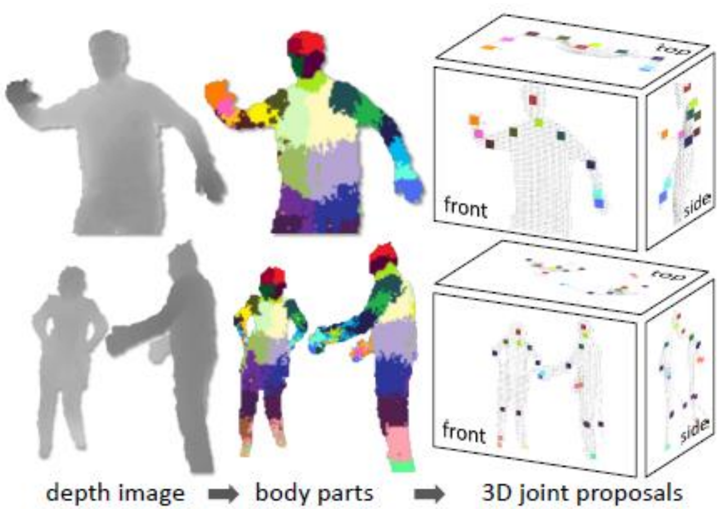


Tomas Izo

Kristen Grauman

Pose and actions

- Automatically annotating human pose (joints)
- Recognizing actions in first-person video



