UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

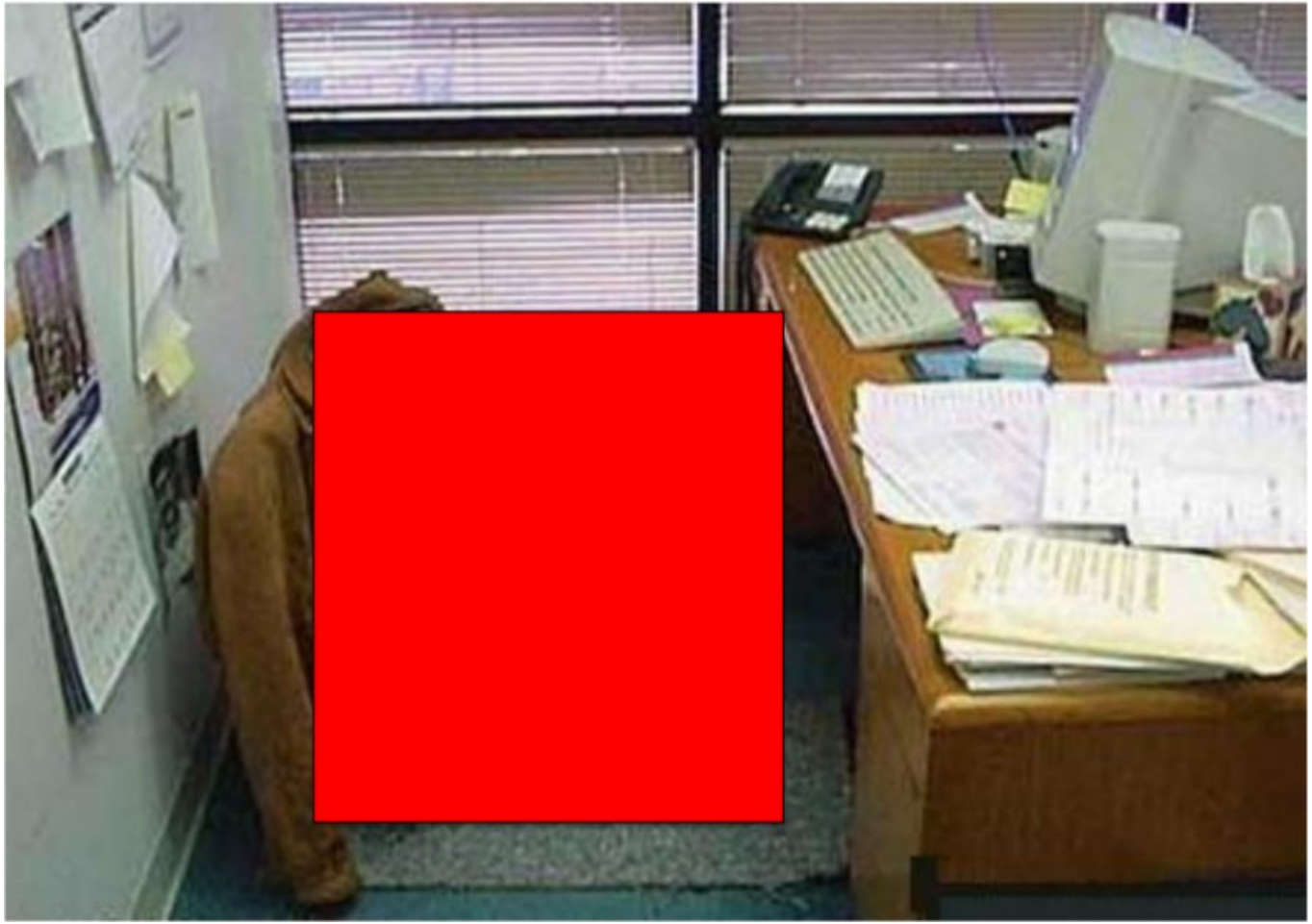
> Reinventa el mundo <

Deep Learning

Week 2 : Convolutional Neural Networks. Convolution, Non-Linear Filtering, Pooling, CE & MSE Loss, Function Approximation. History & Motivation.

