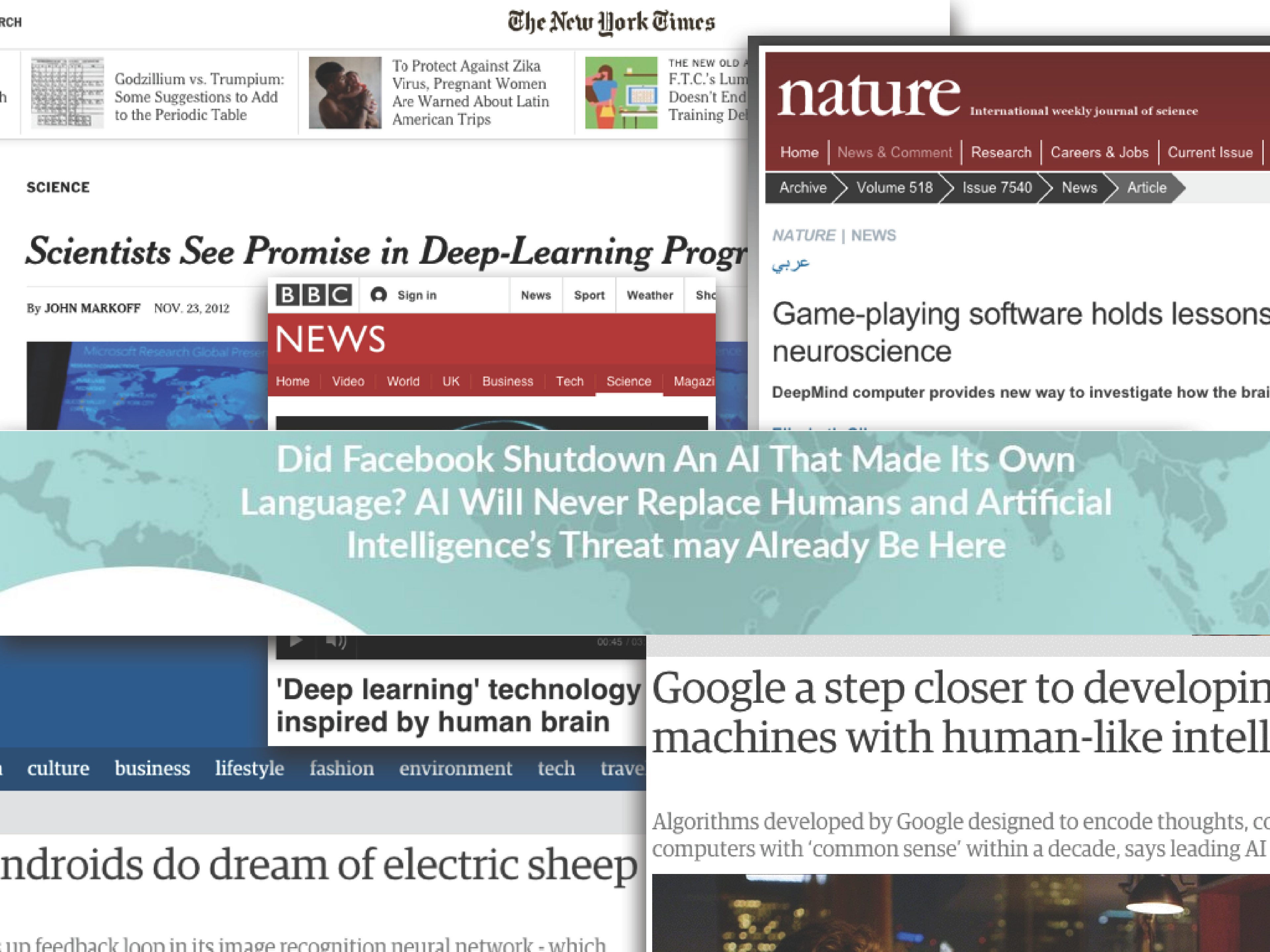
# CS194/294-129: Designing, Visualizing and Understanding Deep Neural Networks

**John Canny**

Spring 2018

CN: 194-129: 41752, 294-129: 41751



The New York Times
Godzillium vs. Trumpium: Some Suggestions to Add to the Periodic Table
To Protect Against Zika Virus, Pregnant Women Are Warned About Latin American Trips
SCIENCE
Scientists See Promise in Deep-Learning Progr
By JOHN MARKOFF NOV. 23, 2012
BBC
Sign in
News
Sport
Weather
NEWS
Home
Video
World
UK
Business
Tech
Science
nature
International weekly journal of science
Home
News & Comment
Research
Careers & Jobs
Current Issue
Archive
Volume 518
Issue 7540
News
Article
NATURE | NEWS
عربي
Game-playing software holds lessons
neuroscience
DeepMind computer provides new way to investigate how the brai
Did Facebook Shutdown An AI That Made Its Own Language? AI Will Never Replace Humans and Artificial Intelligence's Threat may Already Be Here
'Deep learning' technology inspired by human brain
culture
business
lifestyle
fashion
environment
tech
ndroids do dream of electric sheep
Google a step closer to developin
machines with human-like intell
Algorithms developed by Google designed to encode thoughts, co
computers with 'common sense' within a decade, says leading AI

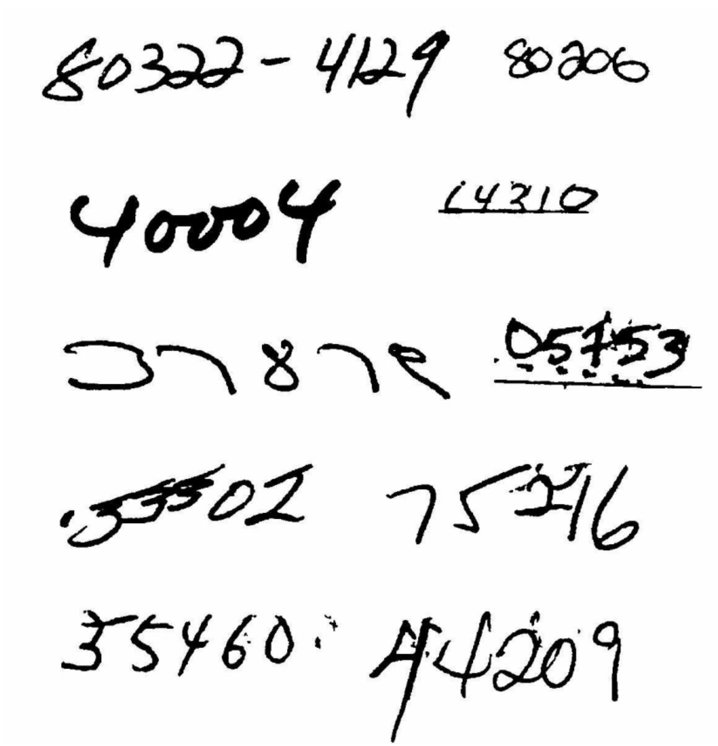
# Deep Learning: Hype or Hope?

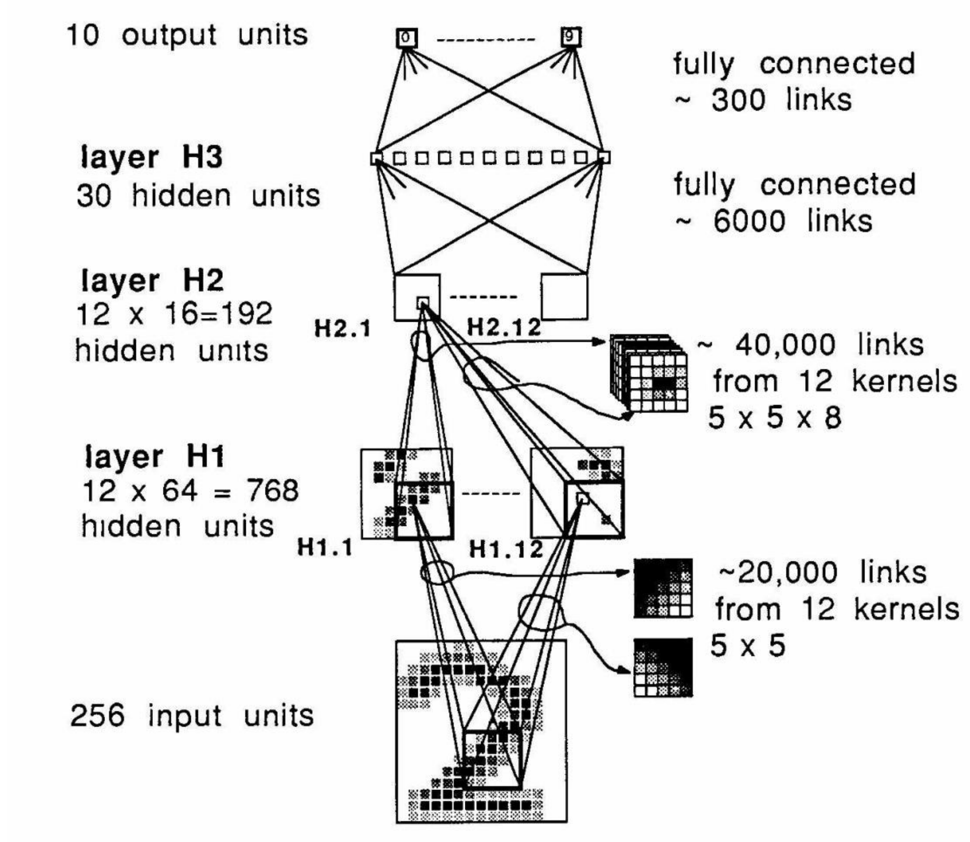
Hype: (n) “extravagant or intensive publicity or promotion”

Hope: (n) “expectation of fulfillment or success”

# Milestones: Digit Recognition

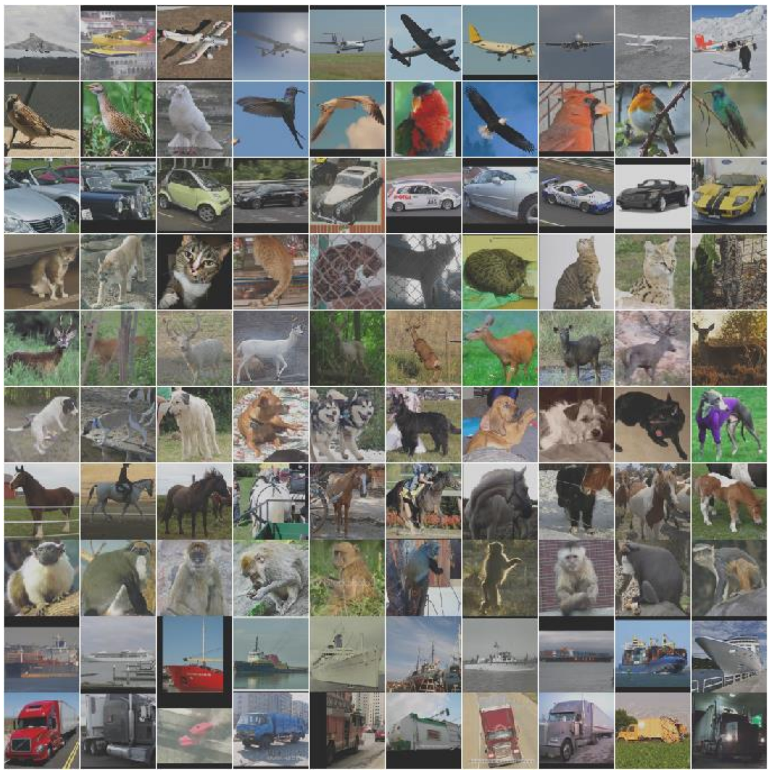
LeNet 1989: recognize zip codes, Yann Lecun, Bernhard Boser and others, ran live in US postal service