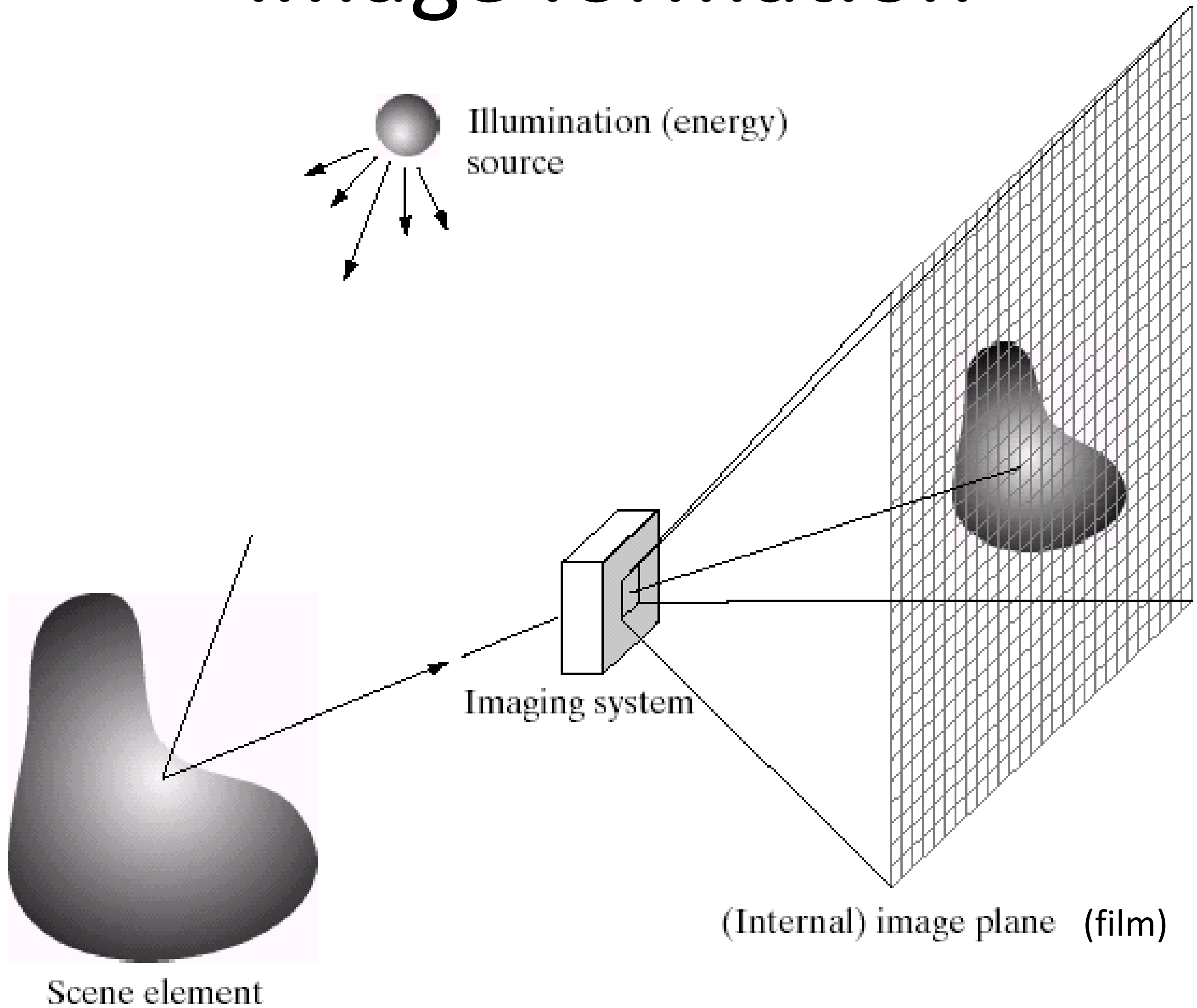
Linear algebra review

See http://cs229.stanford.edu/section/cs229-linalg.pdf for more

What are images? (in Matlab)

- Matlab treats images as matrices of numbers
- To proceed, let’s talk very briefly about how images are formed

Image formation



Slide credit: Derek Hoiem

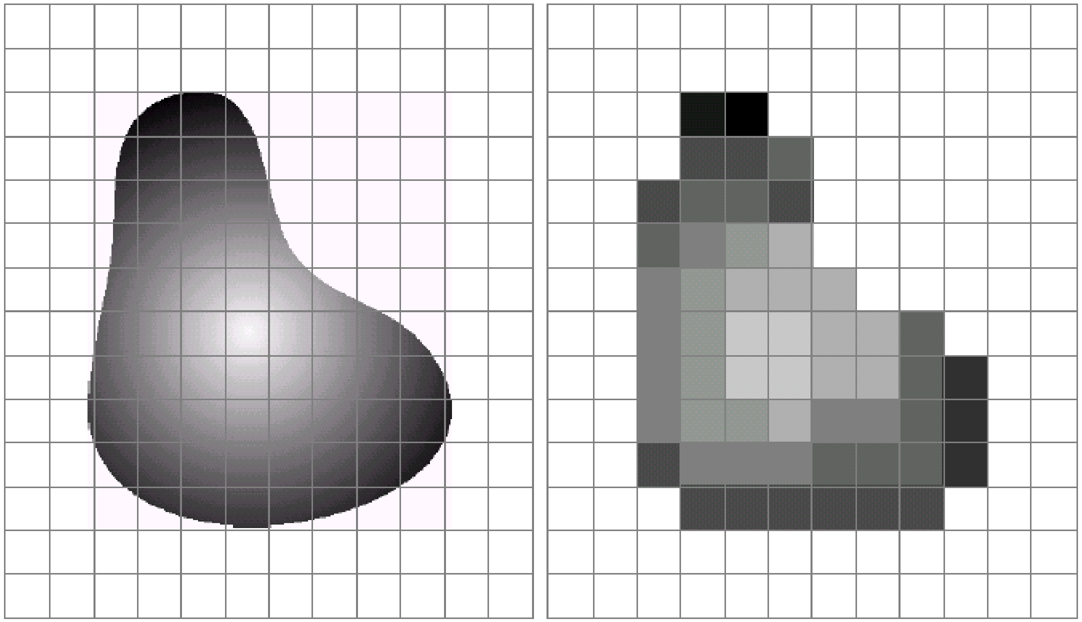
Digital camera

A digital camera replaces film with a sensor array

- Each cell in the array is light-sensitive diode that converts photons to electrons

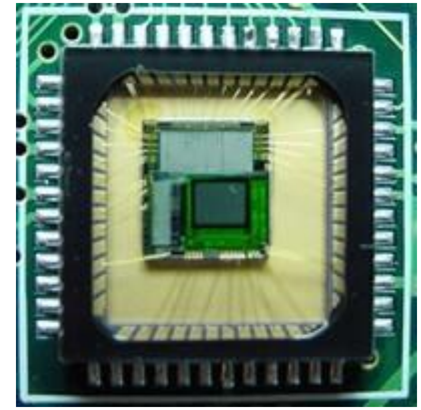
http://electronics.howstuffworks.com/cameras-photography/digital/digital-camera.htm

Digital images



a b

FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.