TRANSFORMATEC

Example borrowed from Alyosha Efros



Example Borrowed from Antonio Torralba

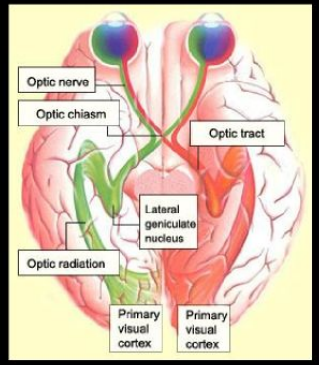
Human Vision Scientists

What we try to do

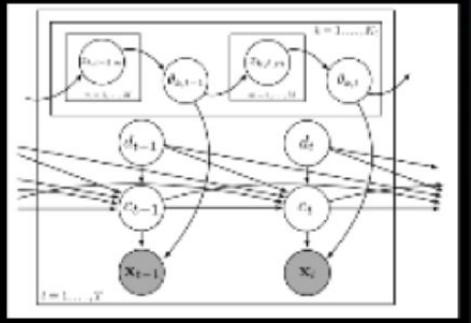
What our mothers think we do

What society thinks we do

What computer scientists think we do

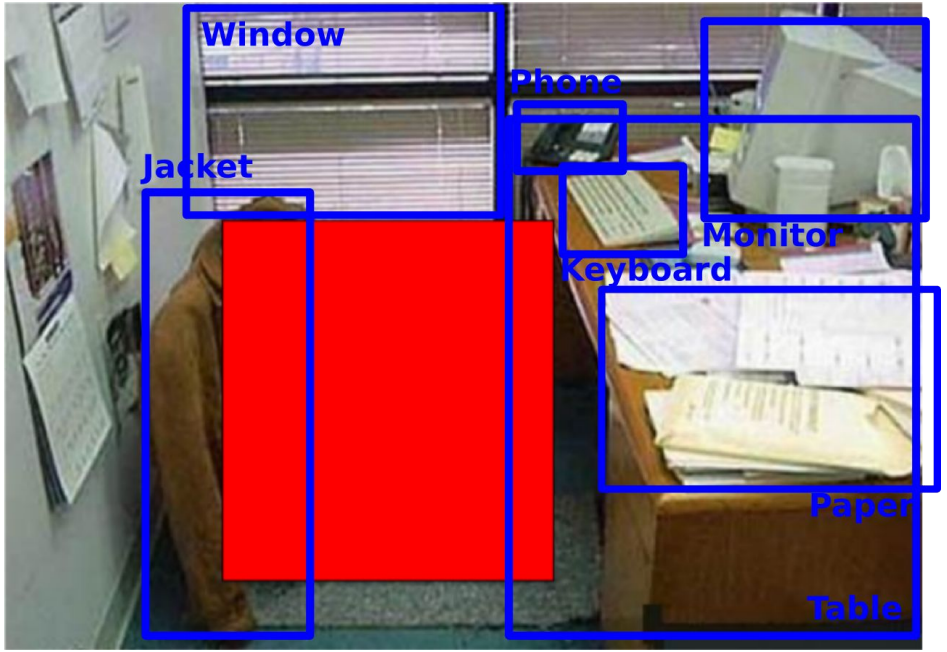


What biologists think we do

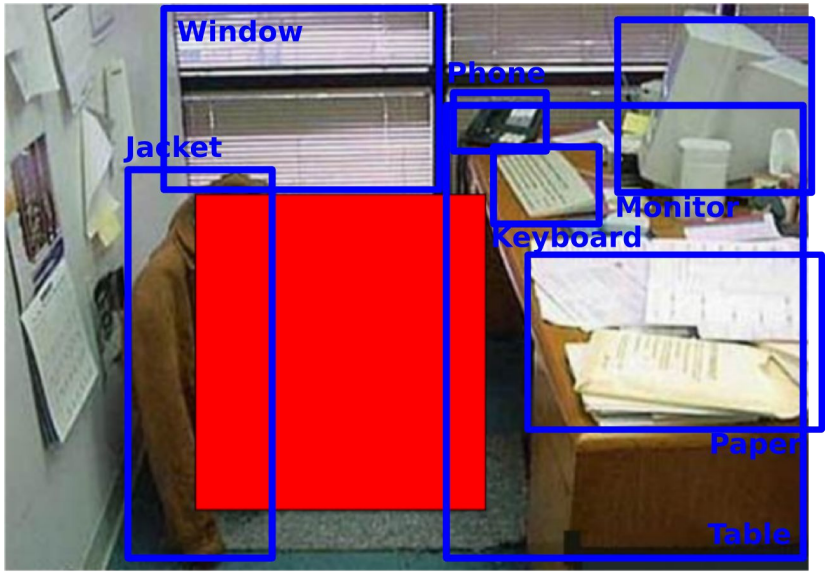


What we really do

‘The Toilet’ revisited

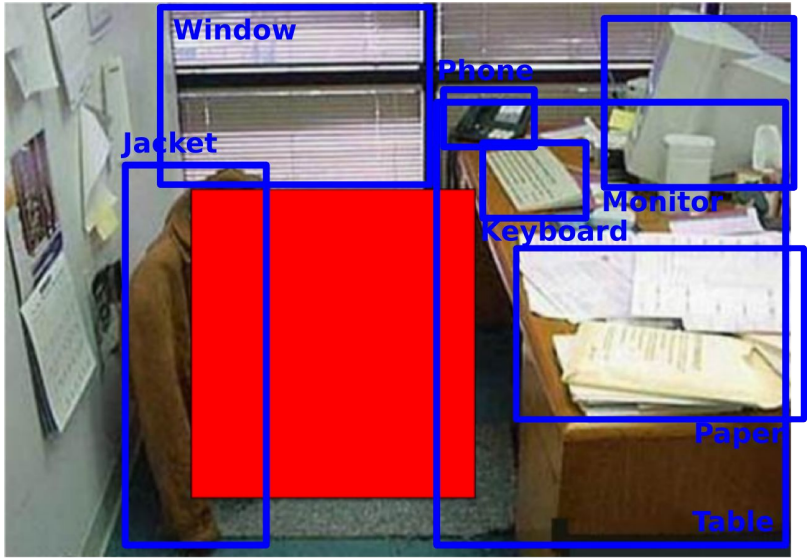


Analyzing Human Inference



Window
Monitor
Keyboard
Paper
Table
Phone
Jacket

Analyzing Human Inference

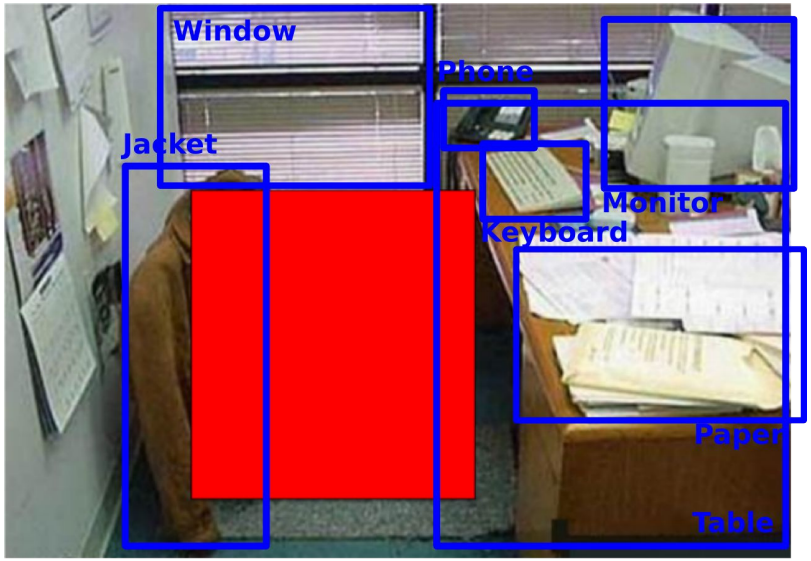


Window
Monitor
Keyboard
Paper
Table
Phone
Jacket

?

Kitchen
Soccer Stadium
Del Playa
Office
Living Room

Analyzing Human Inference



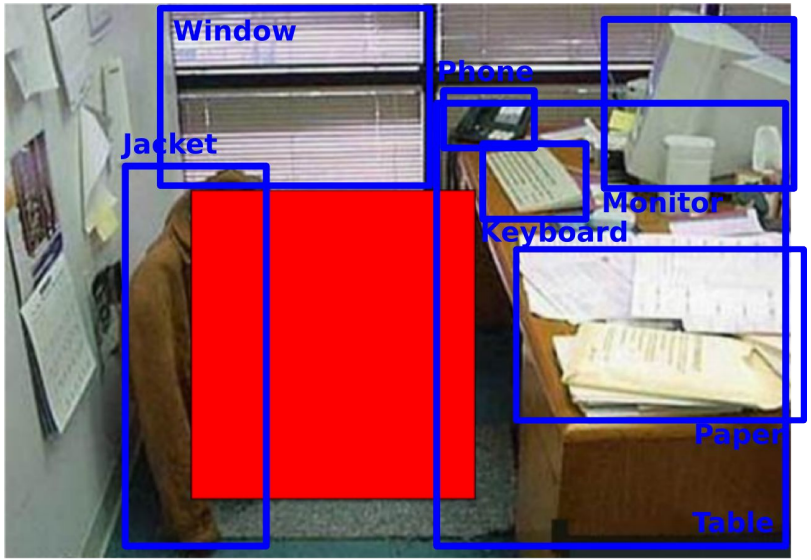
Window
Monitor
Keyboard
Paper
Table
Phone
Jacket

?

Kitchen
Soccer Stadium
Del Playa
Office
Living Room

P1 = 3/7
P2 = 2/7
P3 = 1/7
P4 = 7/7
P5 = 4/7

Analyzing Human Inference



Window
Monitor
Keyboard
Paper
Table
Phone
Jacket

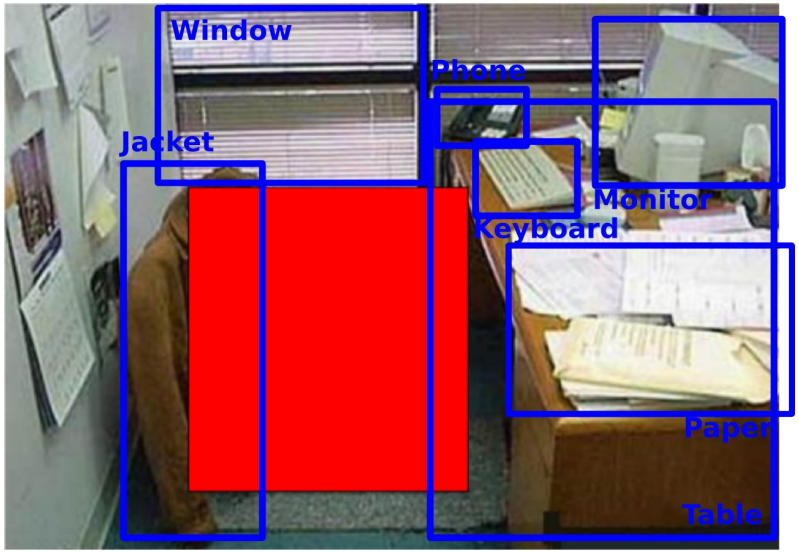
?

Kitchen
Soccer Stadium
Del Playa
Office
Living Room

P1 = 3/7
P2 = 2/7
P3 = 1/7
P4 = 7/7
P5 = 4/7

"Bag of Words"

Analyzing Human Inference



Office

Window
Monitor
Chair
Keyboard
Paper
Table
Phone
Computer
Jacket
Coffee

Window
Monitor
Keyboard
Paper
Table
Phone
Jacket

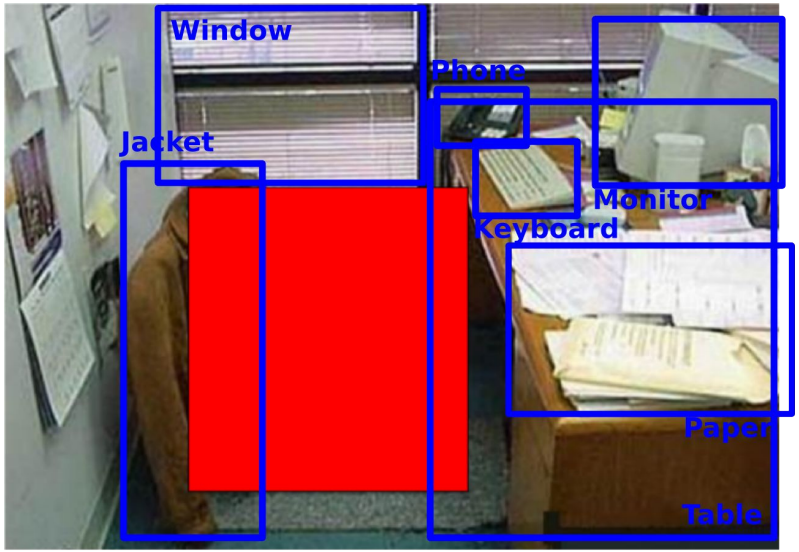
?

Kitchen
Soccer Stadium
Del Playa
Office
Living Room

P1 = 3/7
P2 = 2/7
P3 = 1/7
P4 = 7/7
P5 = 4/7

"Bag of Words"

Analyzing Human Inference



Office

~~Window~~
~~Monitor~~
Chair
~~Keyboard~~
~~Paper~~
~~Table~~
~~Phone~~
Computer
~~Jacket~~
Coffee

Window
Monitor
Keyboard
Paper
Table
Phone
Jacket

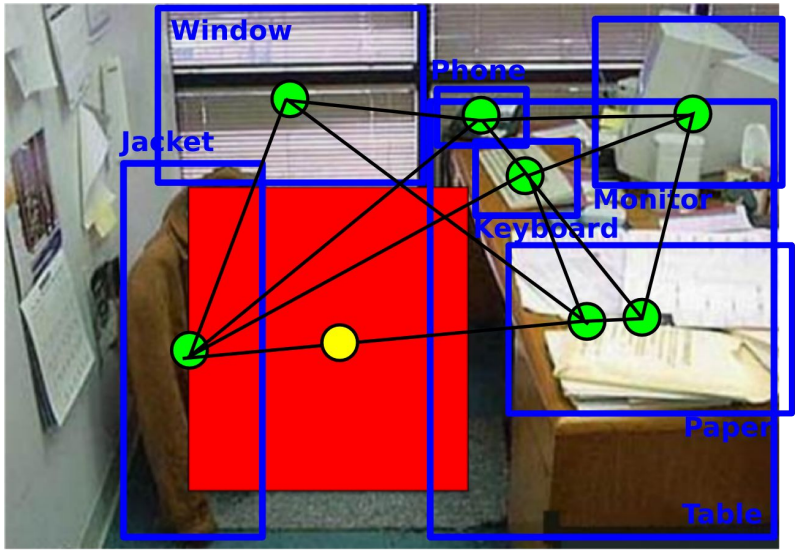
?