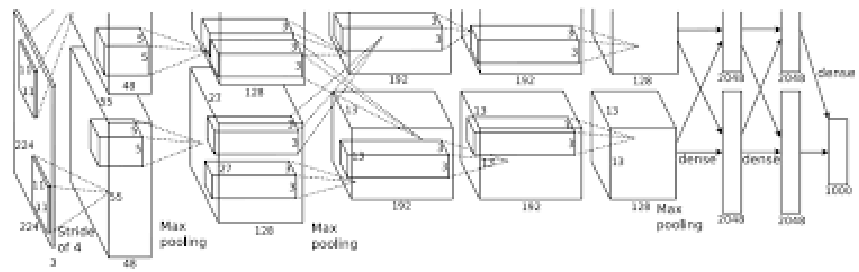


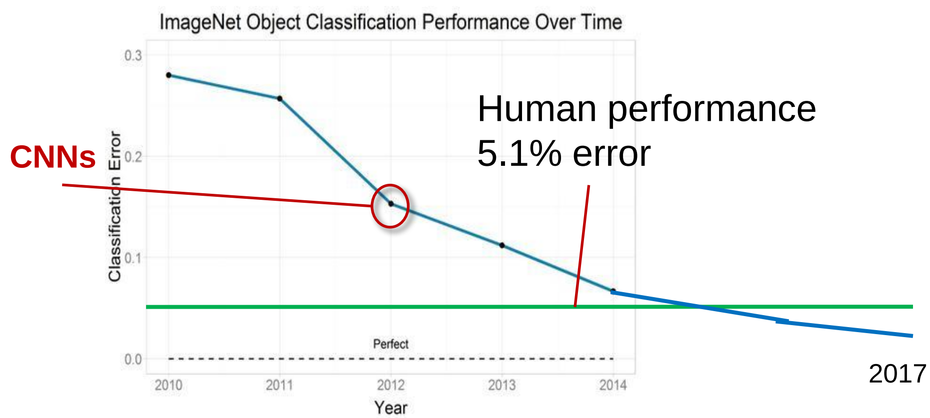


# Milestones: Image Classification

Convolutional NNs: AlexNet (2012): trained on 200 GB of ImageNet Data

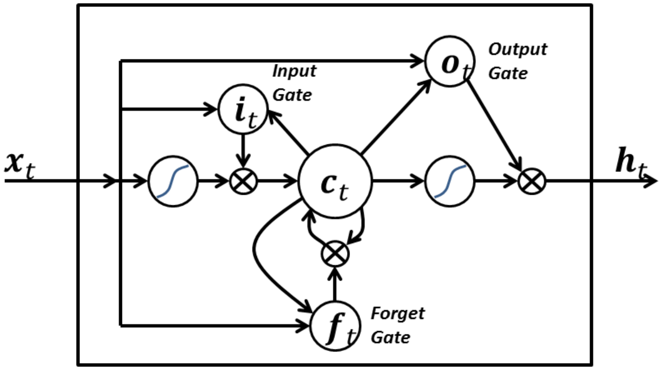


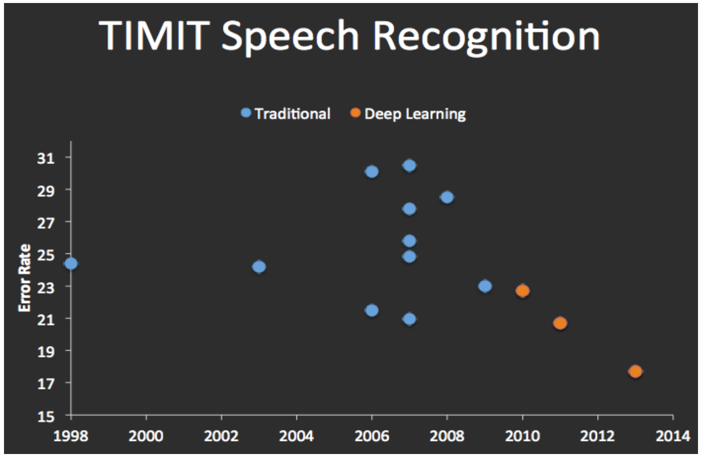


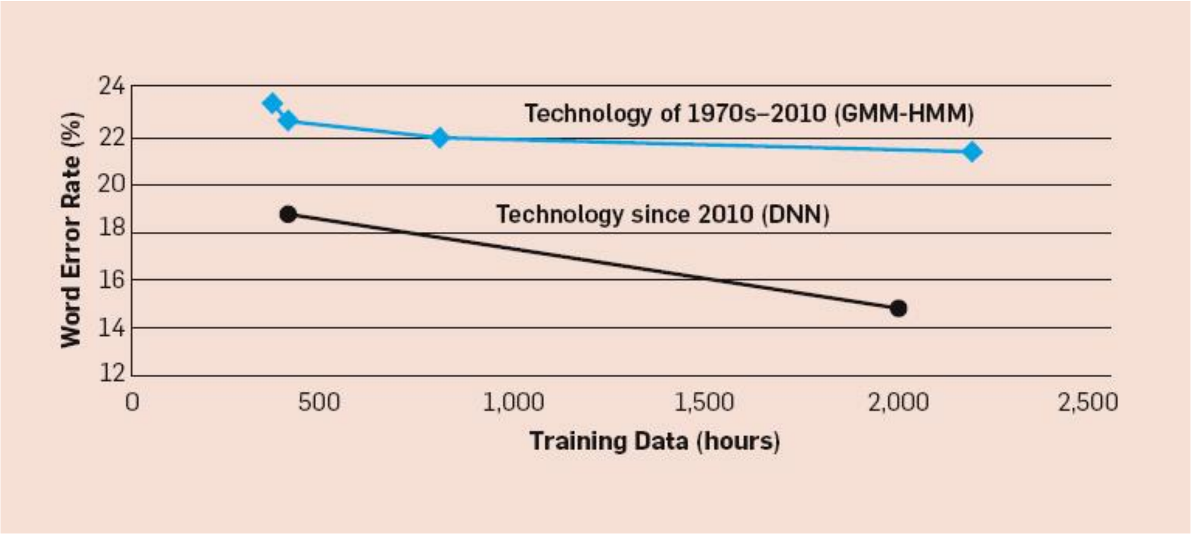



# Milestones: Speech Recognition

## Recurrent Nets: LSTMs (1997):

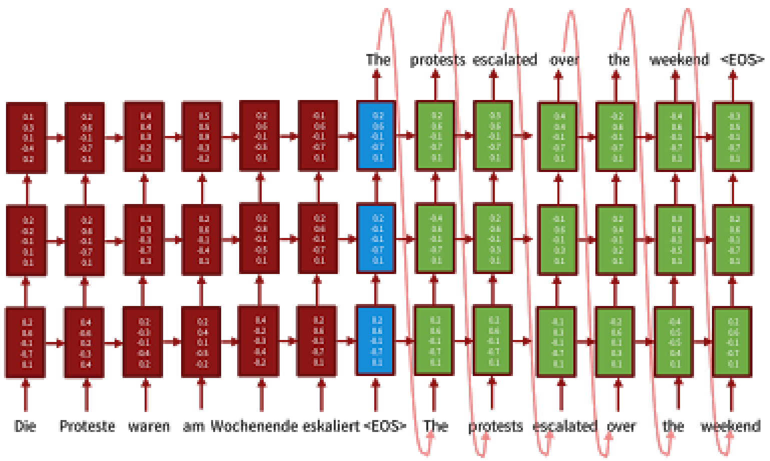


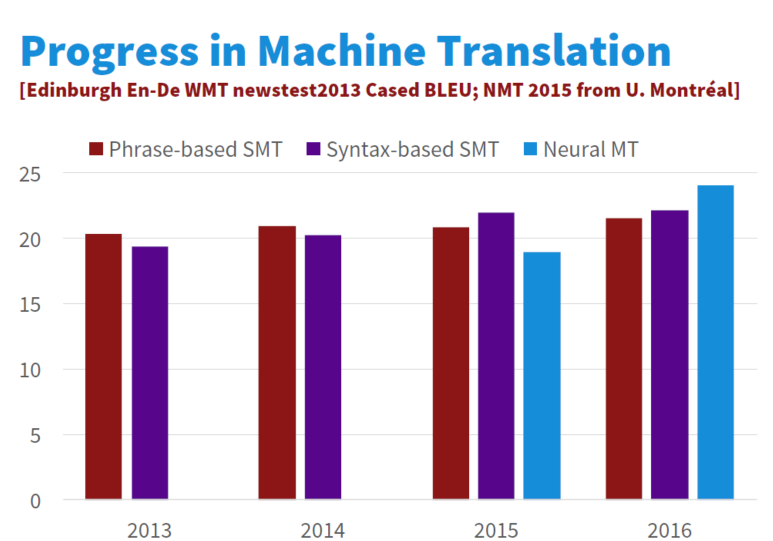



# Milestones: Language Translation

Sequence-to-sequence models with LSTMs and attention:




From [Sennrich 2016, http://www.meta-net.eu/events/meta-forum-2016/slides/09_sennrich.pdf]

Source Luong, Cho, Manning ACL Tutorial 2016.

# Neural Text Processing 2017

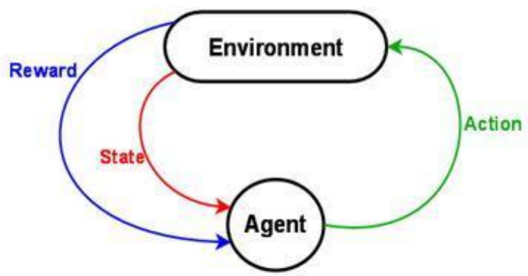
## State of the art results on NLP application-level tasks

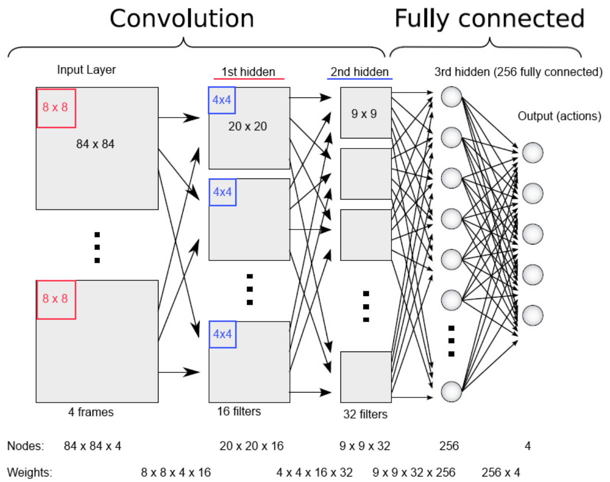
| Task | Test set | Metric | Best non-neural | Best neural | Source |
|---|---|---|---|---|---|
| **Machine Translation** | Enu-deu newstest16 | BLEU | 31.4 | 34.8 | http://matrix.statmt.org |
| | Deu-enu newstest16 | BLEU | 35.9 | 39.9 | http://matrix.statmt.org |
| **Sentiment Analysis** | Stanford sentiment bank | 5-class Accuracy | 71.0 | 80.7 | Socher+ 13 |
| **Question Answering** | WebQuestions test set | F1 | 39.9 | 52.5 | Yih+ 15 |
| **Entity Linking** | Bing Query Entity Linking set | AUC | 72.3 | 78.2 | Gao+ 14b |
| **Image Captioning** | COCO 2015 challenge | Turing test pass% | 25.5 | 32.2 | Fang+ 15 |
| **Sentence compression** | Google 10K dataset | F1 | 0.75 | 0.82 | Fillipova+ 15 |
| **Response Generation** | Sordoni dataset | BLEU-4 | 3.98 | 5.82 | Li+ 16a |

[Menezes & Dolan 17]

# Milestones: Deep Reinforcement Learning

In 2013, Deep Mind's  arcade player bests human expert on six Atari Games. Acquired by Google in 2014,.




In 2016, Deep Mind's alphaGo defeats former world champion Lee Sedol

# Deep Learning: Is it Hype or Hope?

# Deep Learning: Is it Hype or Hope?

Yes !

# Critiques

## How smart is today's artificial intelligence?

Today's AI is super impressive, but it's not intelligent.

By Joss Fong | joss@vox.com | Dec 19, 2017, 9:40am EST

INFOWORLD TECH WATCH

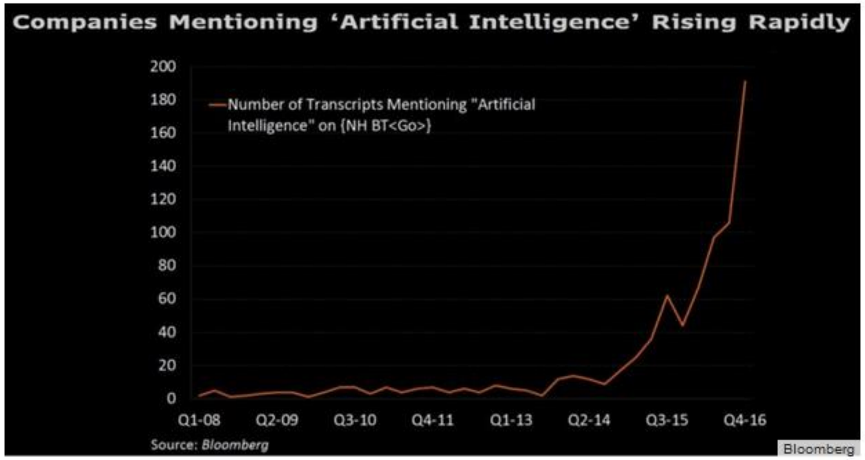
By Matt Asay, InfoWorld | MAR 3, 2017

About

Informed news analysis every weekday

## Artificially inflated: It's time to call BS on AI

We may have hit peak ludicrous mode for AI, flailing in a tsunami of AI-washing




## Is AI Overhyped?

Forbes Technology Council

*Elite CIOs, CTOs & execs offer firsthand insights on tech & business.* FULL BIO

Opinions expressed by Forbes Contributors are their own.

POST WRITTEN BY

Ken Weiner

# Critiques

## How smart is today's artificial intelligence?

Today's AI is super impressive, but it's not intelligent.

By Joss Fong | joss@vox.com | Dec 19, 2017, 9:40am EST

INFOWORLD TECH WATCH

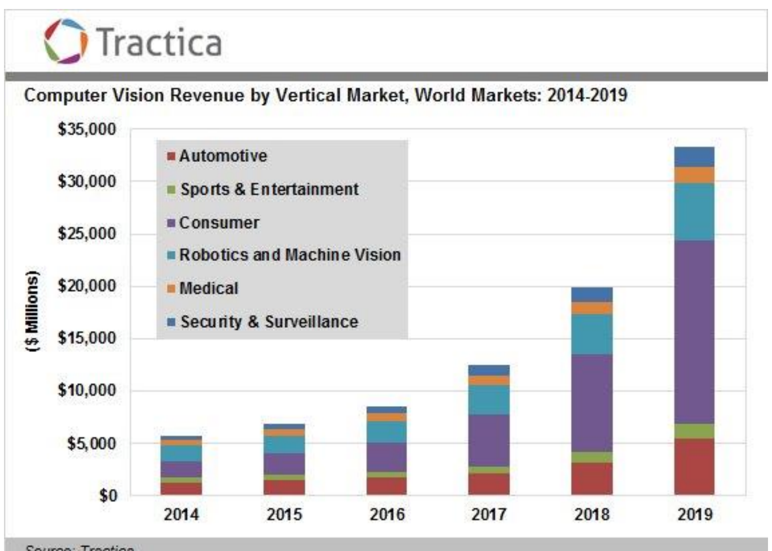
By Matt Asay, InfoWorld | MAR 3, 2017

About

Informed news analysis every weekday

## Artificially inflated: It's time to call BS on AI

We may have hit peak ludicrous mode for AI, flailing in a tsunami of AI-washing




## Is AI Overhyped?

Forbes Technology Council

*Elite CIOs, CTOs & execs offer firsthand insights on tech & business.* FULL BIO

Opinions expressed by Forbes Contributors are their own.

POST WRITTEN BY

Ken Weiner

# Setting Expectations … Badly

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".

# Critiques:

(Some) AI Scientists made wildly optimistic predictions about AI system capabilities in the 80s and 90s leading to an “AI winter.” So optimistic predictions about AI systems today should be ignored…

AI systems are still not “generally intelligent,” … even if they are awfully good at an awful lot of things.

# Critiques:

(Some) AI Scientists made over-optimistic predictions about AI system capabilities in the 80s and 90s leading to an “AI winter.” So optimistic predictions about AI systems today should be ignored…

- In fact, those folks ignored all the data about AI system performance on real problems, and generalized instead from artificial ones.
- The assertion depends on either
  - AI systems → intrinsically poor performance
  - AI researcher → always over-estimate

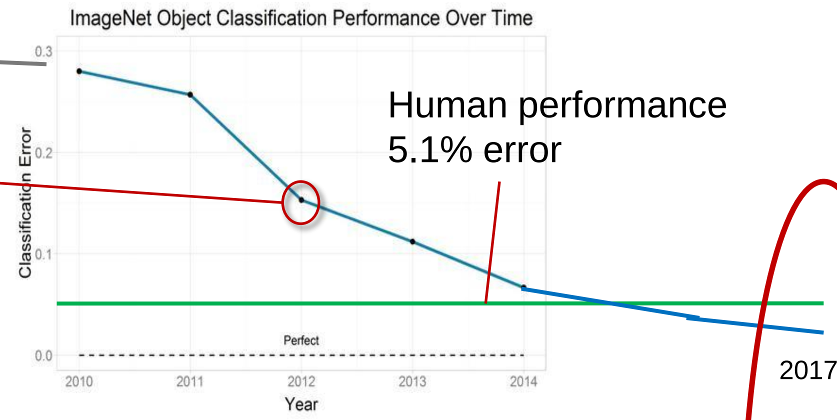
AI systems are still not “generally intelligent,” … even if they are awfully good at an awful lot of things.

- As a concept “general intelligence” has been the bane of AI research.
- Progress in AI has recently been dictated mostly by economic importance, not “AI-hardness”. The “AI-hard” list shrinks every year: Jeopardy, Go, Science Exams, Face recognition, Translation,…

# Data on Classical AI vs. Deep Learning:

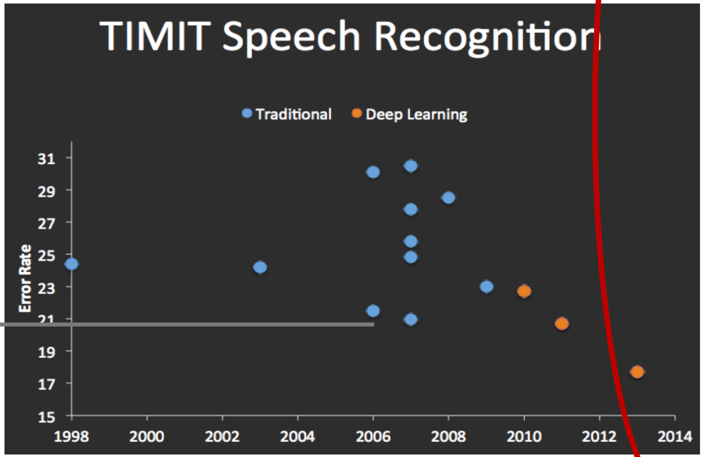
Performance floor

CNNs




Human performance
5.1% error

2017




Performance floor

Where’s the floor ?

# Opportunities:

Of course!

- Science, engineering, entertainment, education, communication, organization, recreation, medicine, driving, games (real and virtual), transportation, commerce, e-trading, name-your-topic…

# Risks:

Yes!

- Economic: displacing jobs

- Existential: security, systems running amok

Hawking, Musk, Gates have been highlighting the risks of new AI technologies.

# Learning about Deep Neural Networks

Yann Lecun (Facebook research head, DNN pioneer) quote: DNNs require:
*“an interplay between intuitive insights, theoretical modeling, practical implementations, empirical studies, and scientific analyses”*

i.e. there isn’t a single framework or core set of principles to explain everything (c.f. graphical models for machine learning).

We try to cover the ground in Lecun’s quote.