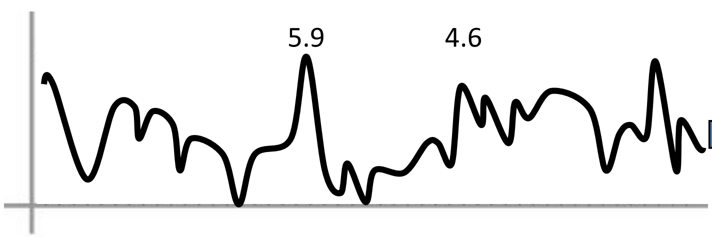


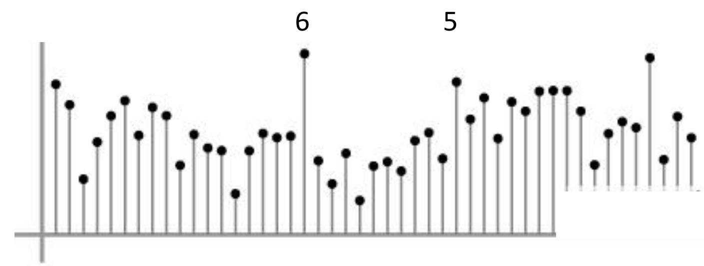
- **Sample** the 2D space on a regular grid
- **Quantize** each sample (round to nearest integer)

Digital images

- **Sample** the 2D space on a regular grid
- **Quantize** each sample (round to nearest integer)
- What does quantizing signal look like?

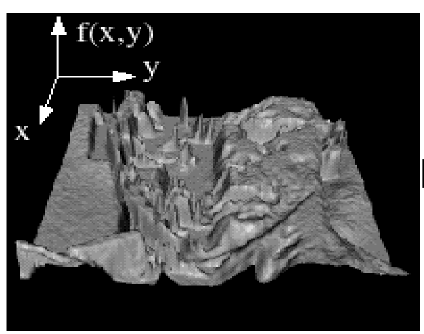
1D





- Image thus represented as a matrix of integer values.

2D



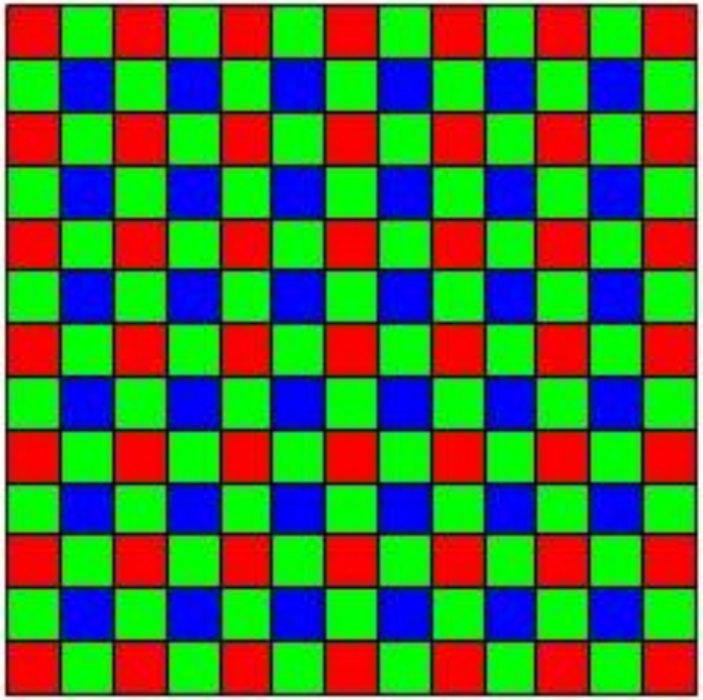
j →

i ↓

62	79	23	119	120	105	4	0
10	10	9	62	12	78	34	0
10	58	197	46	46	0	0	48
176	135	5	188	191	68	0	49
2	1	1	29	26	37	0	77
0	89	144	147	187	102	62	208
255	252	0	166	123	62	0	31
166	63	127	17	1	0	99	30