Kitchen
Soccer Stadium
Del Playa
Office
Living Room

P1 = 3/7
P2 = 2/7
P3 = 1/7
P4 = 7/7
P5 = 4/7

"Bag of Words"

Analyzing Human Inference



Office

~~Window~~
~~Monitor~~
Chair
~~Keyboard~~
~~Paper~~
~~Table~~
~~Phone~~
Computer
~~Jacket~~
Coffee

Window
Monitor
Keyboard
Paper
Table
Phone
Jacket

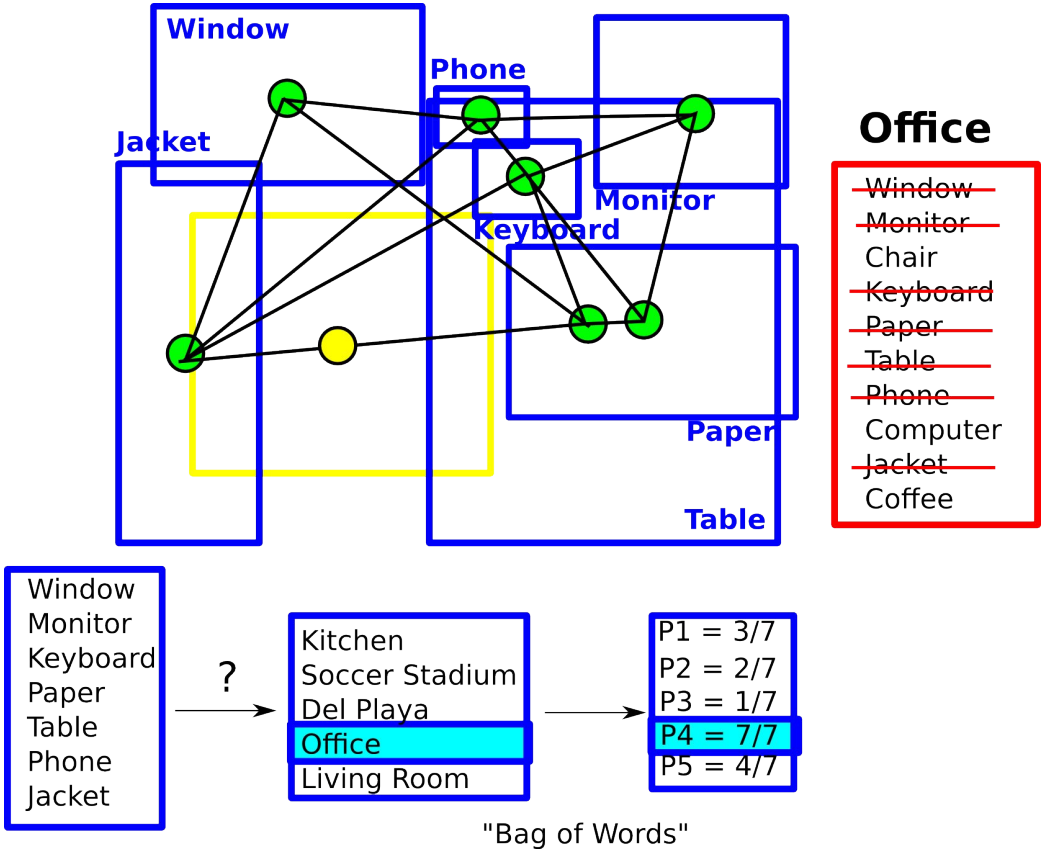
?

Kitchen
Soccer Stadium
Del Playa
Office
Living Room

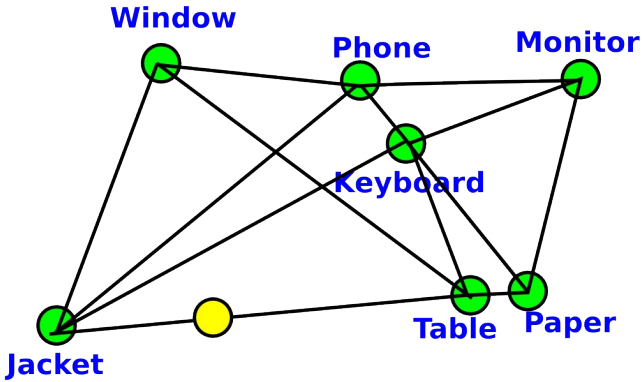
P1 = 3/7
P2 = 2/7
P3 = 1/7
P4 = 7/7
P5 = 4/7

"Bag of Words"

Analyzing Human Inference



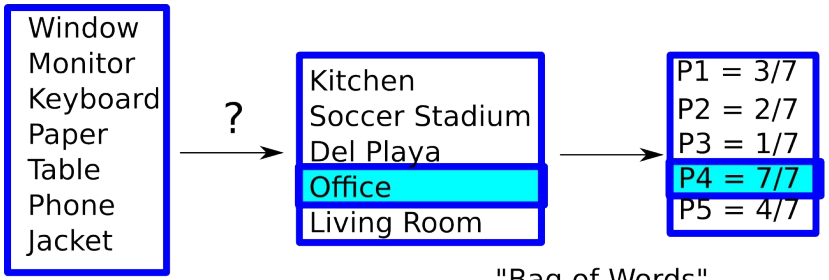
Analyzing Human Inference



What object could be between a **Jacket** and a **Table**?

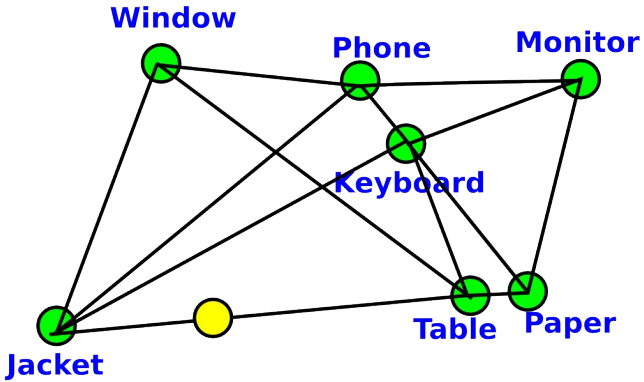
Office

~~Window~~
~~Monitor~~
Chair
~~Keyboard~~
~~Paper~~
~~Table~~
~~Phone~~
Computer
~~Jacket~~
Coffee



"Bag of Words"

Analyzing Human Inference

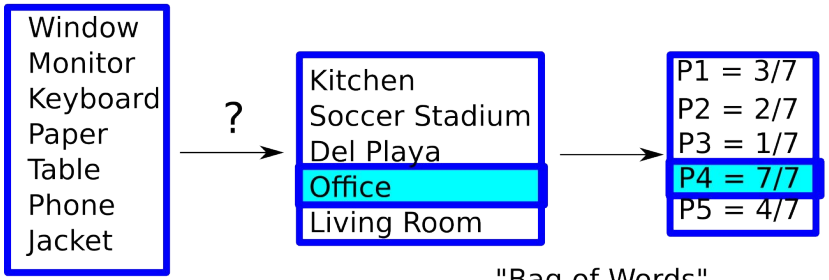


What object could be between a **Jacket** and a **Table,** inside an **Office**?

Chair

Office

~~Window~~
~~Monitor~~
Chair
~~Keyboard~~
~~Paper~~
~~Table~~
~~Phone~~
Computer
~~Jacket~~
Coffee



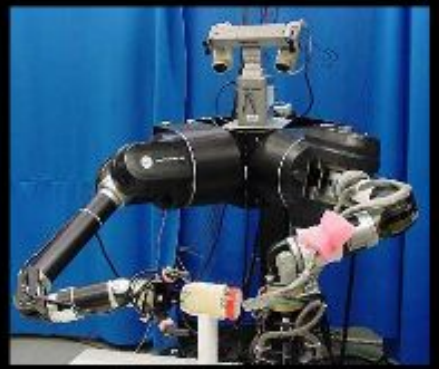
"Bag of Words"

*(Even though it was a toilet!)

Some Key points

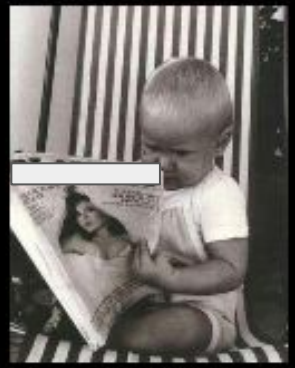
- Humans aren’t conscious (or know) how they do **inference** for object recognition.
- But we can **observe** and try to **decompose** how this pipeline is done, to implement an adapted version for a computer.