# This Course (please interrupt with questions)

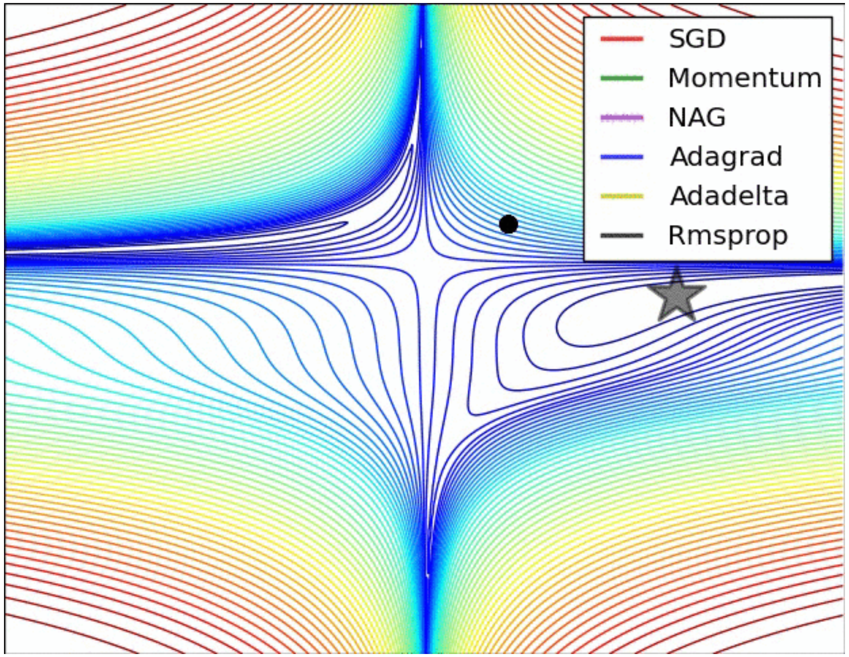
Goals:

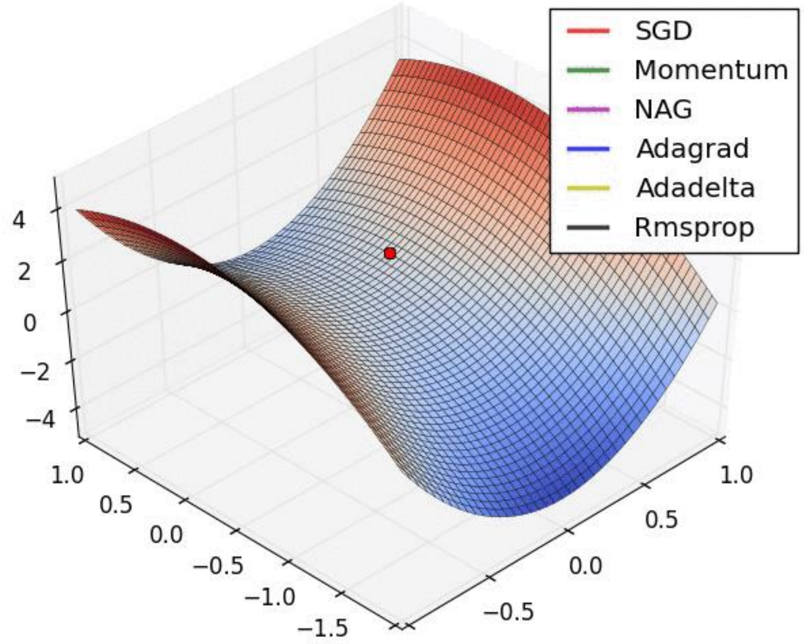
- Introduce deep learning to a broad audience.
- Review principles, techniques and visualization for understanding deep networks.
- Develop skill at designing networks for applications.

Materials:

- Book(s)
- Notes
- Lectures

# The role of Animation

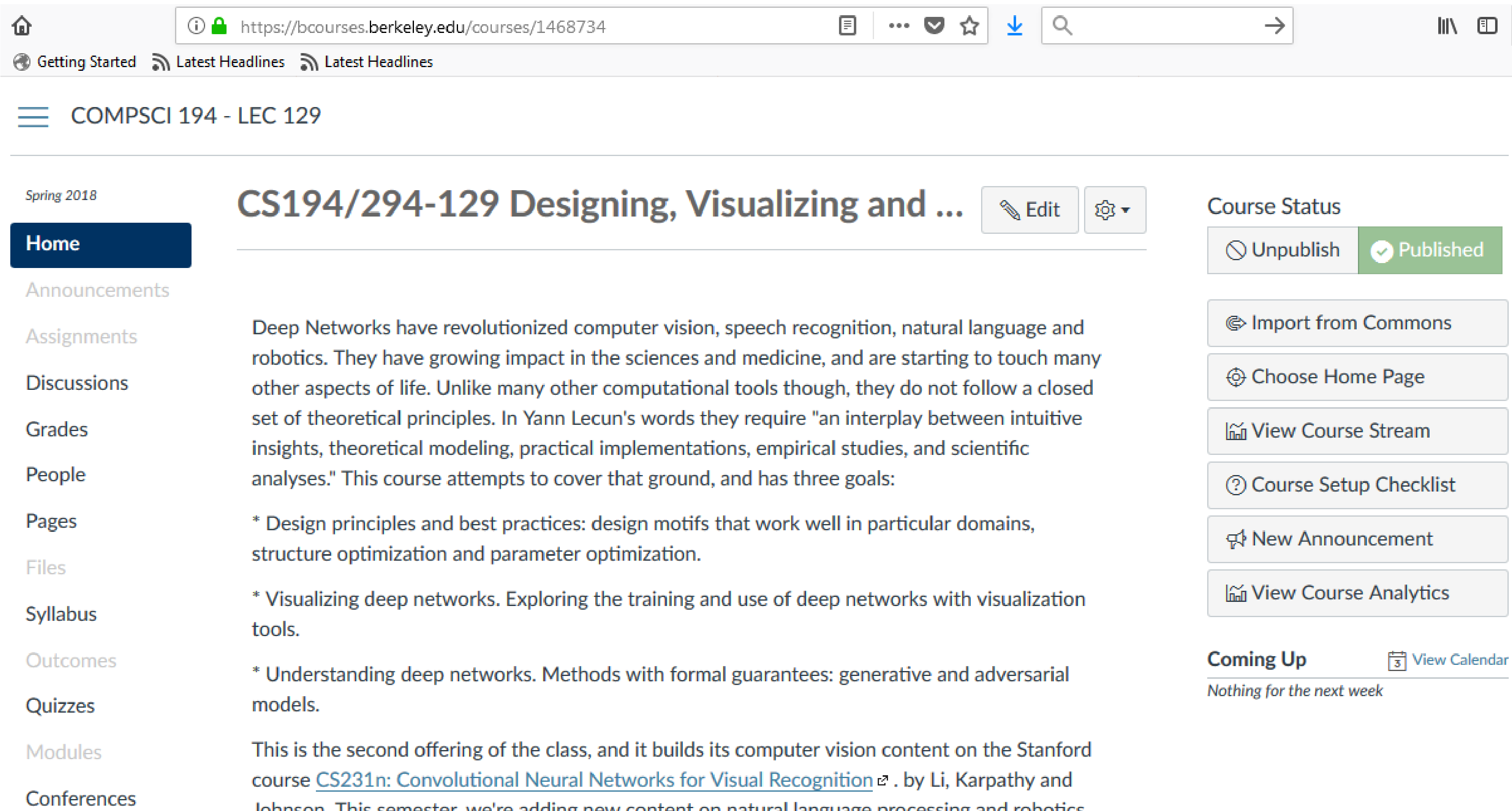



From A. Karpathy's cs231n notes.

# This Course: bCourses page

Please sign in to the bCourses, you should see the course page. Its also linked from my home page.



# bCourses main page has the schedule

## Schedule

| Date | Lecture Topic | Reading | Assignments |
|---|---|---|---|
| W 1/17 | Introduction to Deep Learning and Applications, Course Overview. | Introduction from Deep Learning | Assignment 0: Discussion time poll. |
| M 1/22 | Brief History of Computer Vision, Classification, k-Nearest Neighbors. | Review: chapters 1-4 of Deep Learning and do Python/Numpy tutorial if needed. Image Classification Notes | |
| W 1/24 | Linear Classification, Feature selection. | Linear Classification Notes | |
| M 1/29 | Optimization, Stochastic Gradient Descent. | Chapter 8 of Deep Learning Optimization Notes | |
| W 1/31 | Backpropagation. | Backpropagation Notes | |
| M 2/5 | Training DNNs 1: activation functions, initialization, gradient flow, batch normalization. | Training Neural Networks 1 Training Neural Networks 2 | |
| W 2/7 | Training DNNs 2: parameter updates, ensembles, dropout. | Training Neural Networks 3 | Assignment 1 due |

# bCourses outline (home page)

Computer Vision, General DNN principles

Deep nets and natural language

Imitation and reinforcement learning

## Schedule

| Date | Lecture Topic | Reading | Assignments |
|---|---|---|---|
| W 1/17 | Introduction to Deep Learning and Applications, Course Overview. | Introduction from Deep Learning | Assignment 0: Discussion time poll. |
| M 1/22 | Brief History of Computer Vision, Classification, k-Nearest Neighbors. | Review: chapters 1-4 of Deep Learning and do Python/Numpy tutorial if needed. Image Classification Notes | |
| W 1/24 | Linear Classification, Feature selection. | Linear Classification Notes | |
| M 1/29 | Optimization, Stochastic Gradient Descent. | Chapter 8 of Deep Learning Optimization Notes | |
| W 1/31 | Backpropagation. | Backpropagation Notes | |
| M 2/5 | Training DNNs 1: activation functions, initialization, gradient flow, batch normalization. | Training Neural Networks 1 Training Neural Networks 2 | |
| W 2/7 | Training DNNs 2: parameter updates, ensembles, dropout. | Training Neural Networks 3 | Assignment 1 due |
| M 2/12 | Convolutional Networks: convolution/pooling layers, network design, theory. | Convnet notes | |
| W 2/14 | Convnets for Classification and Detection. Visual challenge datasets. | Convnet notes | Project proposal due |
| M 2/19 | Academic Holiday | | |
| W 2/21 | Visualizing Deep Networks. | Quite a few visualizations will be covered. Browse this list . | |
| M 2/26 | Midterm 1. | | |
| W 2/28 | Generative Adversarial Networks. | Generative Adversarial Networks | Assignment 2 due |
| M 3/5 | Recurrent Networks, LSTMs, applications. | RNN chapter from Deep Learning | |
| W 3/7 | TBD Attention networks. | Recurrent Models of Visual Attention | |
| M 3/12 | Semantic Models for Text. | WordToVec Skip-Thought Vectors | |
| W 3/14 | Natural Language Translation. | Neural Machine Translation by Jointly Learning to Align and Translate | |
| F 3/16 | | | Project Checkpoint 1 due |
| M 3/19 | Memory Networks, Text Question-Answering Systems. | | |
| W 3/21 | Chatbots and Task-Completion Agents. | | Assignment 3 due |
| M 3/26 | Spring Break | | |
| W 3/28 | Spring Break | | |
| M 4/2 | Adversarial Networks Dawn Song and Bo Li Guest lecture | | |
| W 4/4 | Imitation Learning. | | |
| M 4/9 | Midterm 2 | | |
| W 4/11 | Reinforcement Learning: Policy Gradients. | | |
| M 4/16 | Reinforcement Learning: Value-based methods. | | |
| W 4/18 | Imagination and Curiosity. | | |
| M 4/23 | Learning to Learn. Chelsea Finn Guest Lecture. | | |
| W 4/25 | Seeing, Talking and Acting. | | Assignment 4 due |
| F 4/27 | | | Final Project Presentation due |
| M 4/30 | Final project presentations I | | |
| W 5/2 | Final project presentations II | | |
| TBD 4/30-5/4 | Final project poster session | | Final Project Poster due |
| TBD | | | Final Project Report due |

# bCourses outline (home page)

Schedule

| Date | Lecture Topic | Reading | Assignments |
|---|---|---|---|
| W 1/17 | Introduction to Deep Learning and Applications, Course Overview. | Introduction from Deep Learning | Assignment 0: Discussion time poll. |
| M 1/22 | Brief History of Computer Vision, Classification, k-Nearest Neighbors. | Review: chapters 1-4 of Deep Learning and do Python/Numpy tutorial if needed. Image Classification Notes | |
| W 1/24 | Linear Classification, Feature selection. | Linear Classification Notes | |
| M 1/29 | Optimization, Stochastic Gradient Descent. | Chapter 8 of Deep Learning Optimization Notes | |
| W 1/31 | Backpropagation. | Backpropagation Notes | |
| M 2/5 | Training DNNs 1: activation functions, initialization, gradient flow, batch normalization. | Training Neural Networks 1 Training Neural Networks 2 | |
| W 2/7 | Training DNNs 2: parameter updates, ensembles, dropout. | Training Neural Networks 3 | Assignment 1 due |
| M 2/12 | Convolutional Networks: convolution/pooling layers, network design, theory. | Convnet notes | |
| W 2/14 | Convnets for Classification and Detection. Visual challenge datasets. | Convnet notes | Project proposal due |
| M 2/19 | Academic Holiday | | |
| W 2/21 | Visualizing Deep Networks. | Quite a few visualizations will be covered. Browse this list . | |
| M 2/26 | Midterm 1. | | |
| W 2/28 | Generative Adversarial Networks. | Generative Adversarial Networks | Assignment 2 due |
| M 3/5 | Recurrent Networks, LSTMs, applications. | RNN chapter from Deep Learning | |
| W 3/7 | TBD Attention networks. | Recurrent Models of Visual Attention | |
| M 3/12 | Semantic Models for Text. | WordToVec Skip-Thought Vectors | |
| W 3/14 | Natural Language Translation. | Neural Machine Translation by Jointly Learning to Align and Translate | |
| F 3/16 | | | Project Checkpoint 1 due |
| M 3/19 | Memory Networks, Text Question-Answering Systems. | | |
| W 3/21 | Chatbots and Task-Completion Agents. | | Assignment 3 due |
| M 3/26 | Spring Break | | |
| W 3/28 | Spring Break | | |
| M 4/2 | Adversarial Networks **Dawn Song and Bo Li Guest lecture** | | |
| W 4/4 | Imitation Learning. | | |
| M 4/9 | Midterm 2 | | |
| W 4/11 | Reinforcement Learning: Policy Gradients. | | |
| M 4/16 | Reinforcement Learning: Value-based methods. | | |
| W 4/18 | Imagination and Curiosity. | | |
| M 4/23 | Learning to Learn. **Chelsea Finn Guest Lecture.** | | |
| W 4/25 | Seeing, Talking and Acting. | | Assignment 4 due |
| F 4/27 | | | Final Project Presentation due |
| M 4/30 | Final project presentations I | | |
| W 5/2 | Final project presentations II | | |
| TBD 4/30-5/4 | Final project poster session | | Final Project Poster due |
| TBD | | | Final Project Report due |

Capstone Vision lecture

Capstone natural language lecture

Capstone robotics lecture

# This Course

Work:

- Class Participation: 10%
- 2 Midterms: 30%
- Final Project (in groups): 30%
- 4 Assignments : 30%

Audience: primarily EECS undergrads and grads.

This semester the class is a 194/294 (special topics). It doesn’t have centrally scheduled discussion sections, ***but…***

# Optional Discussion Sections

We will schedule 4x 1-hour discussion sections per week.

You wont formally register for these, and you can attend any one of them.

We need to schedule these to fit your schedules, and the GSIs availability, and room availability.

***Please fill out "Assignment 0" on bCourses by Thursday 1/18, 10pm.***

***No discussions this week but we would like to start them next week.***

# bCourses main page has the schedule

## Schedule

| Date | Lecture Topic | Reading | Assignments |
|---|---|---|---|
| W 1/17 | Introduction to Deep Learning and Applications, Course Overview. | Introduction from Deep Learning | Assignment 0: Discussion time poll. |
| M 1/22 | Brief History of Computer Vision, Classification, k-Nearest Neighbors. | Review: chapters 1-4 of Deep Learning and do Python/Numpy tutorial if needed. Image Classification Notes | |
| W 1/24 | Linear Classification, Feature selection. | Linear Classification Notes | |
| M 1/29 | Optimization, Stochastic Gradient Descent. | Chapter 8 of Deep Learning Optimization Notes | |
| W 1/31 | Backpropagation. | Backpropagation Notes | |
| M 2/5 | Training DNNs 1: activation functions, initialization, gradient flow, batch normalization. | Training Neural Networks 1 Training Neural Networks 2 | |
| W 2/7 | Training DNNs 2: parameter updates, ensembles, dropout. | Training Neural Networks 3 | Assignment 1 due |

# Discussion Section Poll

When filling out the poll, consider:

- If you rarely or never attend section, its best if you don't fill out the poll so we can match times to students who do.
- If you almost always attend sections, please place an asterisk next to your name in the poll so we can maximize coverage for those students who attend most of the time.
- Please fill out ***all*** the times you can make so we have maximum chance of satisfying everyone's constraints.
- The poll is only visible to instructors.
- ***Please complete the poll by 10pm, Thursday, Jan 18th.***

# Course Staff

John Canny

GSIs:

Erin Grant

Carlos Florensa

# Prerequisites

- Knowledge of calculus and linear algebra, Math 53/54 or equivalent.
- Probability and Statistics, CS70 or Stat 134. CS70 is bare minimum preparation, a stat course is better.
- Machine Learning: CS189, strongly encouraged but not required.
- Programming, CS61B or equivalent. Assignments will mostly use Python.