Adapted from S. Seitz

Digital color images



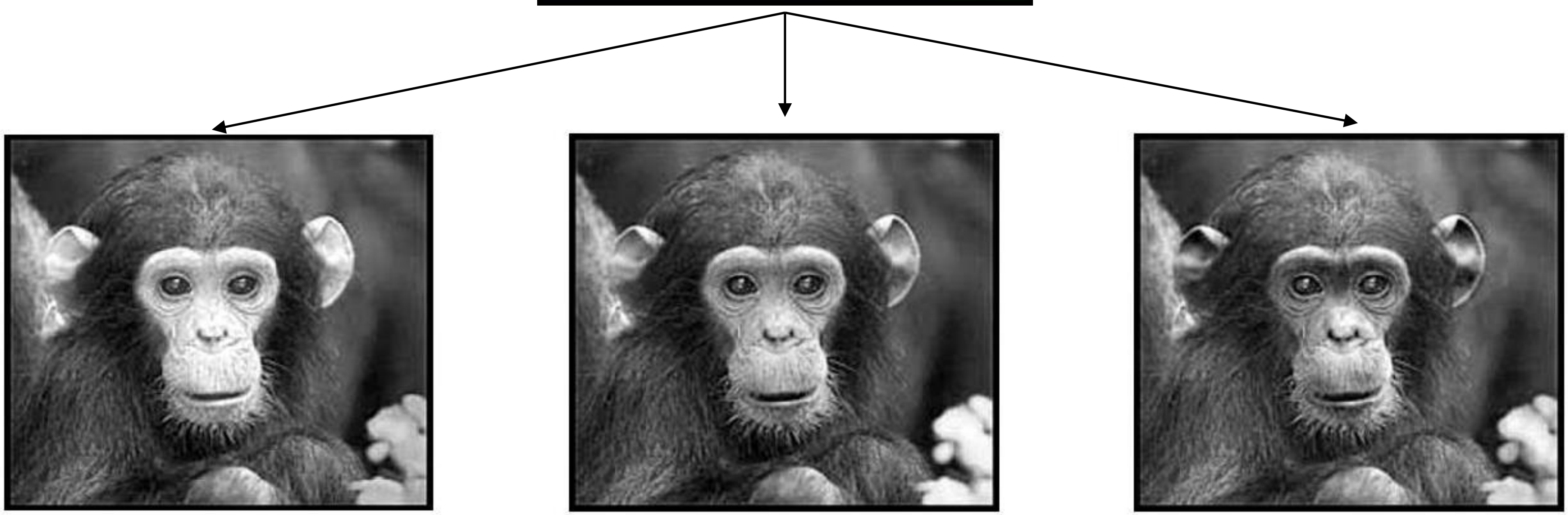
Bayer filter

Slide credit: Kristen Grauman

Digital color images

Color images,
RGB color space:

Split image into
three channels

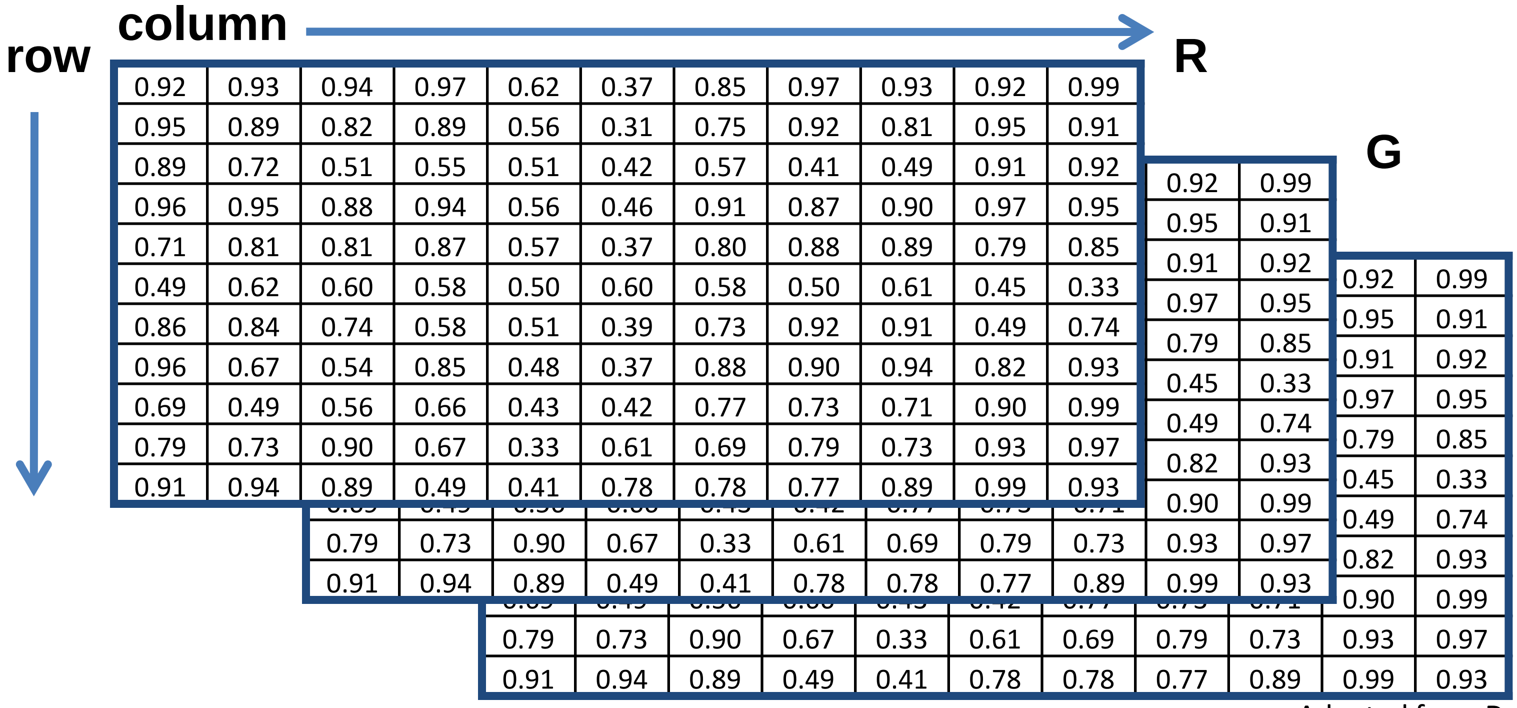


R G B

Adapted from Kristen Grauman

Images in Matlab

- Color images represented as a matrix with multiple channels (=1 if grayscale)
- Suppose we have a NxM RGB image called “im”
  - im(1,1,1) = top-left pixel value in R-channel
  - im(y, x, b) = y pixels **down**, x pixels **to right** in the b^th^ channel
  - im(N, M, 3) = bottom-right pixel in B-channel
- imread(filename) returns a uint8 image (values 0 to 255)
  - Convert to double format with double or im2double



0.92	0.93	0.94	0.97	0.62	0.37	0.85	0.97	0.93	0.92	0.99
0.95	0.89	0.82	0.89	0.56	0.31	0.75	0.92	0.81	0.95	0.91
0.89	0.72	0.51	0.55	0.51	0.42	0.57	0.41	0.49	0.91	0.92
0.96	0.95	0.88	0.94	0.56	0.46	0.91	0.87	0.90	0.97	0.95
0.71	0.81	0.81	0.87	0.57	0.37	0.80	0.88	0.89	0.79	0.85
0.49	0.62	0.60	0.58	0.50	0.60	0.58	0.50	0.61	0.45	0.33
0.86	0.84	0.74	0.58	0.51	0.39	0.73	0.92	0.91	0.49	0.74
0.96	0.67	0.54	0.85	0.48	0.37	0.88	0.90	0.94	0.82	0.93
0.69	0.49	0.56	0.66	0.43	0.42	0.77	0.73	0.71	0.90	0.99
0.79	0.73	0.90	0.67	0.33	0.61	0.69	0.79	0.73	0.93	0.97
0.91	0.94	0.89	0.49	0.41	0.78	0.78	0.77	0.89	0.99	0.93