Computer Vision Scientists



What we try to do

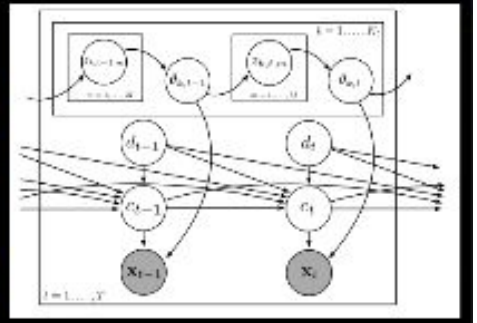
What our mothers think we do



What society thinks we do

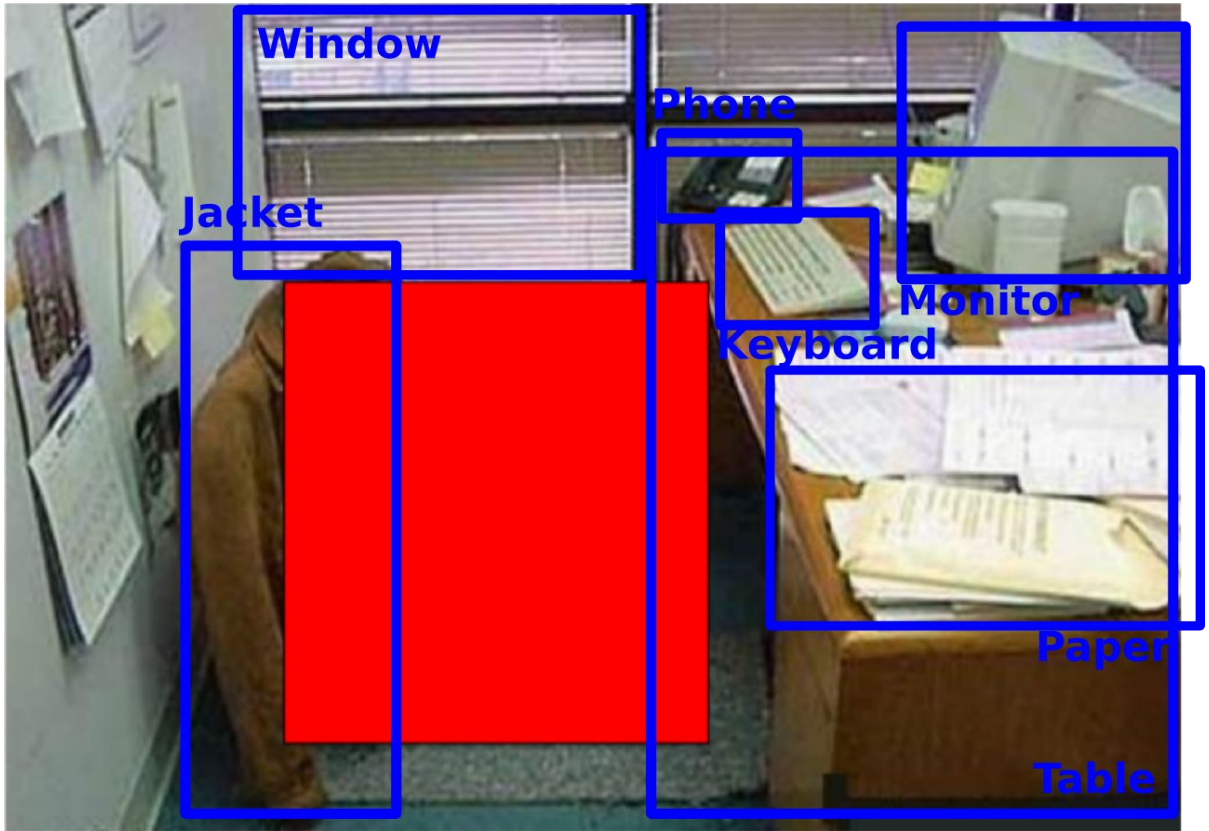
What psychologists think we do

What biologists think we do

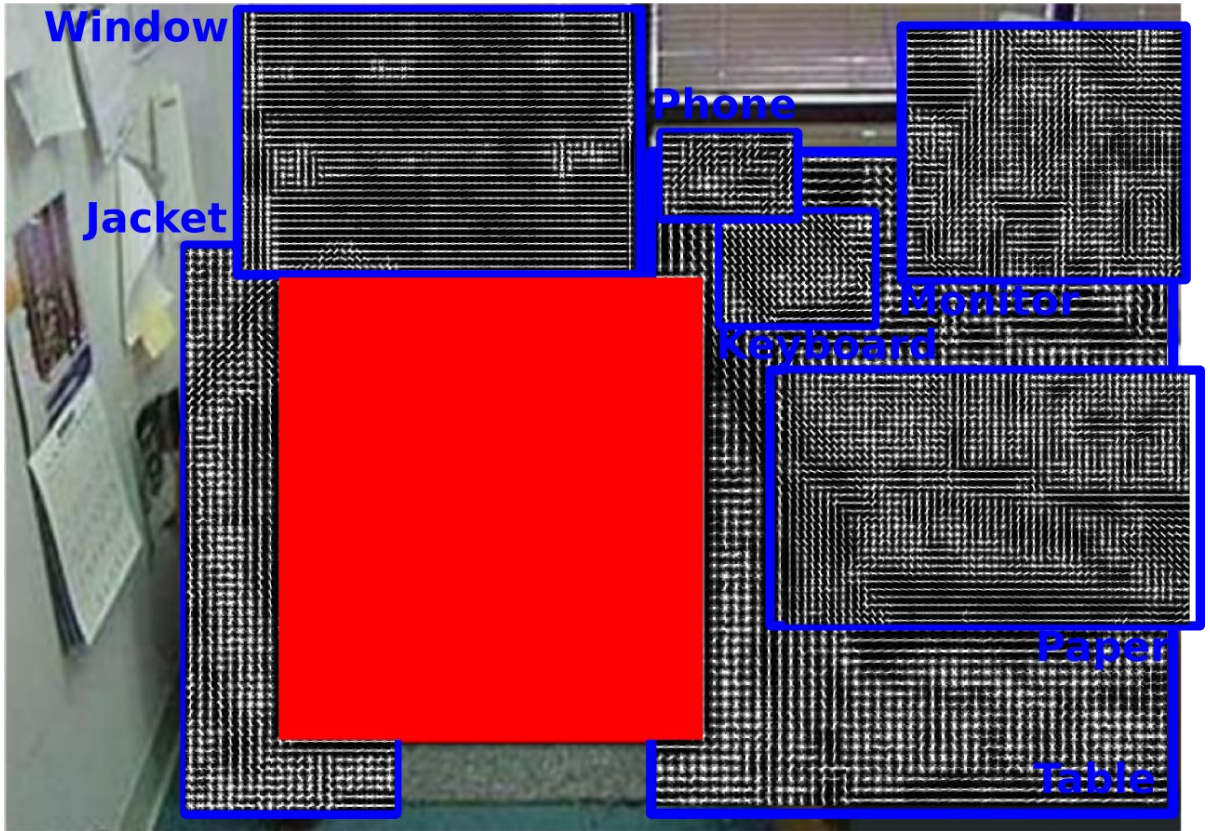


What we really do

Inference in Computers



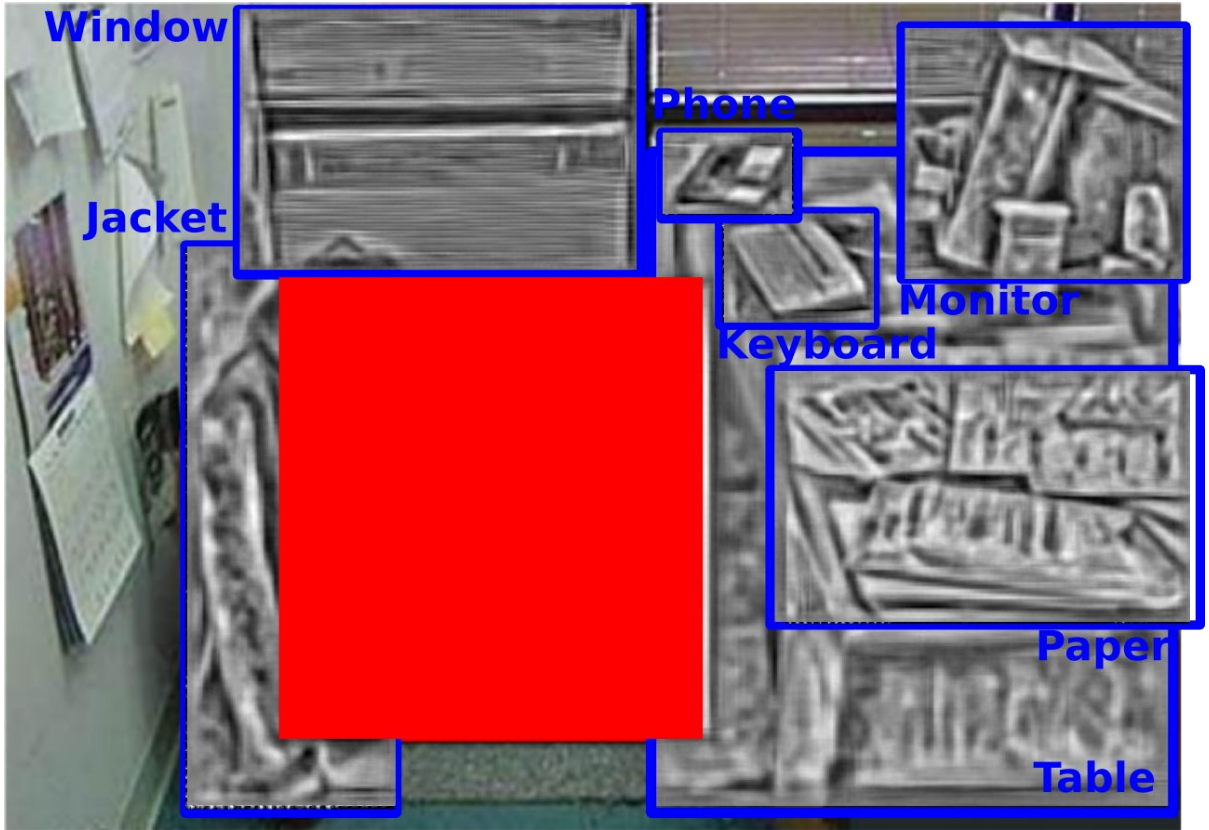
Inference in Computers



How we see what computers ‘see’

HOG (Dallel & Triggs, 2005)

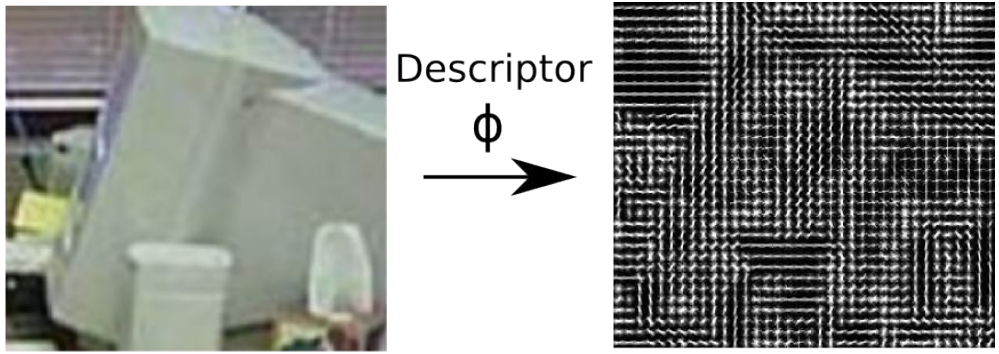
Inference in Computers



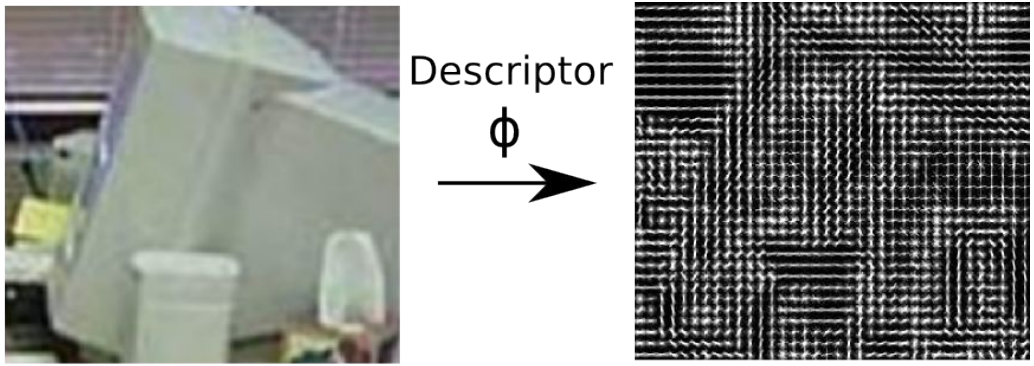
How computers ‘see’ what we see.