# Logistics

- Course Number: CS 194/294-129, SP 2018, UC Berkeley
- On bCourses, publicly readable.
- Course numbers 194-129: 41752, 294-129: 41751
- Instructor: John Canny lastname@berkeley.edu
- Time: MW 5pm - 6:30pm
- Location: 105 Northgate
- Discussion: Join Piazza for announcements and to ask questions about the course
- Office hours:
  - John Canny - M 2:00-3:00, in 637 Soda
- Webcasts (slides + audio) should be available from the bCourses page.

# Course Project

- Will consume about 2/3 of the semester.
- In teams of 3-4.
- Can be combined with other course projects
- We encourage “open-source” projects that can be archived.
- You will “check-in” with the GSIs several times during the semester.
- Final poster and report due at the end of the semester.

# Acknowledgement

- A grant from Google provided partial support for GSIs and readers. Thanks!

# Questions ?

Coming up: Some rationale for deep neural networks…

# The Waitlist:

The class is full for now.

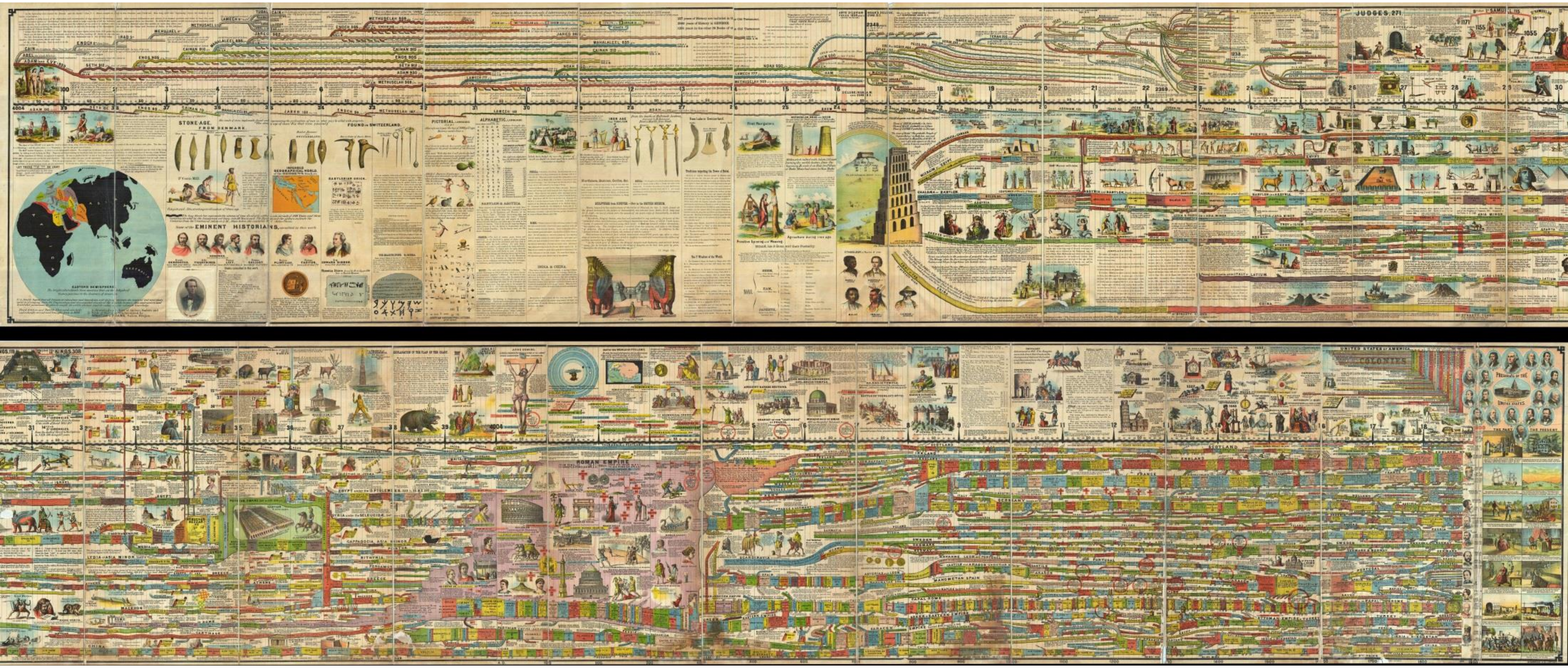
There may be some movement from the waitlist, but this is impossible to predict. Last time, there was less movement than normal…

This course should be 182/282A next time and fully-resourced and taught at larger scale.

To know more, consult a professional psychic…

# Some History

- Reading: the Deep Learning Book, Introduction



# Phases of Neural Network Research

- 1940s-1960s: Cybernetics: Brain like electronic systems, morphed into modern control theory and signal processing.
- 1960s-1980s: **D**igital computers, automata theory, computational complexity theory: simple shallow circuits are very limited in what they can represent…
- 1980s-1990s: Connectionism: complex, non-linear networks, back-propagation.
- 1990s-2010s: Computational learning theory, graphical models: Learning is computationally hard, simple shallow circuits are very limited in what they can learn…
- 2006→: Deep learning: End-to-end training, large datasets, explosion in applications.

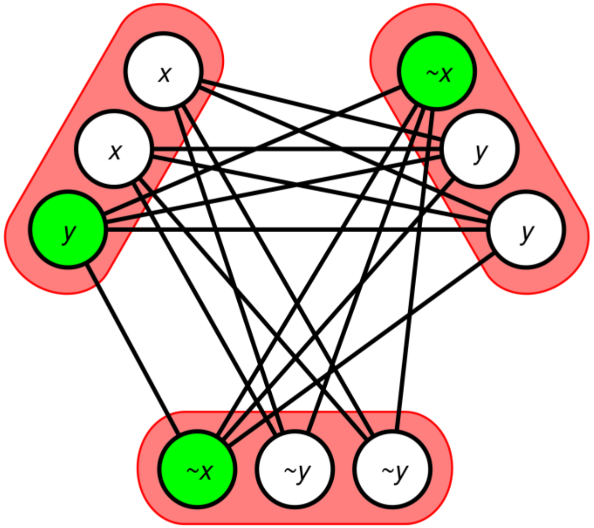
# Citations of the “LeNet” paper

- Recall the LeNet was a modern visual classification network that recognized digits for zip codes. Its citations look like this:


- The 2000s were a golden age for machine learning, and marked the ascent of graphical models. But not so for neural networks.

# Why the success of DNNs is surprising

- From both complexity and learning theory perspectives, simple networks are very limited.
  - Can't compute parity with a small network.
  - NP-Hard to learn "simple" functions like 3SAT formulae, and i.e. training a DNN is NP-hard.



# Why the success of DNNs is surprising

- The most successful DNN training algorithm is a version of gradient descent which will only find local optima. In other words, it's a greedy algorithm. Backprop:

$$\text{loss} = f(g(h(y)))$$

$$d\,\text{loss}/dy = f'(g) \times g'(h) \times h'(y)$$

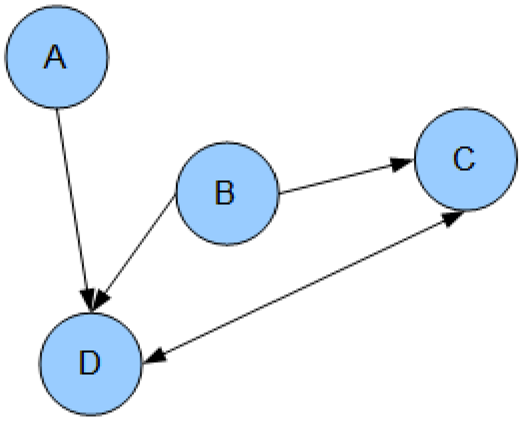
- Greedy algorithms are even more limited in what they can represent and how well they learn.

- If a problem has a greedy solution, its regarded as an "easy" problem.

# Why the success of DNNs is surprising

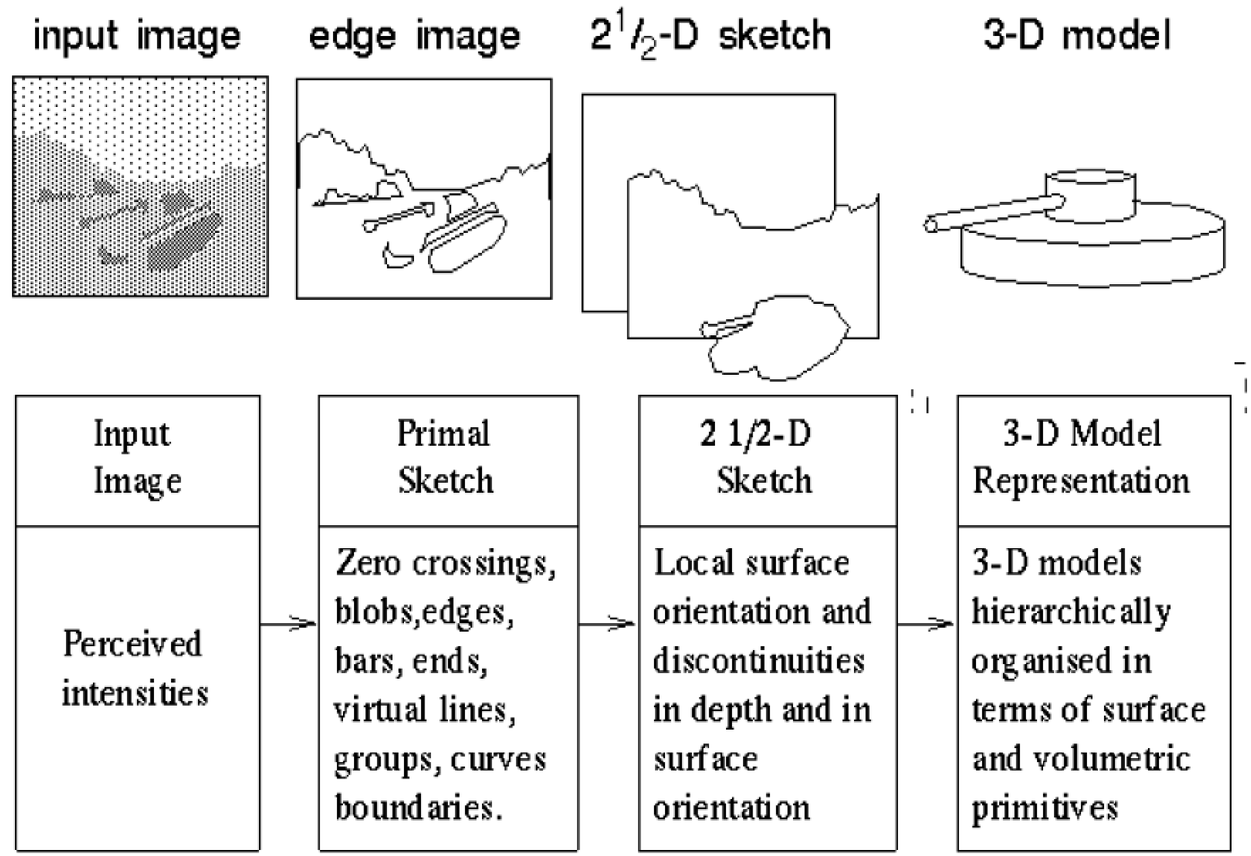
- In graphical models, values in a network represent random variables, and have a clear meaning. The network structure encodes dependency information, i.e. you can represent rich models.

- In a DNN, node activations encode nothing in particular, and the network structure only encodes (trivially) how they derive from each other.



# Why the success of DNNs is ~~surprising~~ obvious

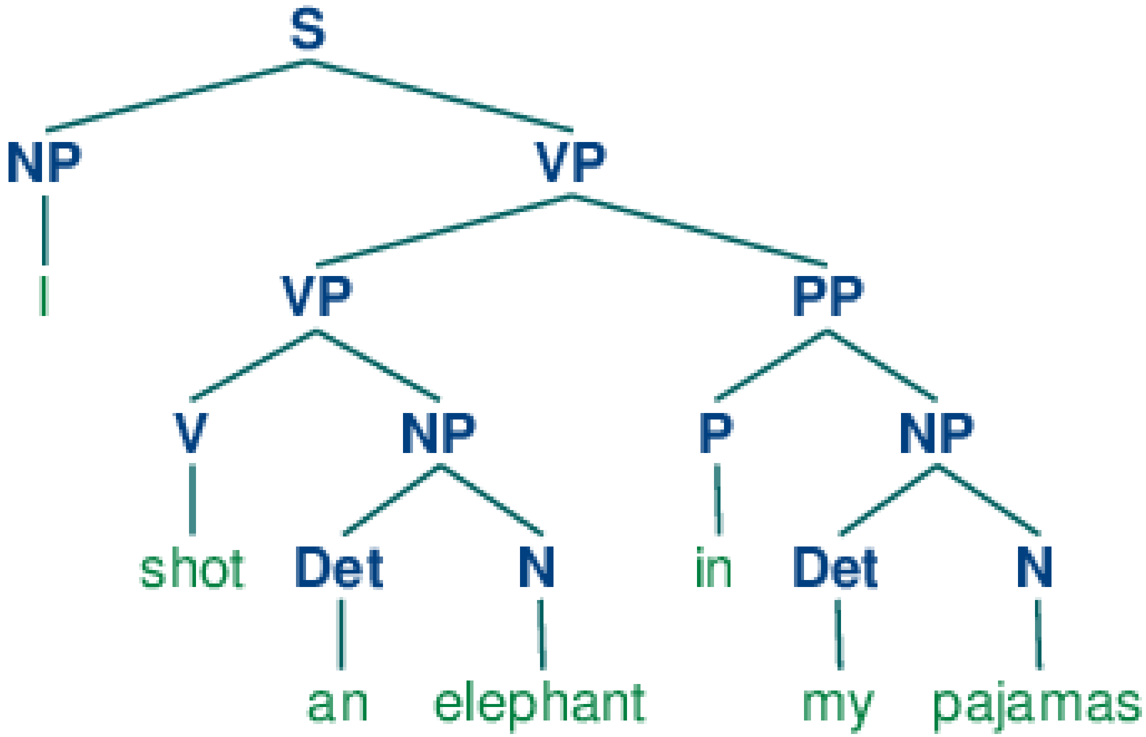
- Hierarchical representations are ubiquitous in AI. Computer vision:




Stages of Visual Representation, David Marr, 1970s

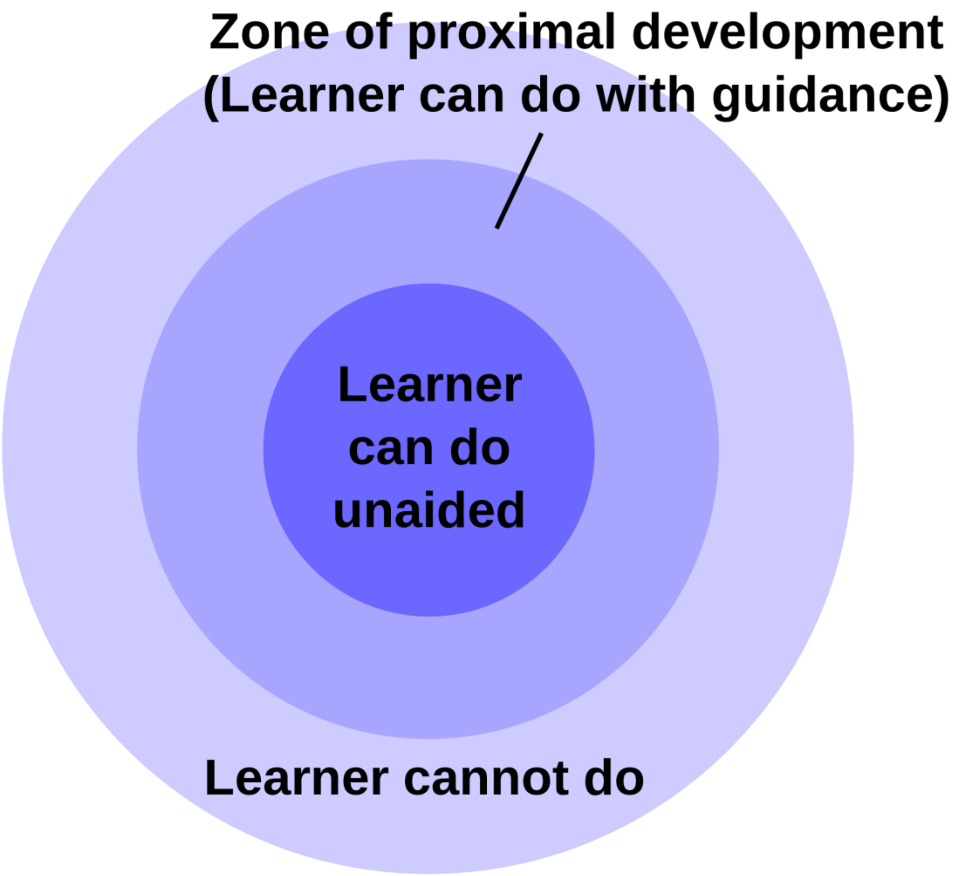
# Why the success of DNNs is ~~surprising~~ obvious

- Hierarchies are ubiquitous in natural language:



# Why the success of DNNs is ~~surprising~~ obvious

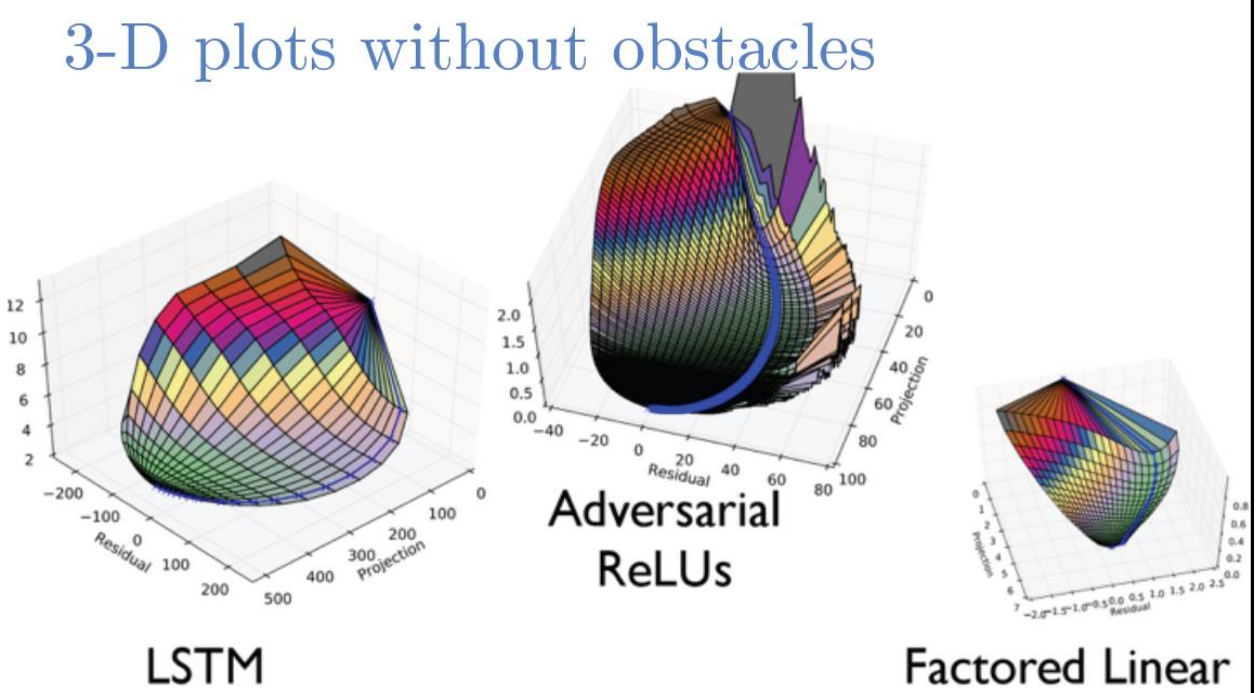
- Human Learning: is deeply layered.




Deep expertise

# Why the success of DNNs is ~~surprising~~ obvious

- What about greedy optimization?
- Less obvious, but it looks like many learning problems (e.g. image classification) are actually “easy” i.e. have reliable steepest descent paths to a good model.



# Questions ?

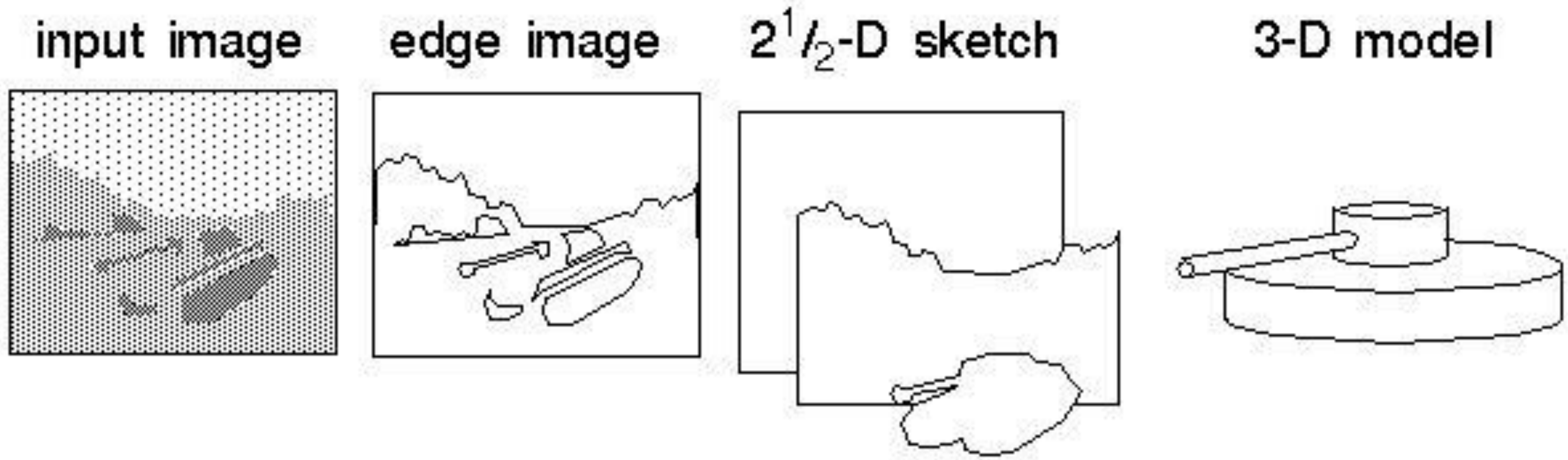
Coming up: Computer Vision intro…

# Computer Vision – a brief history

David Marr’s approach 1970-80s

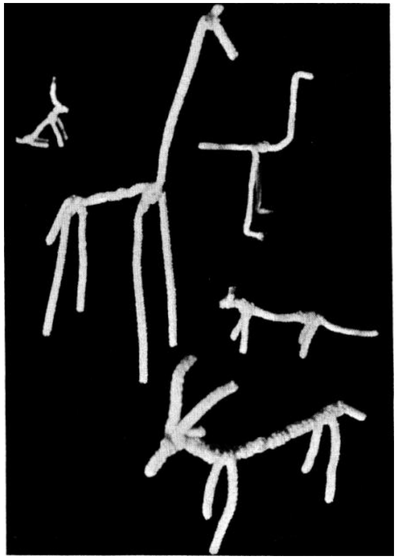
- Computation
- Algorithm
- Implementation

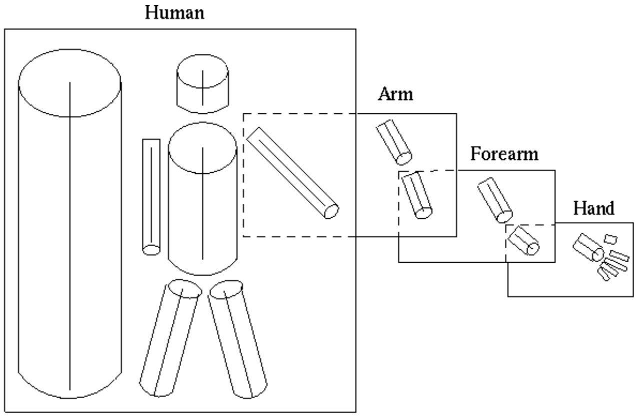
primal sketch and
2 ½ D sketch,



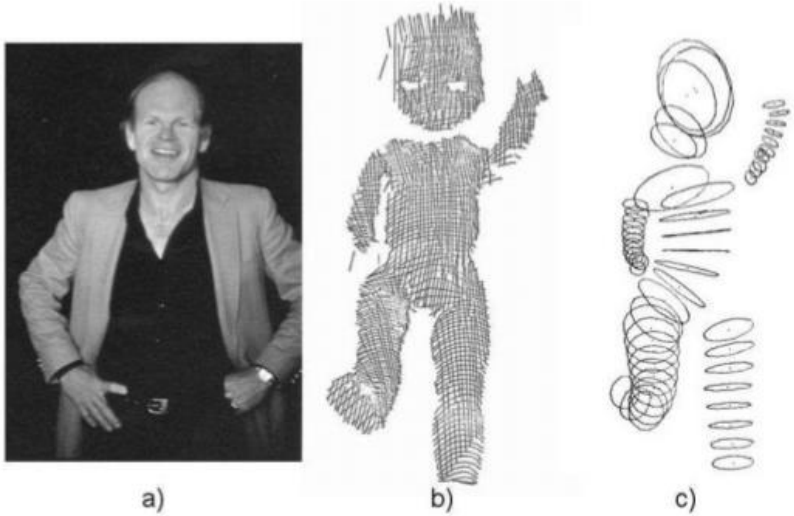
# Computer Vision - Shape description

Skeletons and Cylinders: Marr and Nishihara 1978:



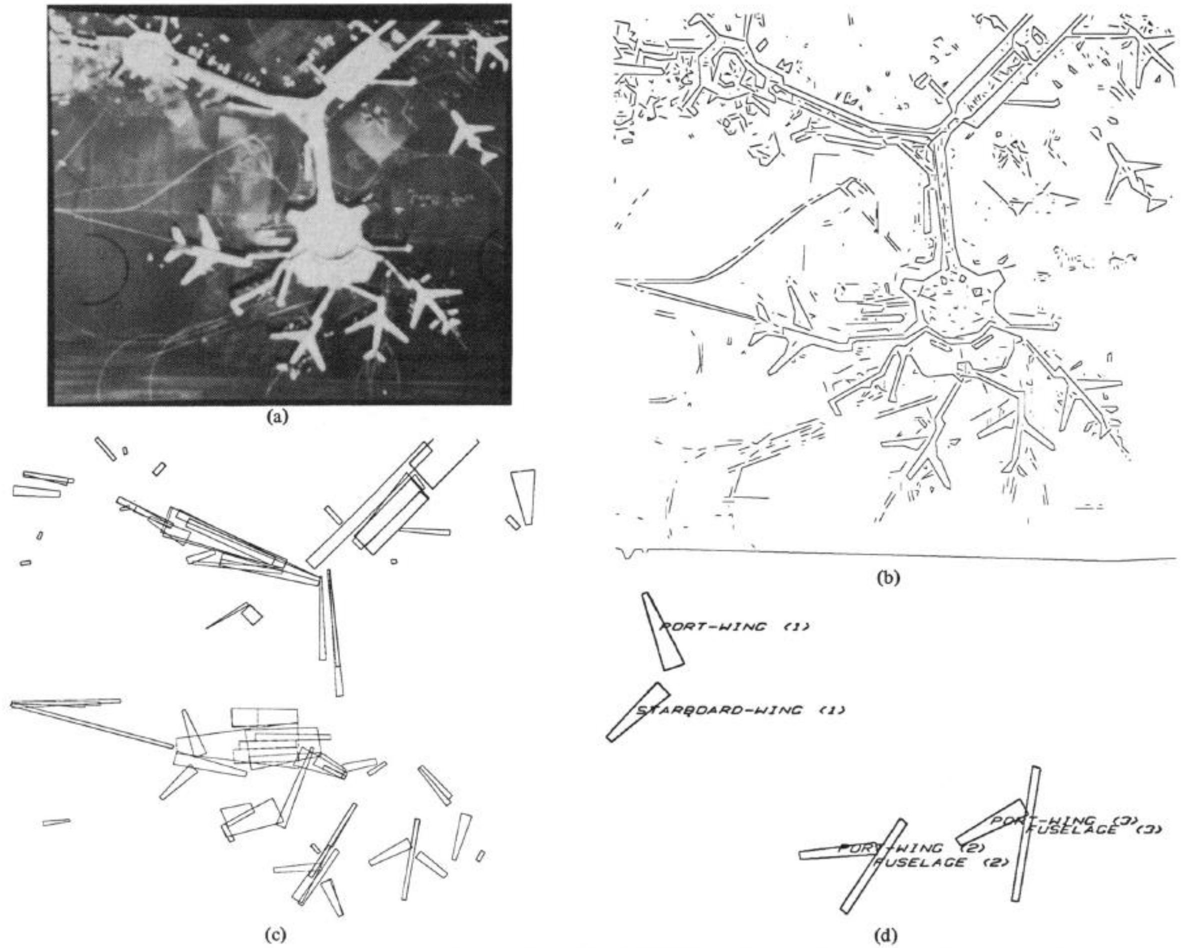



Tom Binford, generalized cylinders 1971



# Computer Vision - Shape matching

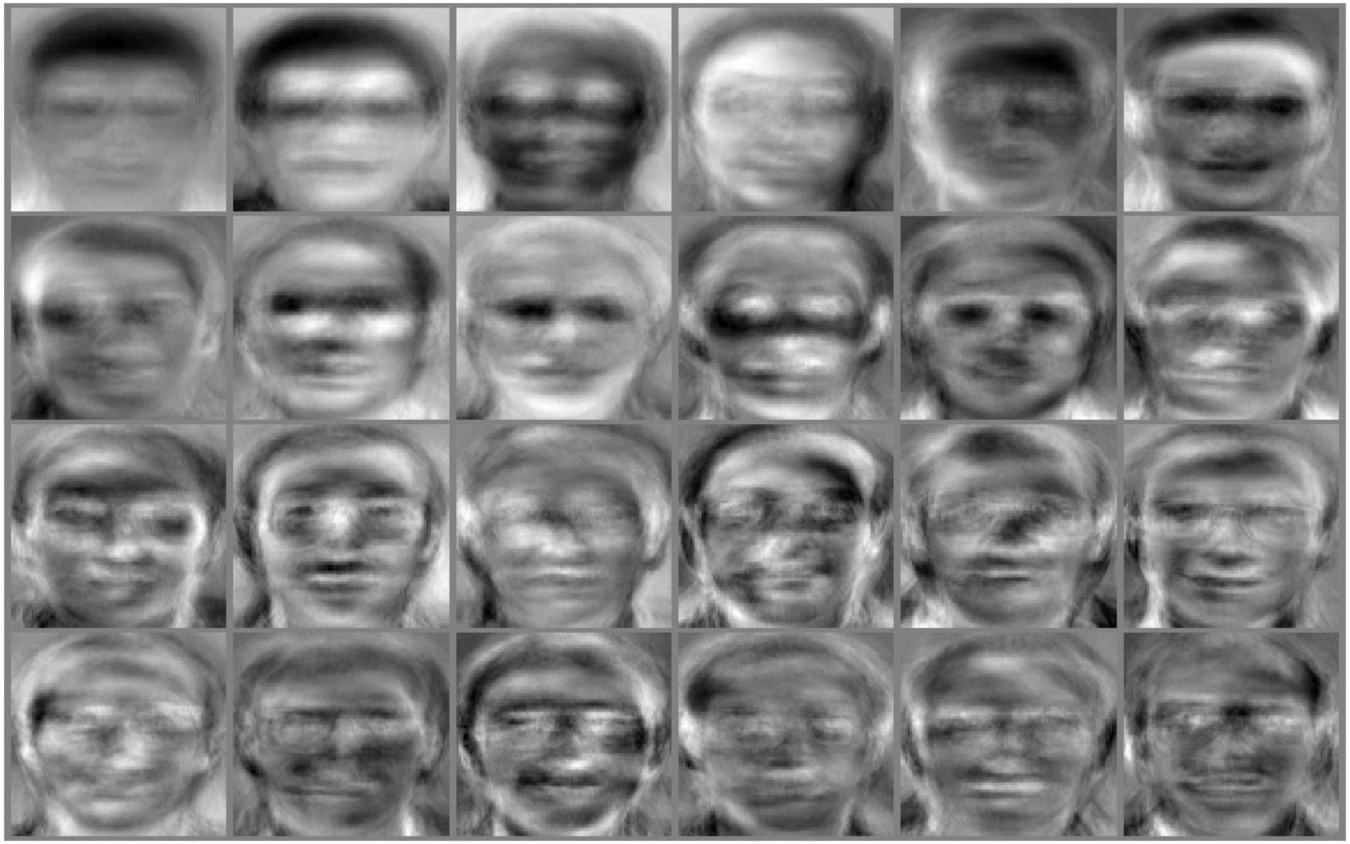
ACRONYM (Brooks 1982) matching with generalized cylinders + search!:




(a) (b) (c) (d)

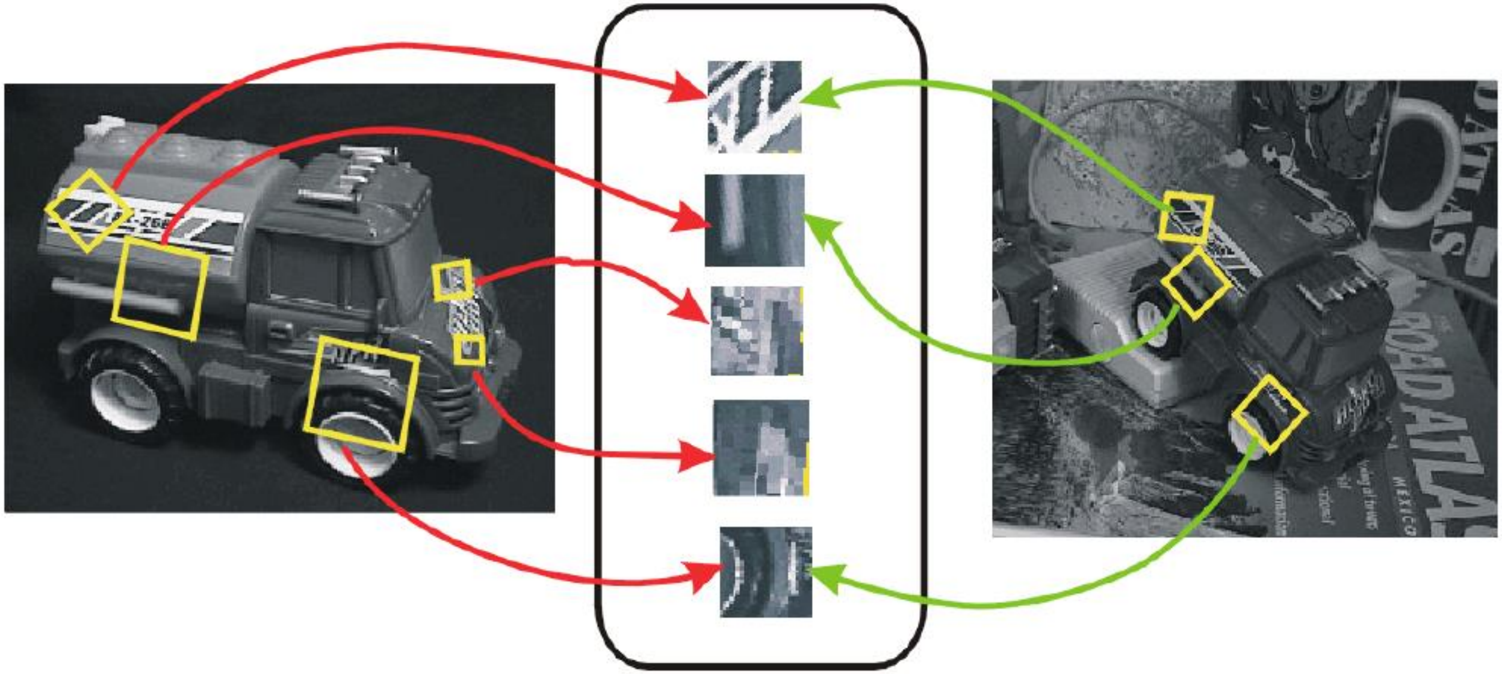
# Computer Vision - Eigenfaces

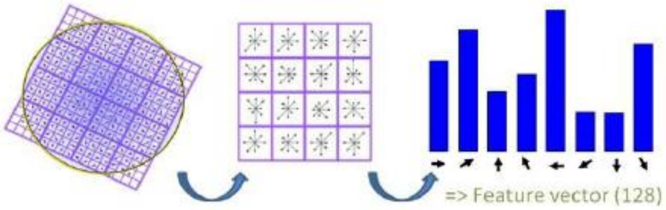
Turk and Pentland 1991:



A data-driven approach?

# Computer Vision – Invariant Features

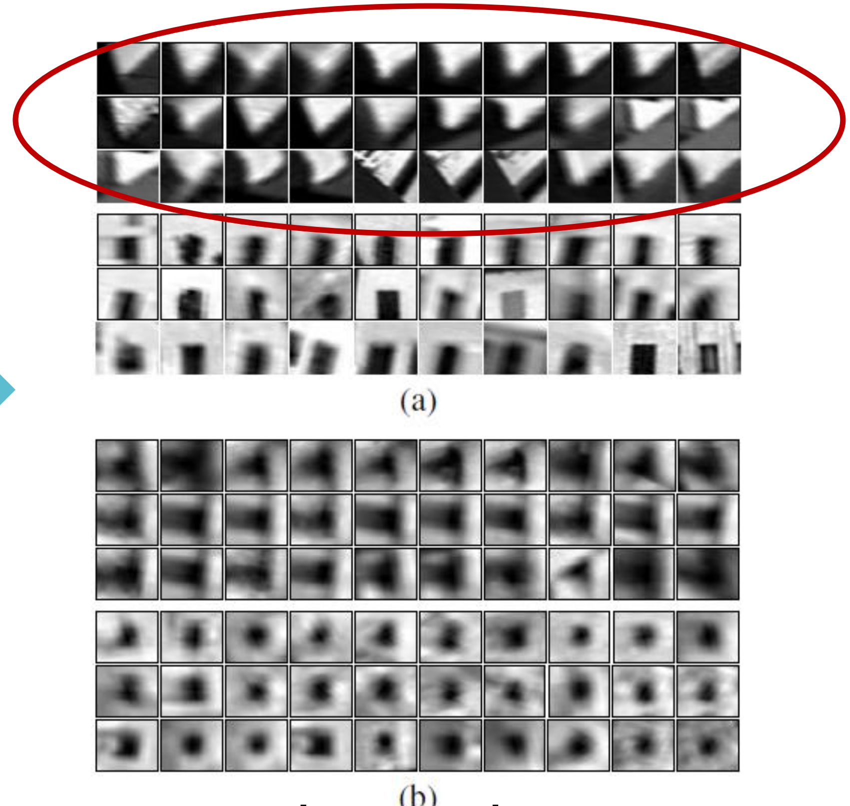





“SIFT” & Object Recognition, David Lowe, 1999

# Computer Vision – Bag-of-words

Sivic and Zisserman (2003) “Video Google”

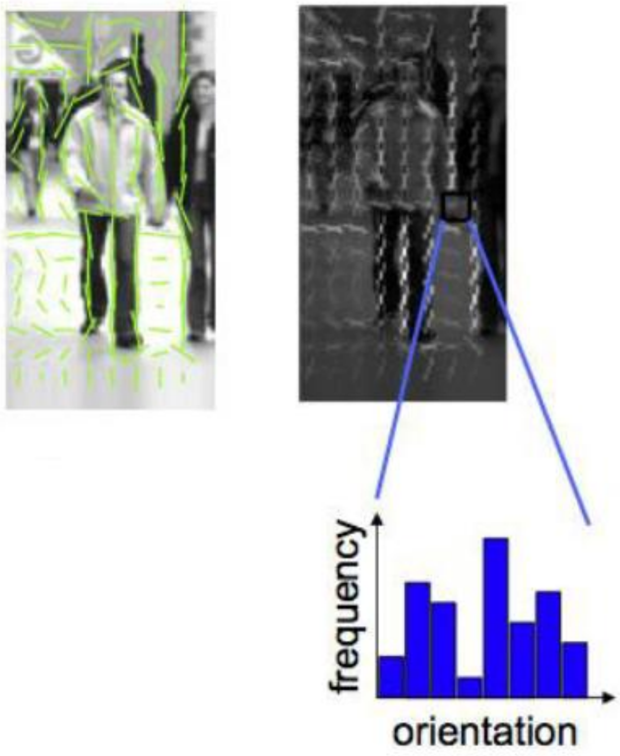


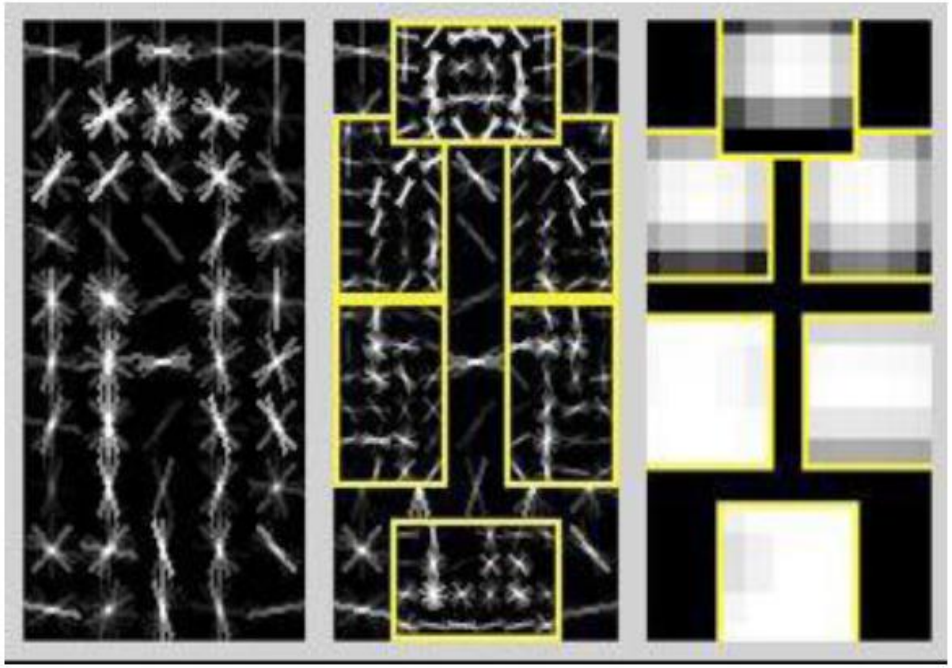

SIFT → SA and MS regions

(Shape Adapted, Maximally Stable)

code words

# Computer Vision - Shape matching



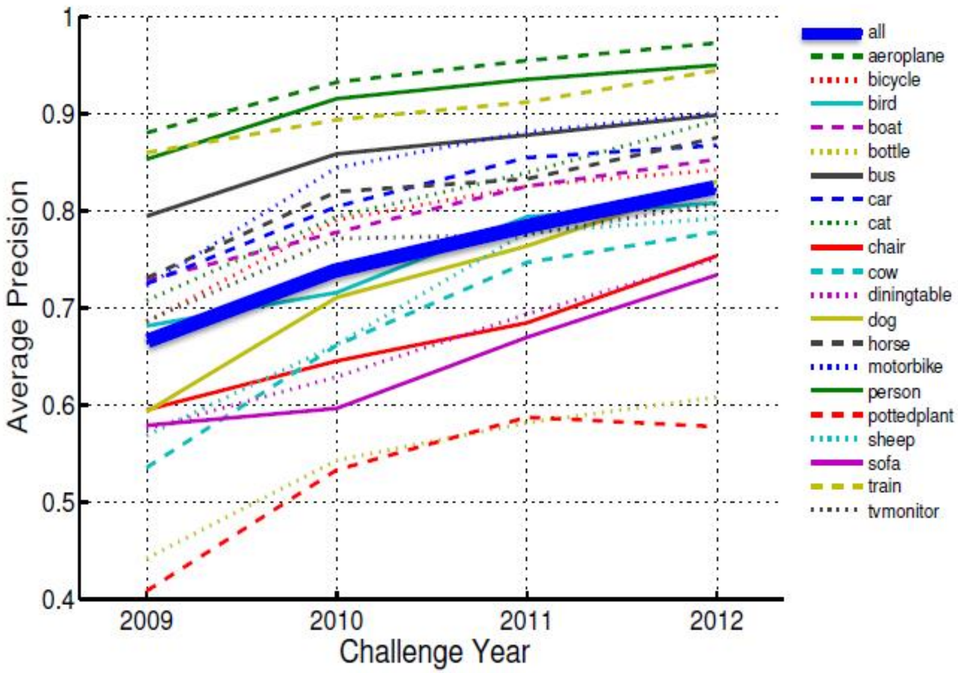
Histogram of Gradients (HoG)
Dalal & Triggs, 2005

Deformable Part Model
Felzenswalb, McAllester, Ramanan, 2009

# Computer Vision – Challenge Datasets

## PASCAL Visual Object Challenge (20 object categories)

[Everingham et al. 2006-2012]



# IMAGENET

www.image-net.org

## **22K** categories and **14M** images

- Animals
  - Bird
  - Fish
  - Mammal
  - Invertebrate
- Plants
  - Tree
  - Flower
- Food
- Materials
- Structures
- Artifact
  - Tools
  - Appliances
  - Structures
- Person
- Scenes
  - Indoor
  - Geological Formations
- Sport Activities

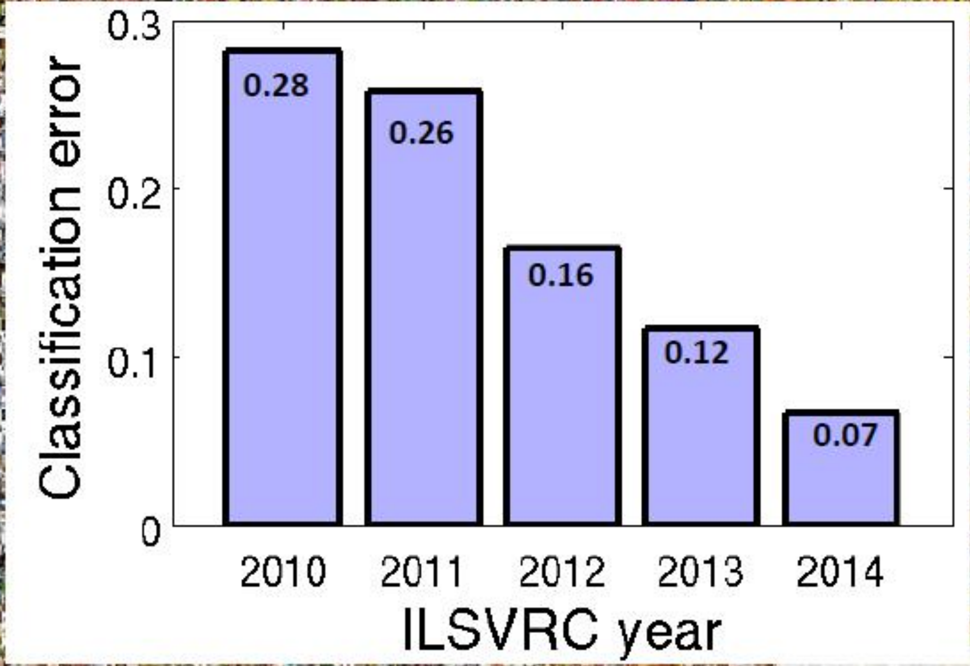
Deng, Dong, Socher, Li, Li, & Fei-Fei, 2009