Adapted from Derek Hoiem

Vectors and Matrices

- Vectors and matrices are just collections of ordered numbers that represent something: movements in space, scaling factors, word counts, movie ratings, pixel brightnesses, etc.
- We'll define some common uses and standard operations on them.

Vector

- A column vector $\mathbf{v} \in \mathbb{R}^{n \times 1}$ where

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

- A row vector $\mathbf{v}^T \in \mathbb{R}^{1 \times n}$ where

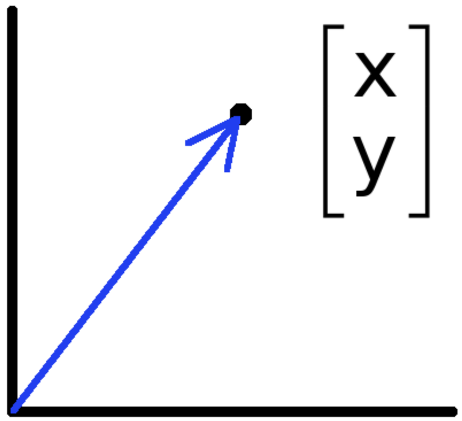
$$\mathbf{v}^T = \begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix}$$

T denotes the transpose operation

Vector

- You’ll want to keep track of the orientation of your vectors when programming in MATLAB.
- You can transpose a vector *V* in MATLAB by writing **V’**.

Vectors have two main uses



- Vectors can represent an offset in 2D or 3D space
- Points are just vectors from the origin

- Data can also be treated as a vector
- Such vectors don't have a geometric interpretation, but calculations like "distance" still have value

Matrix

- A matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is an array of numbers with size $m \downarrow$ by $n \rightarrow$, i.e. m rows and n columns.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & & & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}$$

- If $m = n$, we say that $\mathbf{A}$ is square.

Matrix Operations

- Addition

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} a+1 & b+2 \\ c+3 & d+4 \end{bmatrix}$$

  - Can only add a matrix with matching dimensions, or a scalar.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + 7 = \begin{bmatrix} a+7 & b+7 \\ c+7 & d+7 \end{bmatrix}$$

- Scaling

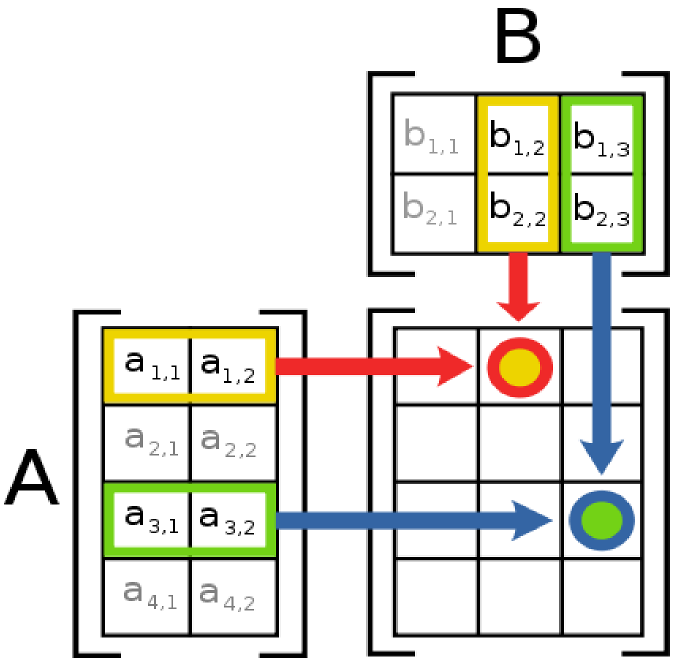
$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times 3 = \begin{bmatrix} 3a & 3b \\ 3c & 3d \end{bmatrix}$$