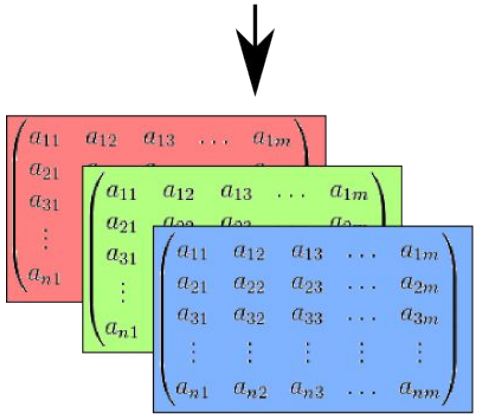
Hoggles (Vondrick, et. al, 2013)

How does the computer represent objects?

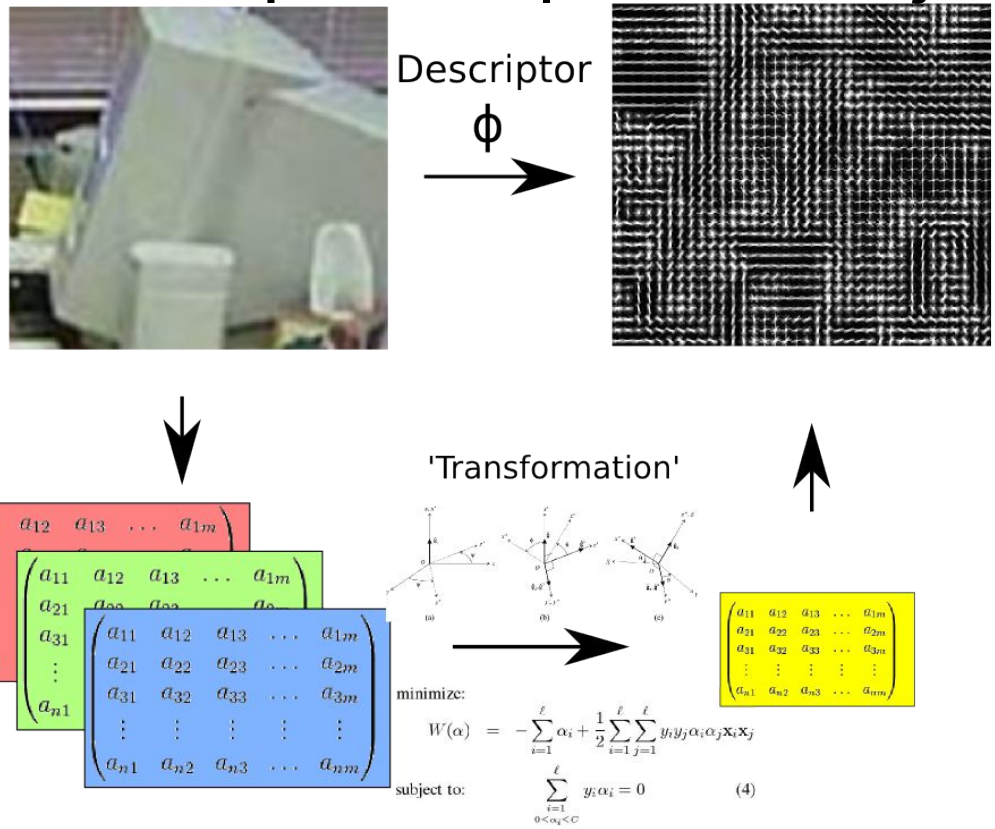


How does the computer represent objects?

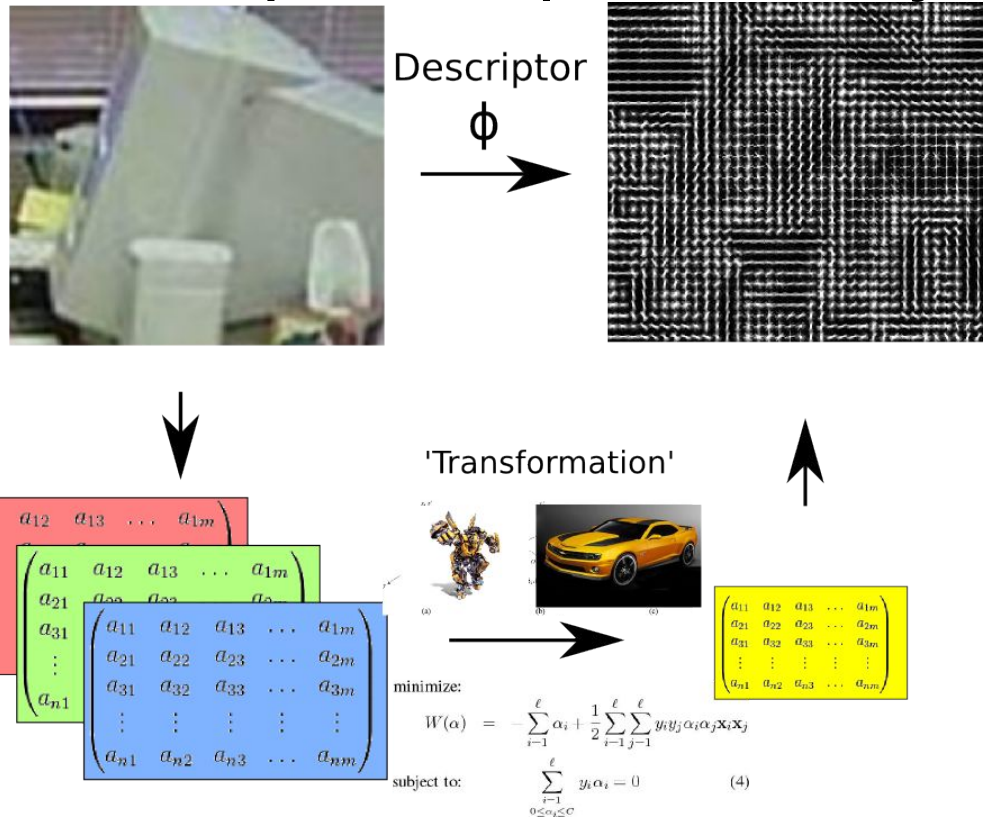




How does the computer represent objects?

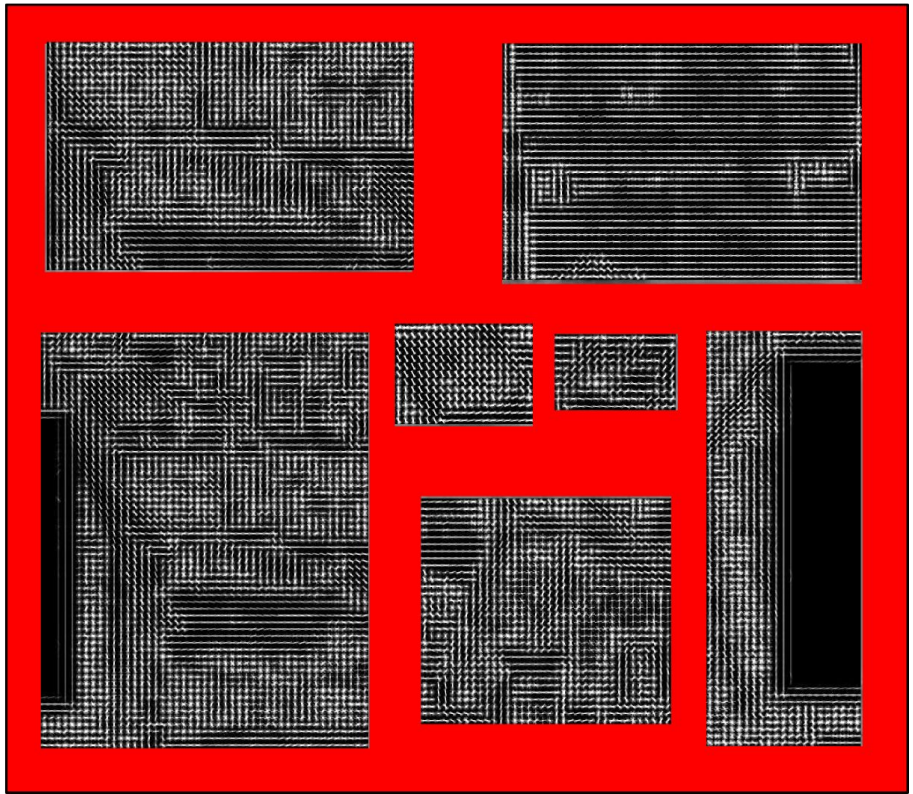


How does the computer represent objects?

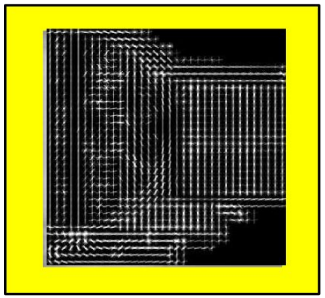


How does the computer do recognition?