# IMAGENET Large Scale Visual Recognition Challenge

The Image Classification Challenge:

1,000 object classes

1,431,167 images

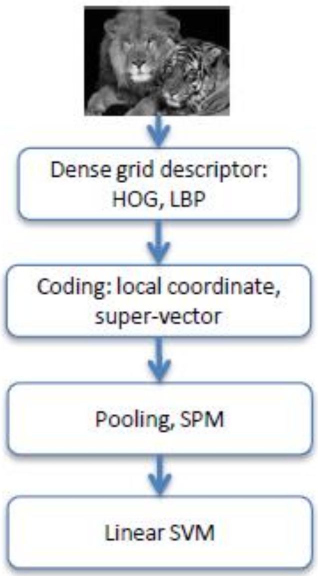



Russakovsky et al. arXiv, 2014

# IMAGENET Large Scale Visual Recognition Challenge

## Year 2010

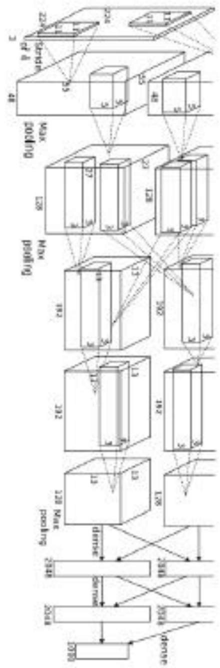
NEC-UIUC




[Lin CVPR 2011]

## Year 2012

SuperVision



[Krizhevsky NIPS 2012]

## Year 2014

GoogLeNet

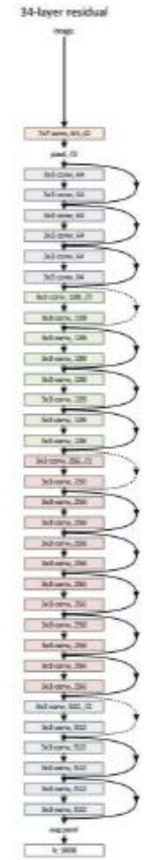

[Szegedy arxiv 2014]

VGG


[Simonyan arxiv 2014]

## Year 2015

MSRA



# Image Classification: a core task in Computer Vision

(assume given set of discrete labels)
{dog, cat, truck, plane, ...}

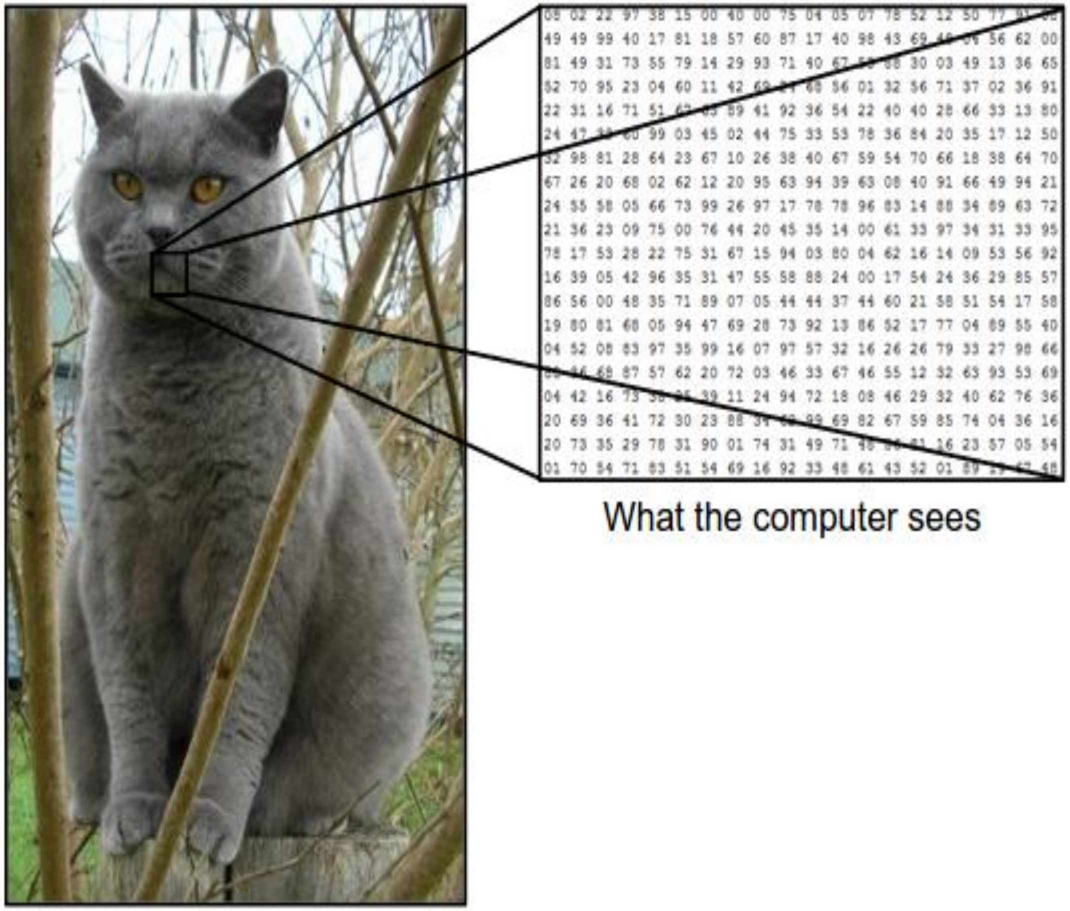
→ cat

# The problem:*semantic gap*

Images are represented as 3D arrays of numbers, with integers between [0, 255].

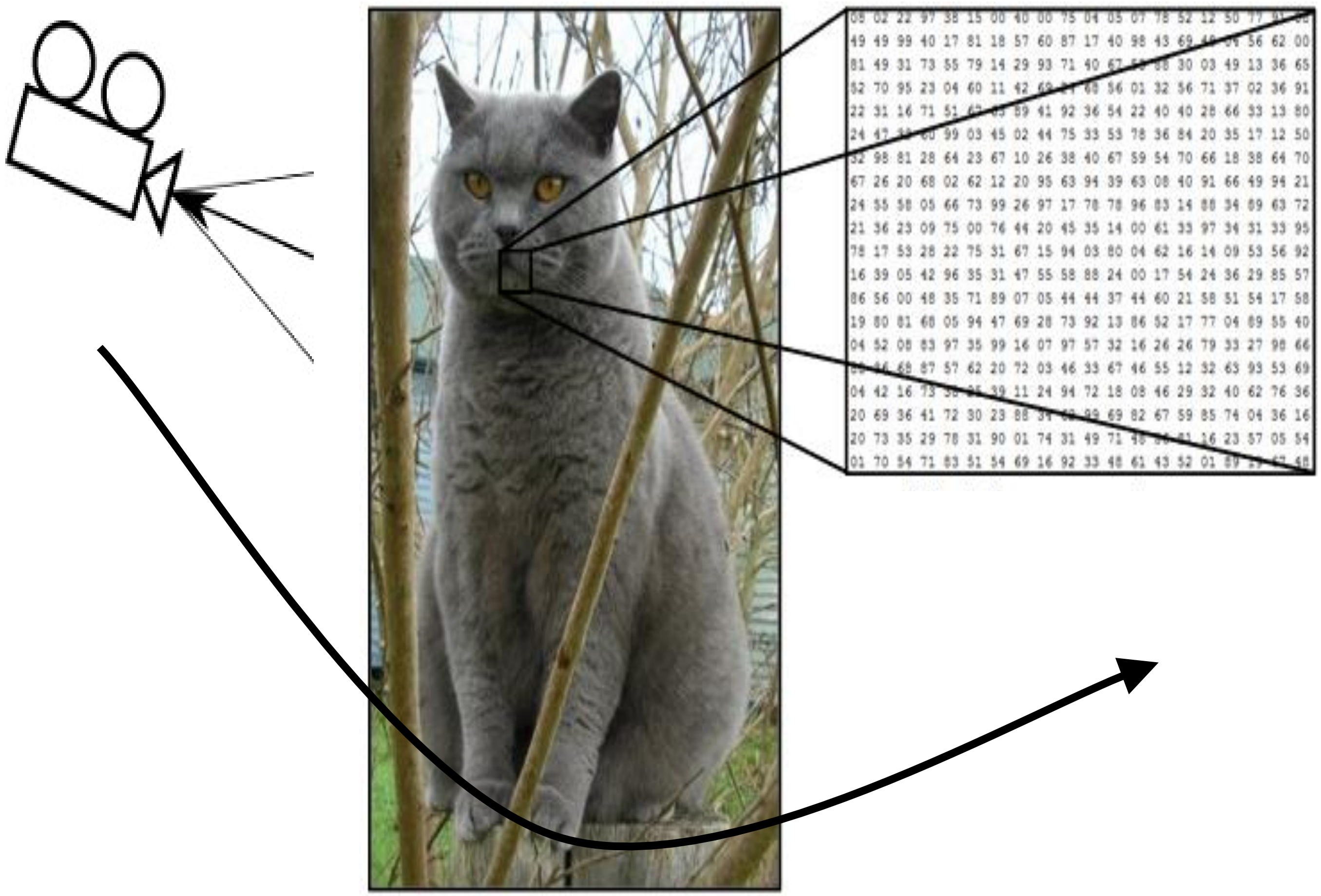
E.g.
300 x 100 x 3

(3 for 3 color channels RGB)



What the computer sees

# Challenges: Viewpoint Variation



# Challenges: Illumination

# Challenges: Deformation

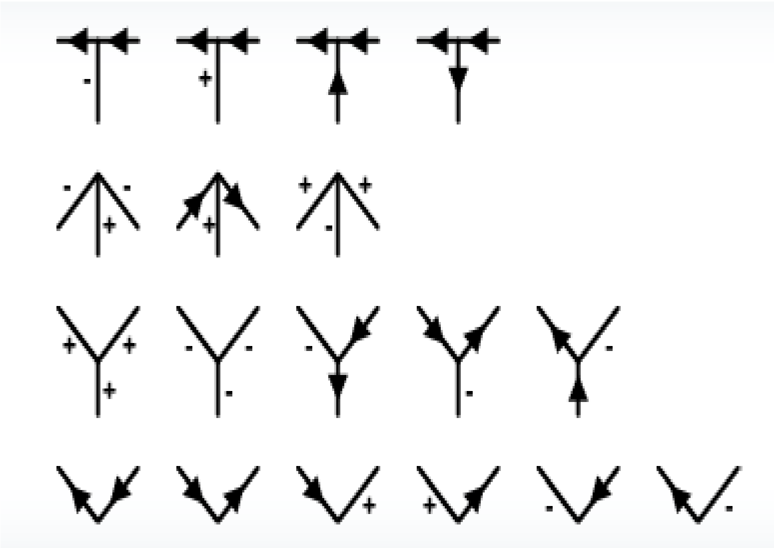
# Challenges: Occlusion

# Challenges: Background clutter

# Challenges: Intraclass variation

# Classifying using constraints? ?

???



# Data-driven approach:

1. Collect a dataset of images and labels
2. Use Machine Learning to train an image classifier
3. Evaluate the classifier on a withheld set of test images

```
def train(train_images, train_labels):
  # build a model for images -> labels...
  return model

def predict(model, test_images):
  # predict test_labels using the model...
  return test_labels
```

**Example training set**

# First classifier: Nearest Neighbor Classifier

```
def train(train_images, train_labels):
  # build a model for images -> labels...
  return model

def predict(model, test_images):
  # predict test_labels using the model...
  return test_labels
```

Remember all training images and their labels

Predict the label of the most similar training image

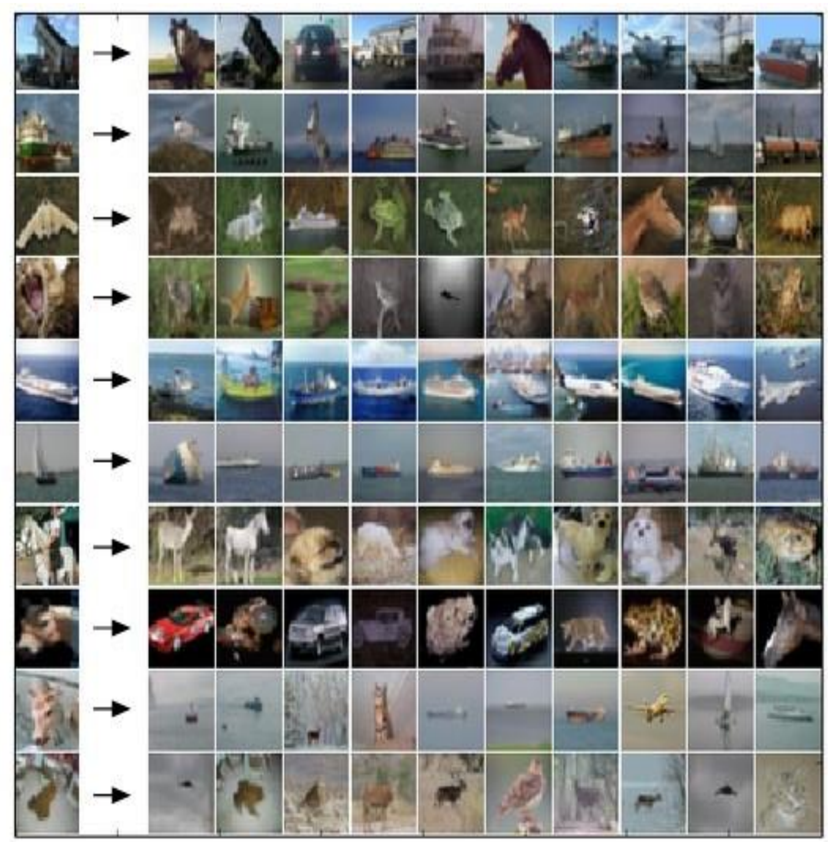
# Example dataset: CIFAR-10

**10** labels
**50,000** training images, each image is tiny: 32x32
**10,000** test images.

# Example dataset: CIFAR-10

**10** labels
**50,000** training images
**10,000** test images.

For every test image (first column), examples of nearest neighbors in rows

# How do we compare the images?
# What is the **distance metric**?

**L1 distance:**

$$d_1(I_1, I_2) = \sum_p \left| I_1^p - I_2^p \right|$$

test image

| | | | |
|---|---|---|---|
| 56 | 32 | 10 | 18 |
| 90 | 23 | 128 | 133 |
| 24 | 26 | 178 | 200 |
| 2 | 0 | 255 | 220 |

\-

training image

| | | | |
|---|---|---|---|
| 10 | 20 | 24 | 17 |
| 8 | 10 | 89 | 100 |
| 12 | 16 | 178 | 170 |
| 4 | 32 | 233 | 112 |

=

pixel-wise absolute value differences

| | | | |
|---|---|---|---|
| 46 | 12 | 14 | 1 |
| 82 | 13 | 39 | 33 |
| 12 | 10 | 0 | 30 |
| 2 | 32 | 22 | 108 |

$\xrightarrow{\text{add}}$ 456

The choice of distance is a **hyperparameter**
common choices:

L1 (Manhattan) distance

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

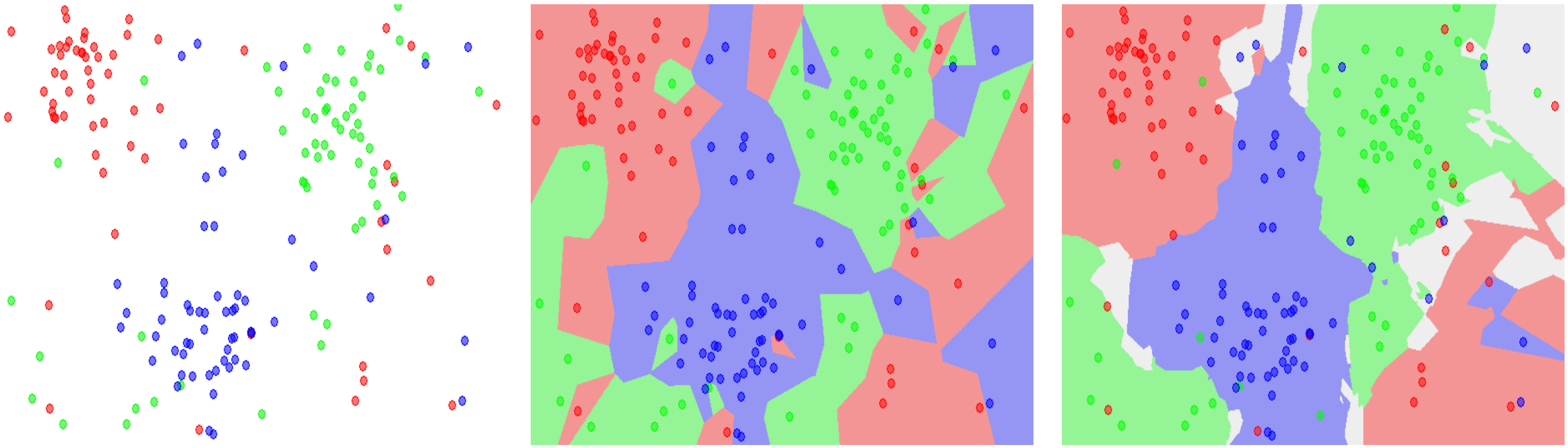
L2 (Euclidean) distance

$$d_2(I_1, I_2) = \sqrt{\sum_p \left(I_1^p - I_2^p\right)^2}$$

# k-Nearest Neighbor

## find the k nearest images, have them vote on the label

the data | NN classifier | 5-NN classifier



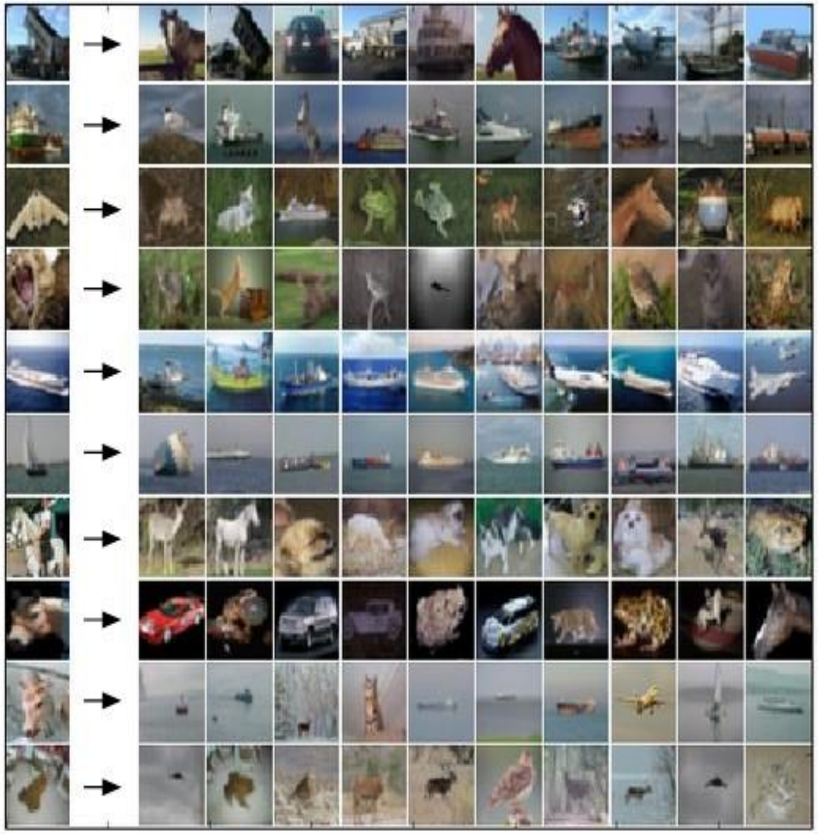
http://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

Slides based on cs231n by Fei-Fei Li & Andrej Karpathy & Justin Johnson

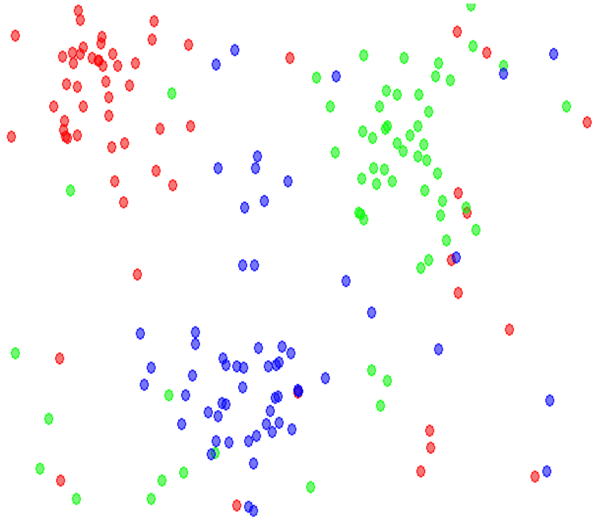
# Example dataset: CIFAR-10

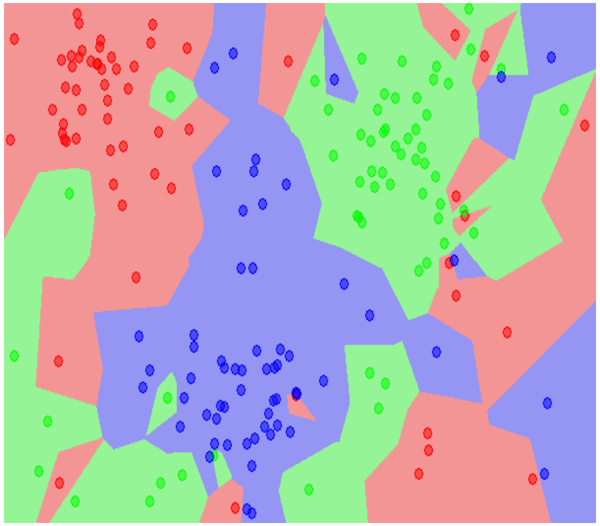
**10** labels
**50,000** training images
**10,000** test images.

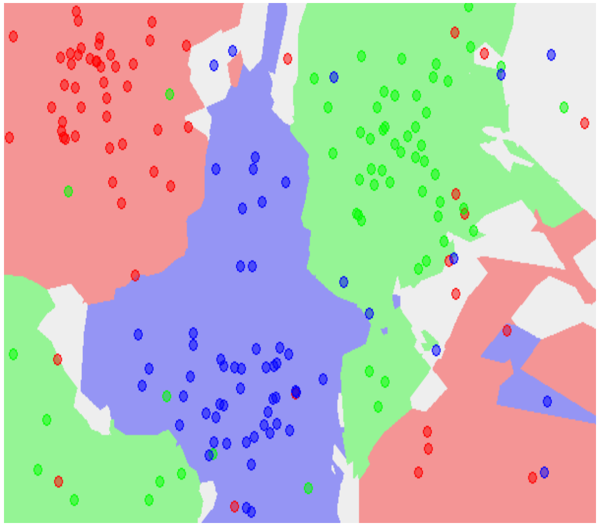

For every test image (first column), examples of nearest neighbors in rows



| the data | NN classifier | 5-NN classifier |
| --- | --- | --- |

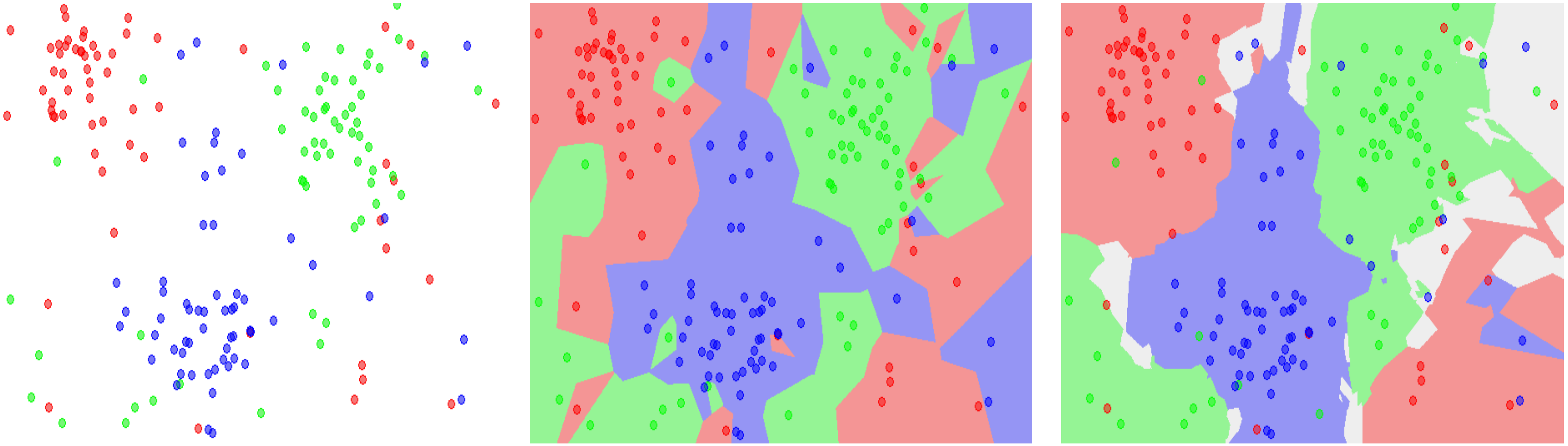






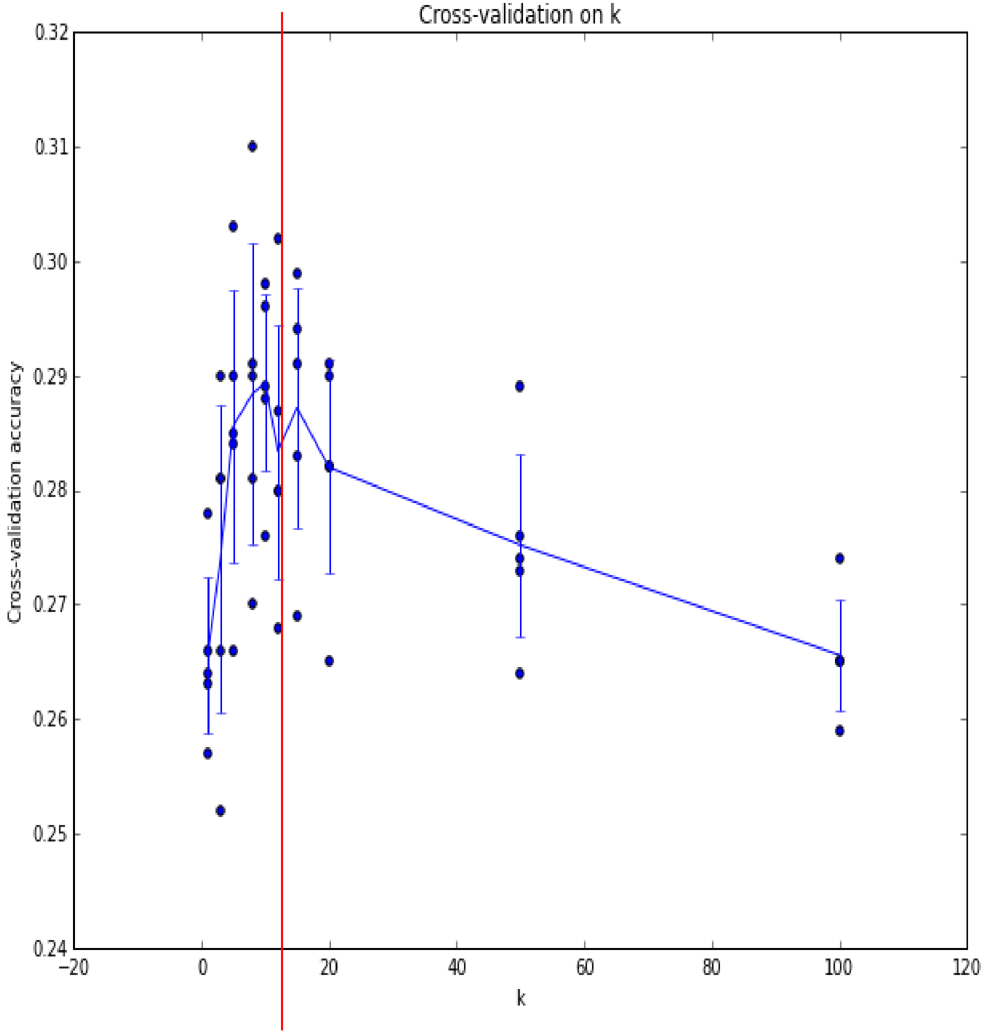
Q: what is the accuracy of the nearest neighbor classifier on the training data, when using the Euclidean distance?

the data | NN classifier | 5-NN classifier



Q2: what is the accuracy of the **k-**nearest neighbor classifier on the training data?

# Hyperparameter tuning:




Example of
5-fold cross-validation
for the value of **k.**

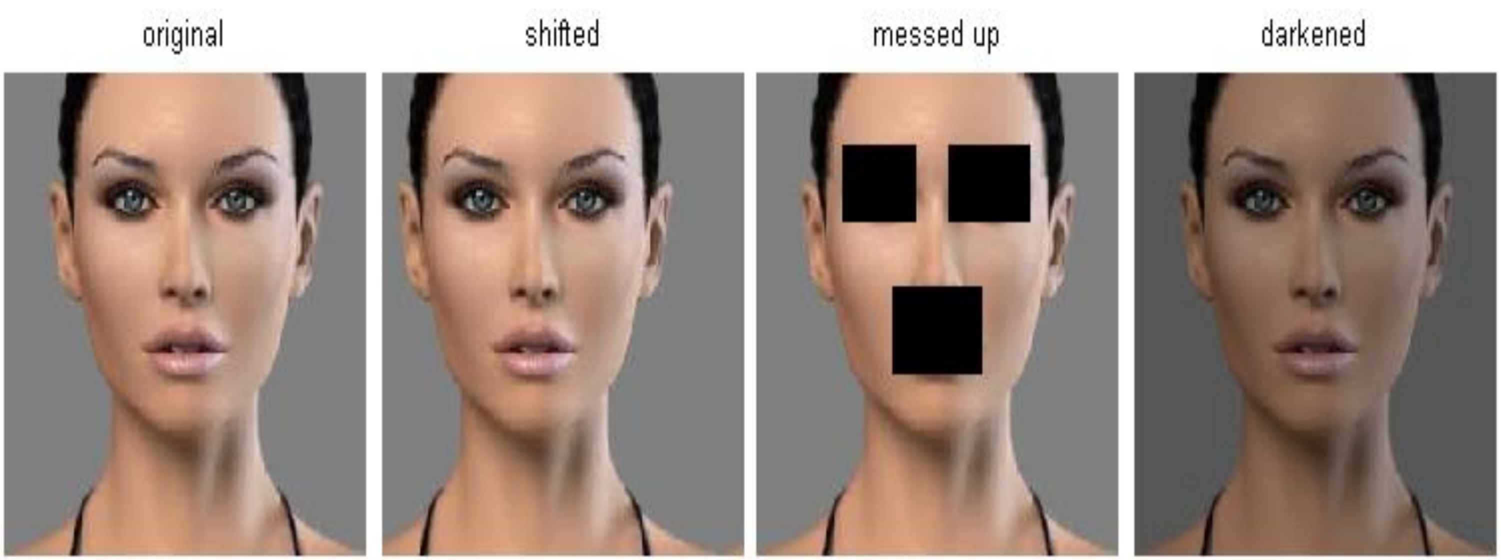
Each point: single
outcome.

The line goes
through the mean, bars
indicated standard
deviation

(Seems that k ~= 7 works best
for this data)

# k-Nearest Neighbor on images **never used.**

- terrible performance at test time
- distance metrics on level of whole images can be very unintuitive




(all 3 images have same L2 distance to the one on the left)

# Summary

- **Image Classification:** given a **Training Set** of labeled images, predict labels on **Test Set**. Common to report the **Accuracy** of predictions (fraction of correct predictions)
- We introduced the **k-Nearest Neighbor Classifier**, which predicts labels based on nearest images in the training set
- We saw that the choice of distance and the value of k are **hyperparameters** that are tuned using a **validation set**, or through **cross-validation** if the size of the data is small.
- Once the best set of hyperparameters is chosen, the classifier is evaluated once on the test set, and reported as the performance of kNN on that data.