Matrix Multiplication

- Let X be an *a*x*b* matrix, Y be an *b*x*c* matrix
- Then Z = X*Y is an *a*x*c* matrix
- Second dimension of first matrix, and first dimension of first matrix have to be the same, for matrix multiplication to be possible
- Practice: Let X be an 10x5 matrix. Let's factorize it into 3 matrices…

Matrix Multiplication

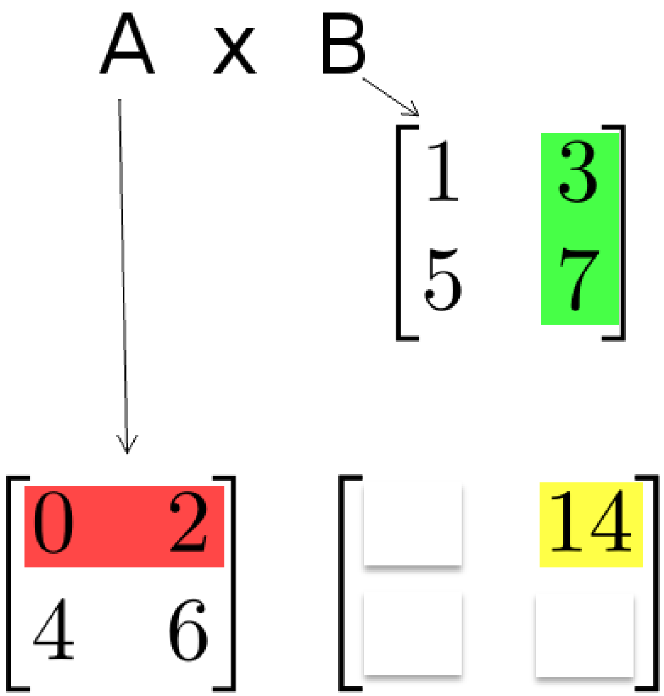
- The product AB is:



- Each entry in the result is (that row of A) dot product with (that column of B)

Matrix Multiplication

- Example:



$$0 \cdot 3 + 2 \cdot 7 = 14$$

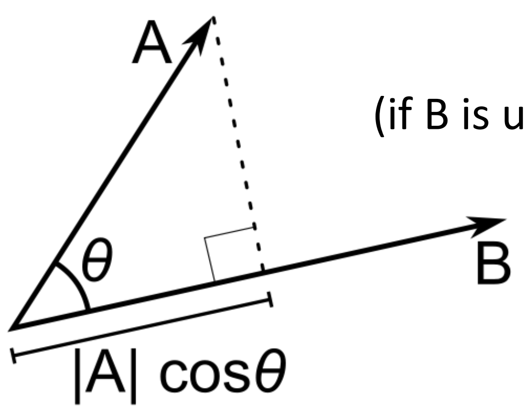
  – Each entry of the matrix product is made by taking the dot product of the corresponding row in the left matrix, with the corresponding column in the right one.

Inner Product

- Multiply corresponding entries of two vectors and add up the result

$$\mathbf{x}^T\mathbf{y} = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^{n} x_i y_i \qquad \text{(scalar)}$$

- x·y is also |x||y|Cos(*angle between x and y*)
- If B is a unit vector, then A·B gives the length of A which lies in the direction of B (projection)



(if B is unit-length hence norm is 1)

Different Types of Product

- $\boldsymbol{x}, \boldsymbol{y}$ = column vectors (nx1)
- $\boldsymbol{X}, \boldsymbol{Y}$ = matrices (mxn)
- x, y = scalars (1x1)

- $\boldsymbol{x}^T \boldsymbol{y} = \boldsymbol{x} \cdot \boldsymbol{y}$ = inner product (1xn x nx1 = scalar)
- $\boldsymbol{x} \otimes \boldsymbol{y} = \boldsymbol{x}\, \boldsymbol{y}^T$ = outer product (nx1 x 1xn = matrix)