Dictionary of Trained Objects:

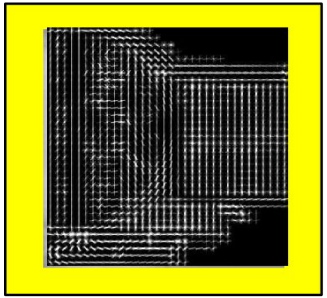


Query:



How does the computer do recognition?

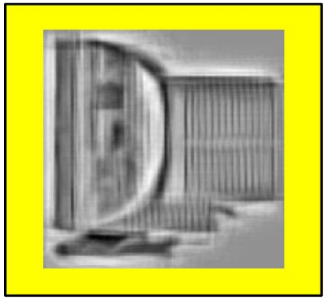
Query:



Dictionary of Trained Objects:

How does the computer do recognition?

Query:

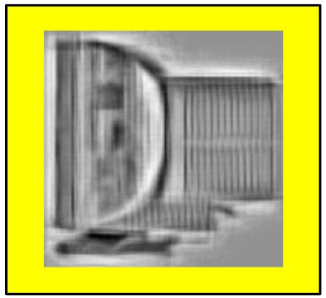


Dictionary of Trained Objects:

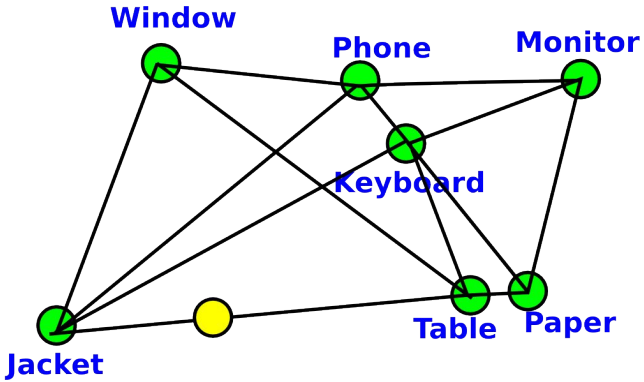
How does the computer do recognition?

Dictionary of Trained Objects:

Query:



Analyzing Computer Inference



What object could be between a **Jacket** and a **Table**?

Office

~~Window~~
~~Monitor~~
Chair
~~Keyboard~~
~~Paper~~
~~Table~~
~~Phone~~
Computer
~~Jacket~~
Coffee

$$P(Jacket|Office)$$
$$P(Window|Office)$$
$$P(Keyboard|Office)$$
$$P(Table|Office)$$
$$P(Phone|Office)$$
$$P(Monitor|Office)$$
$$P(Paper|Office)$$

Window
Monitor
Keyboard
Paper
Table
Phone
Jacket

?

Kitchen
Soccer Stadium
Del Playa
Office
Living Room

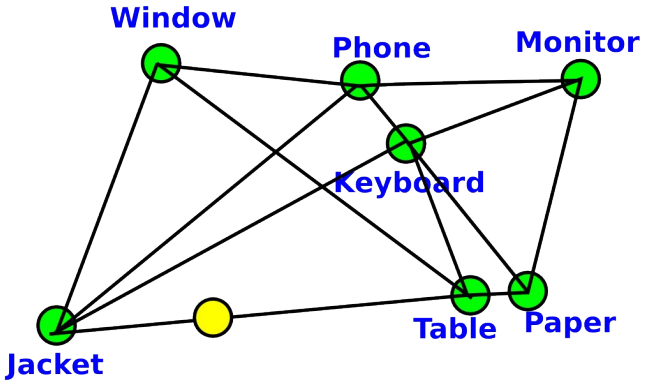
P1 = 3/7
P2 = 2/7
P3 = 1/7
P4 = 7/7
P5 = 4/7

"Bag of Words"

$$P(Chair|Office) = 0.7$$
$$P(Computer|Office) = 0.82$$
$$P(Coffee|Office) = 0.96$$

*Probabilities are non-normalized

Analyzing Computer Inference



What object could be between a **Jacket** and a **Table**?

Office

~~Window~~
~~Monitor~~
Chair
~~Keyboard~~
~~Paper~~
~~Table~~
~~Phone~~
Computer
~~Jacket~~
Coffee

$$P(Jacket|Office)$$
$$P(Window|Office)$$
$$P(Keyboard|Office)$$
$$P(Table|Office)$$
$$P(Phone|Office)$$
$$P(Monitor|Office)$$
$$P(Paper|Office)$$

Window
Monitor
Keyboard
Paper
Table
Phone
Jacket

?

Kitchen
Soccer Stadium
Del Playa
Office
Living Room

P1 = 3/7
P2 = 2/7
P3 = 1/7
P4 = 7/7
P5 = 4/7

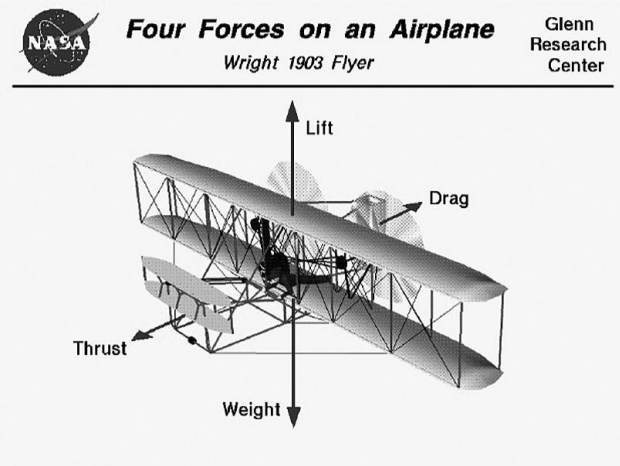
"Bag of Words"

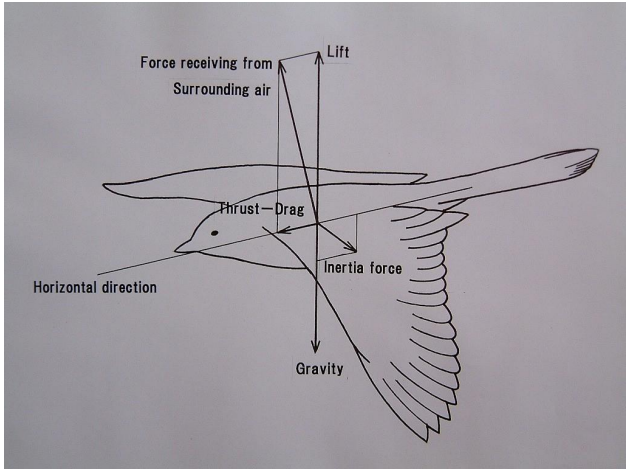
$$P(Chair|Office, Location) = 0.87$$
$$P(Computer|Office, Location) = 0.6$$
$$P(Coffee|Office, Location) = 0.3$$

*Probabilities are non-normalized

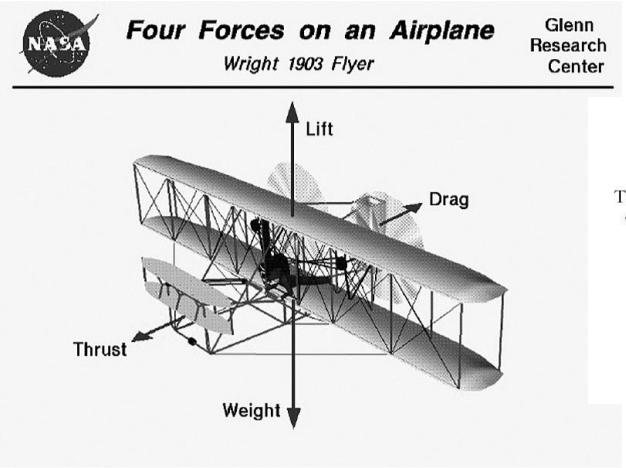
Goes back to Marr 1982 (Vision)

Goes back to Marr 1982 (Vision)

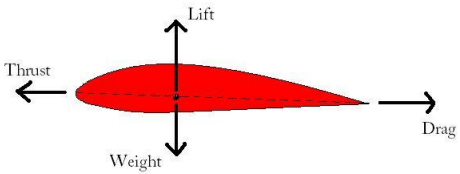


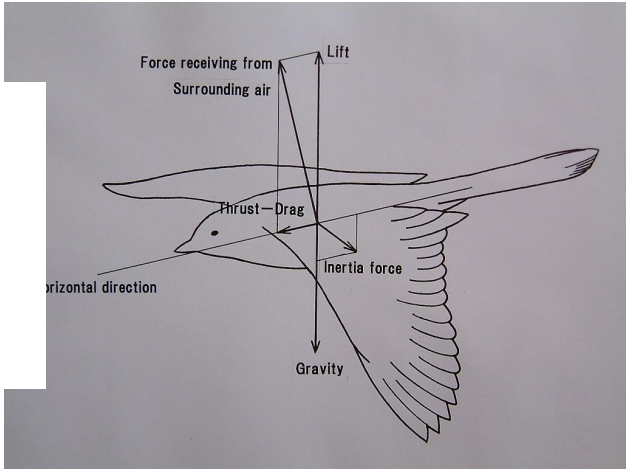


Goes back to Marr 1982 (Vision)



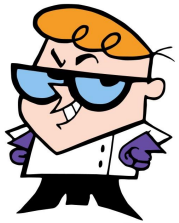
Wings!