- $\boldsymbol{X} * \boldsymbol{Y}$ = matrix product
- $\boldsymbol{X} .* \boldsymbol{Y}$ = element-wise product

Inverse

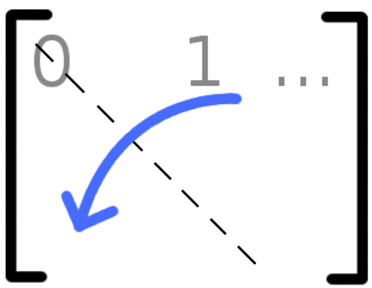
- Given a matrix **A**, its inverse $\mathbf{A}^{-1}$ is a matrix such that $\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$

- E.g. $\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{3} \end{bmatrix}$

- Inverse does not always exist. If $\mathbf{A}^{-1}$ exists, **A** is *invertible* or *non-singular*. Otherwise, it's *singular*.

Matrix Operations

- Transpose – flip matrix, so row 1 becomes column 1



$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix}^T = \begin{bmatrix} 0 & 2 & 4 \\ 1 & 3 & 5 \end{bmatrix}$$

- A useful identity:

$$(ABC)^T = C^T B^T A^T$$

Norms

- L1 norm

$$\|\boldsymbol{x}\|_1 := \sum_{i=1}^{n} |x_i|$$

- L2 norm

$$\|\boldsymbol{x}\| := \sqrt{x_1^2 + \cdots + x_n^2}$$

- L^p norm (for real numbers $p \geq 1$)

$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

Matrix Rank

- Column/row rank

$$\text{col-rank}(\mathbf{A}) = \text{ the maximum number of linearly independent column vectors of } \mathbf{A}$$
$$\text{row-rank}(\mathbf{A}) = \text{ the maximum number of linearly independent row vectors of } \mathbf{A}$$

- Column rank always equals row rank
- Matrix rank $\text{rank}(\mathbf{A}) \triangleq \text{col-rank}(\mathbf{A}) = \text{row-rank}(\mathbf{A})$
- If a matrix is not full rank, inverse doesn’t exist
  – Inverse also doesn’t exist for non-square matrices

Matrix Operation Properties

- Matrix addition is commutative and associative
  - A + B = B + A
  - A + (B + C) = (A + B) + C
- Matrix multiplication is associative and distributive but *not* commutative
  - A(B*C) = (A*B)C
  - A(B + C) = A*B + A*C
  - A*B != B*A

Special Matrices

- Identity matrix **I**
  - Square matrix, 1's along diagonal, 0's elsewhere
  - **I** · [another matrix] = [that matrix]

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Diagonal matrix
  - Square matrix with numbers along diagonal, 0's elsewhere
  - A diagonal · [another matrix] scales the rows of that matrix

$$\begin{bmatrix} 3 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 2.5 \end{bmatrix}$$

Special Matrices

- Symmetric matrix

$$\mathbf{A}^T = \mathbf{A}$$

$$\begin{bmatrix} 1 & 2 & 5 \\ 2 & 1 & 7 \\ 5 & 7 & 1 \end{bmatrix}$$

Matrix Operations

- MATLAB example:

$$AX = B$$

$$A = \begin{bmatrix} 2 & 2 \\ 3 & 4 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

```
>> x = A\B
x =
     1.0000
    -0.5000
```

Matlab

Matlab tutorial

http://www.cs.pitt.edu/~kovashka/cs1674_fa18/tutorial.m
http://www.cs.pitt.edu/~kovashka/cs1674_fa18/myfunction.m
http://www.cs.pitt.edu/~kovashka/cs1674_fa18/myotherfunction.m

Please cover whatever we don't finish at home.

Other tutorials and exercises

- https://people.cs.pitt.edu/~milos/courses/cs2750/Tutorial/
- http://www.math.udel.edu/~braun/M349/Matlab_probs2.pdf
- http://www.facstaff.bucknell.edu/maneval/help211/basicexercises.html
  - Do Problems 1-8, 12
  - Most also have solutions
  - Ask the TA if you have any problems