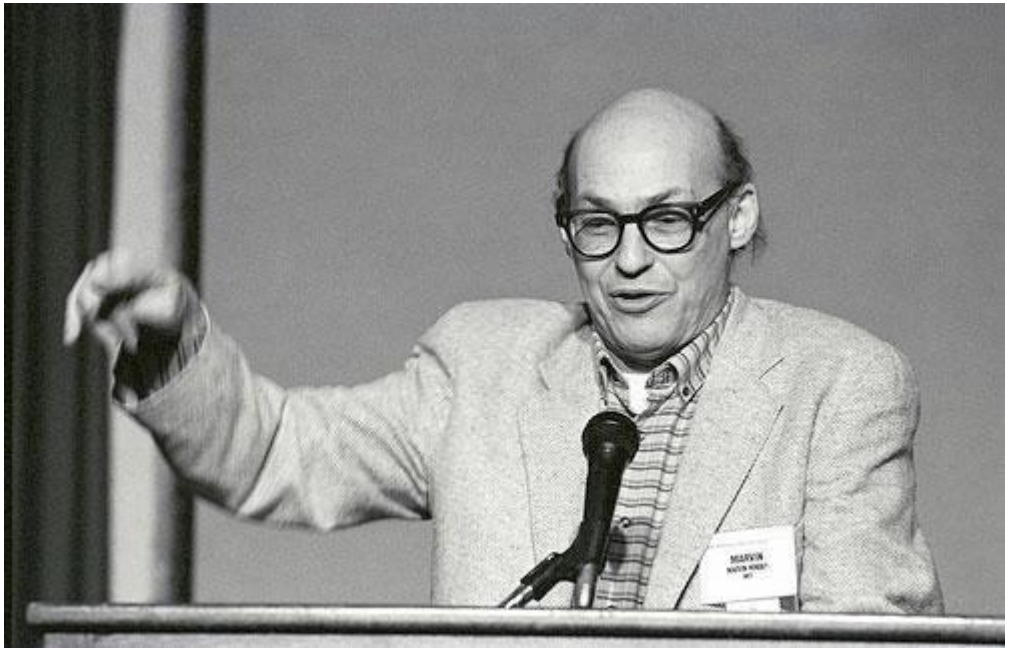


(Summer of 1966)

But how did Computer Vision start?



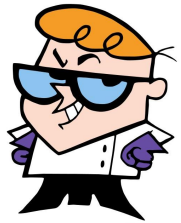
Graduate Student



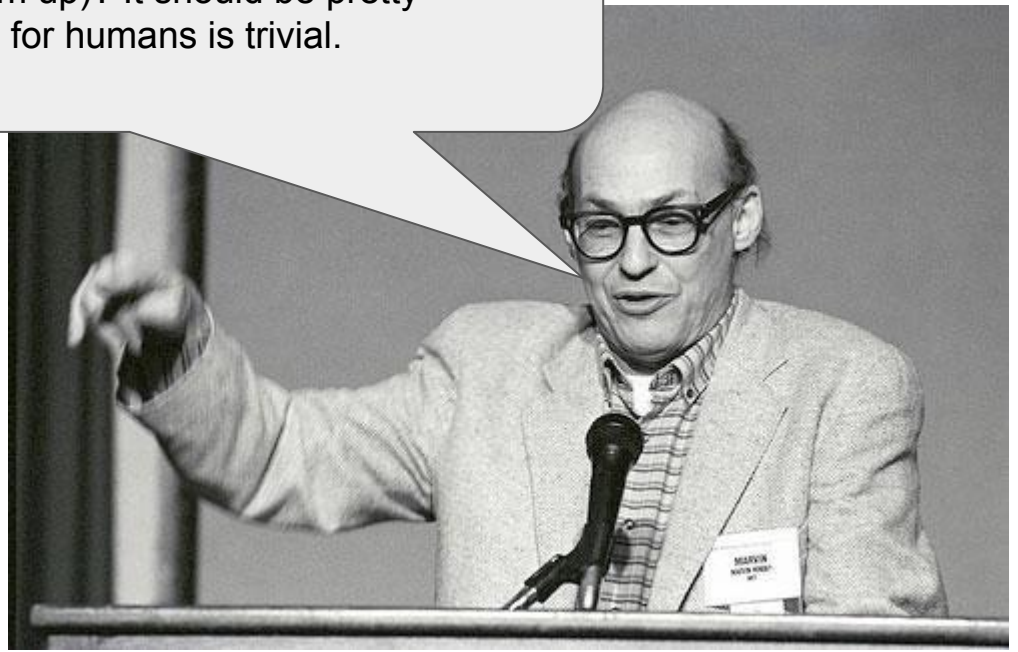
Marvin Minsky

But how did Computer Vision start?

Why don’t you build a computer that can recognize objects, as a vanilla summer project (to warm up)? It should be pretty straightforward. I mean seeing for humans is trivial.



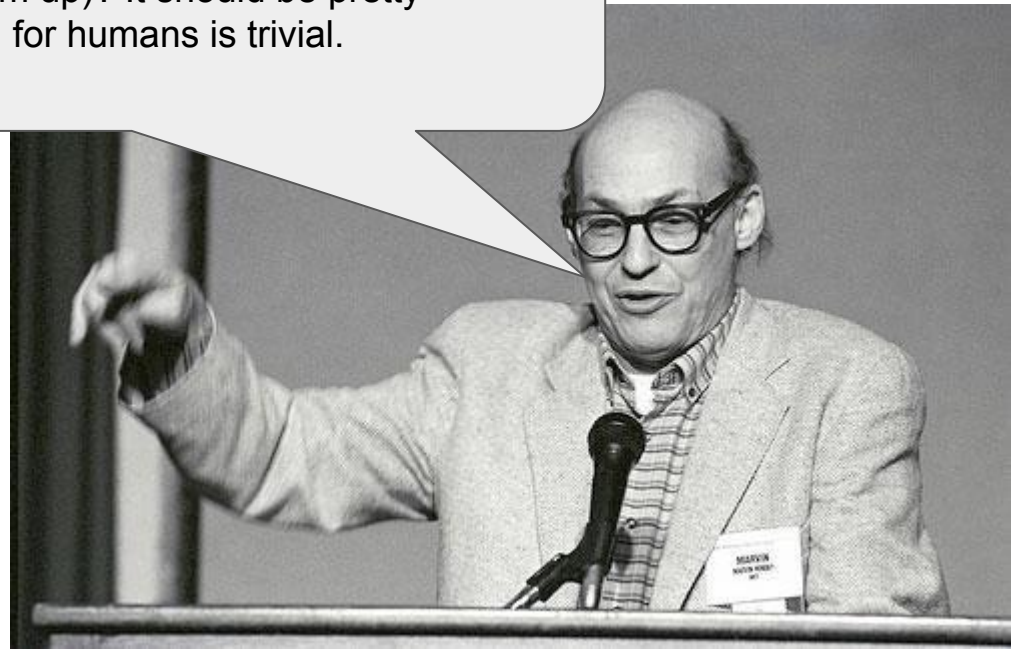
Graduate Student



Marvin Minsky

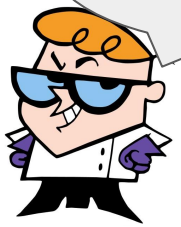
But how did Computer Vision start?

Why don’t you build a computer that can recognize objects, as a vanilla summer project (to warm up)? It should be pretty straightforward. I mean seeing for humans is trivial.



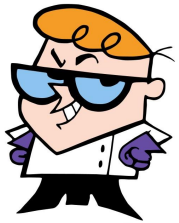
Marvin Minsky

Sounds Good!

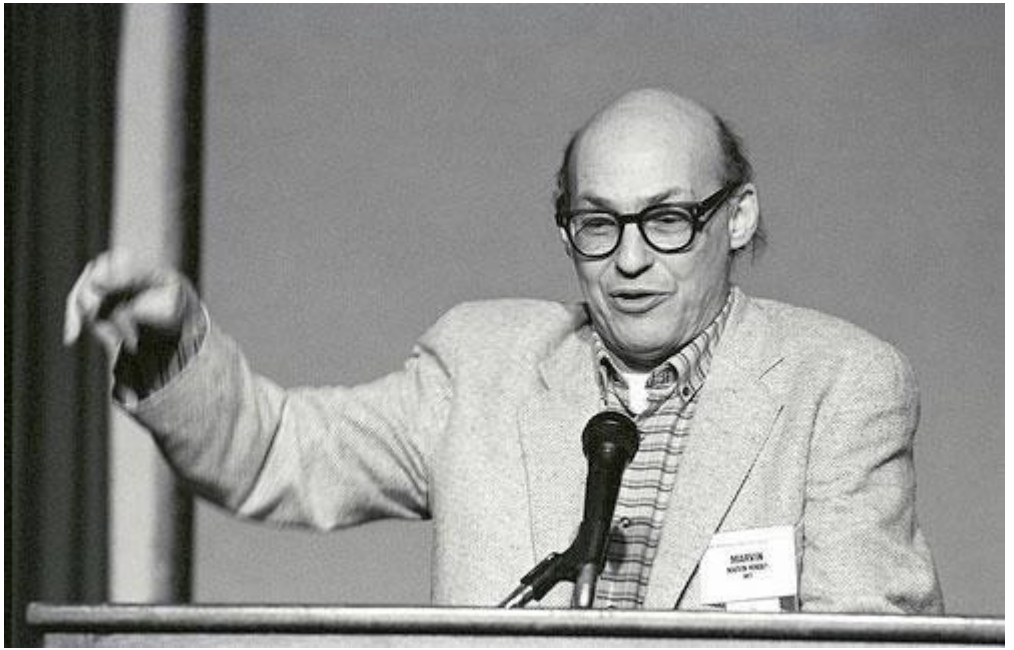


Graduate Student

But how did Computer Vision start?

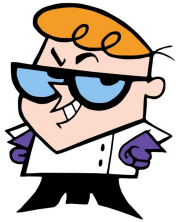


Graduate Student

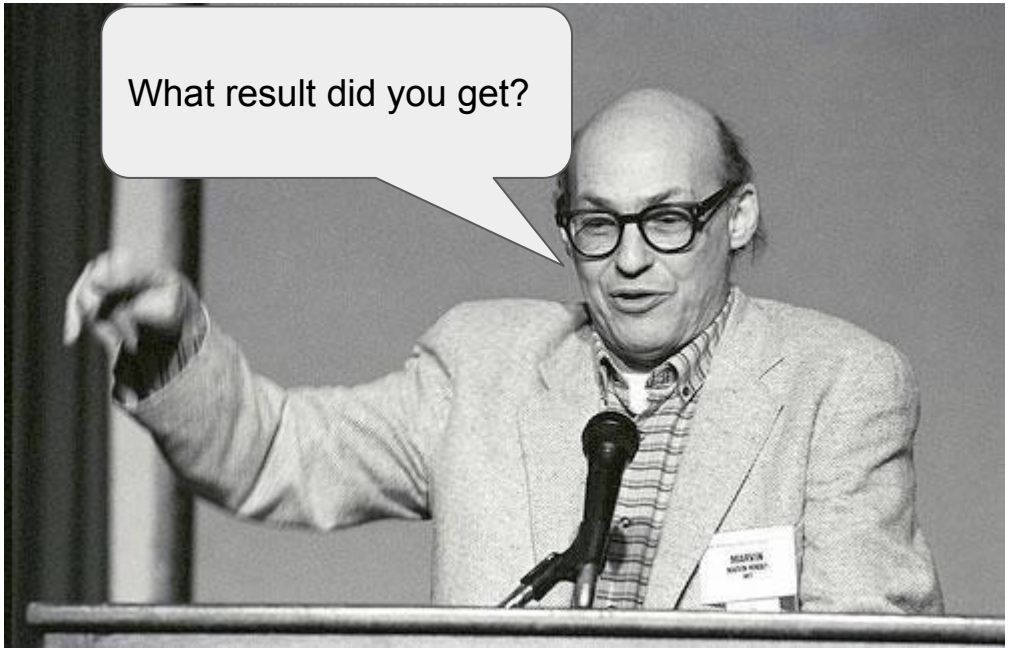


Marvin Minsky

But how did Computer Vision start?



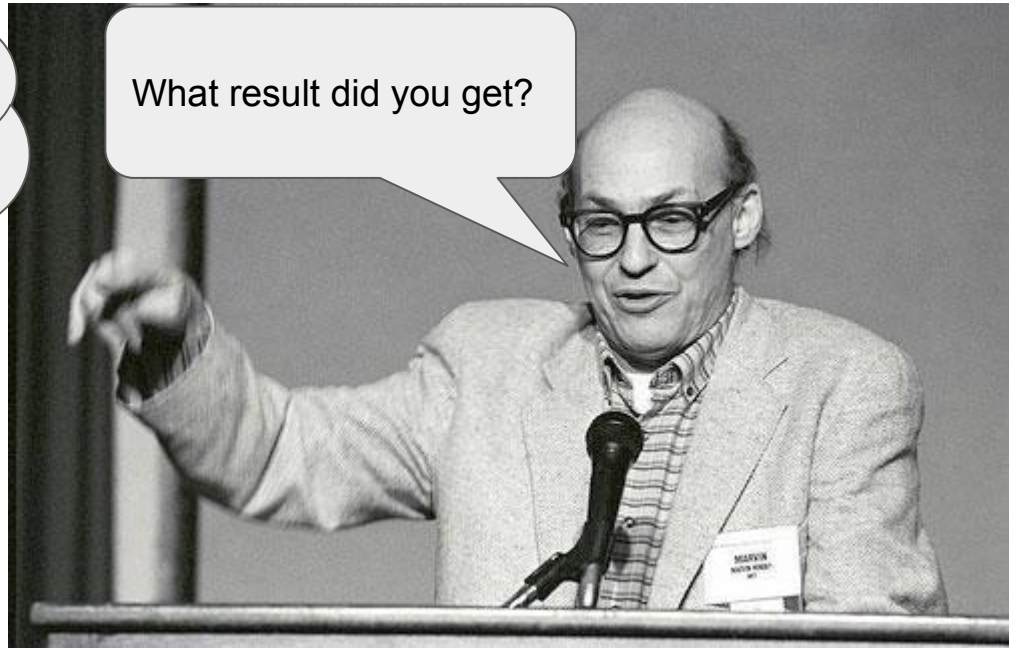
Graduate Student



Marvin Minsky

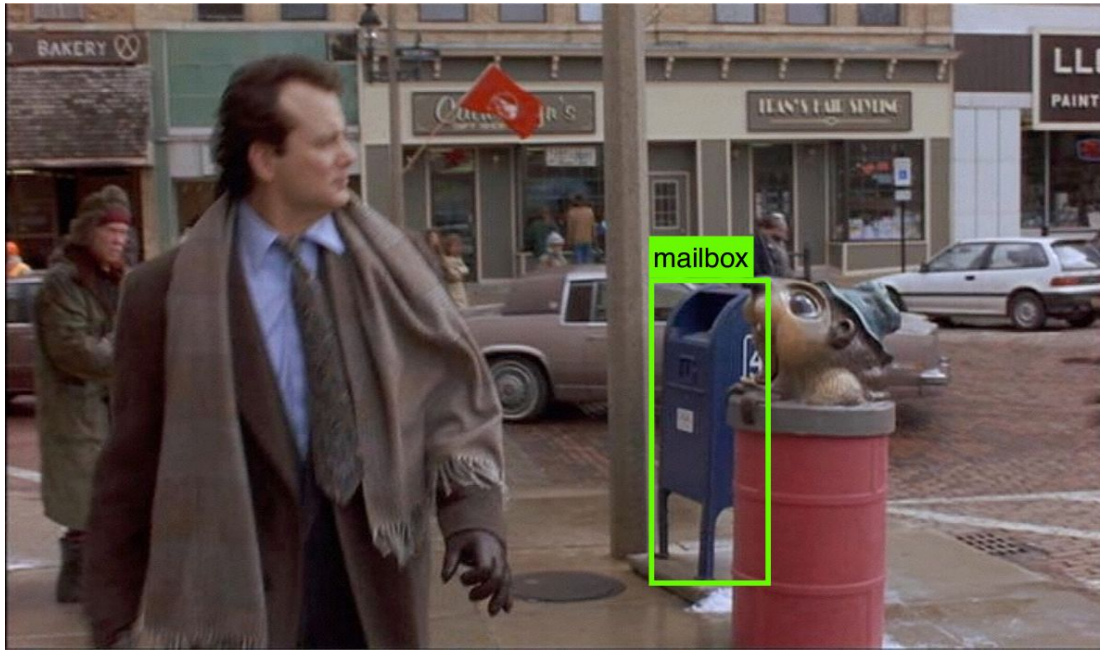
But how did Computer Vision start?

Graduate Student



Marvin Minsky

(...) and where is Computer Vision now?



From [Vedaldi BMVC ‘14]

Exemplar-SVM for Object Detection

Figure 4. **Exemplar-SVMs.** A few "train" exemplars with their top detections on the PASCAL VOC test-set. Note that each exemplar's HOG has its own dimensions. Note also how each detector is specific not just to the train's orientation, but even to the type of train.

Malisiewicz et. al, ICCV, 2011.

Deformable Parts Model

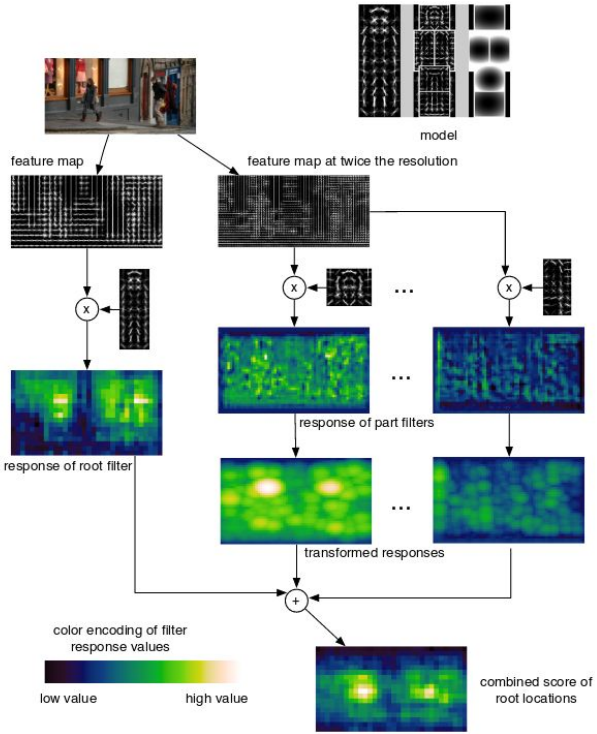


Fig. 4. The matching process at one scale. Responses from the root and part filters are computed a different resolutions in the feature pyramid. The transformed responses are combined to yield a final score for each root location. We show the responses and transformed responses for the "head" and "right shoulder" parts. Note how the "head" filter is more discriminative. The combined scores clearly show two good hypothesis for the object at this scale.

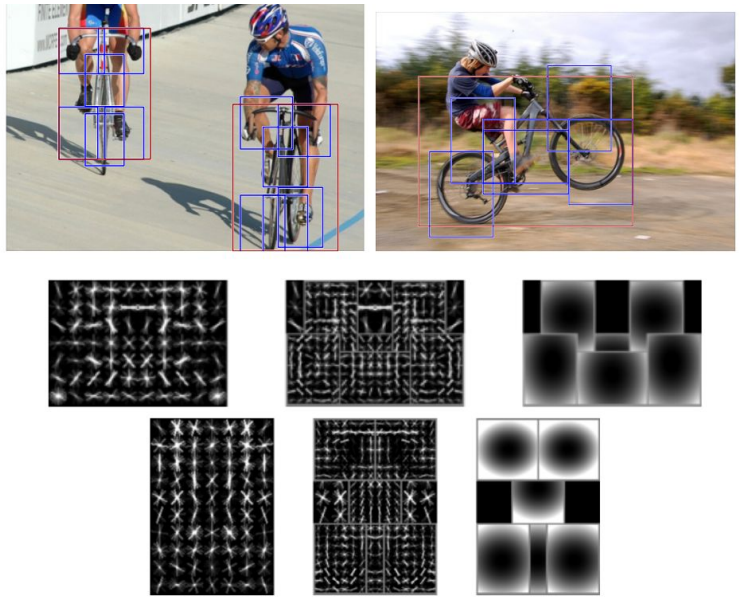
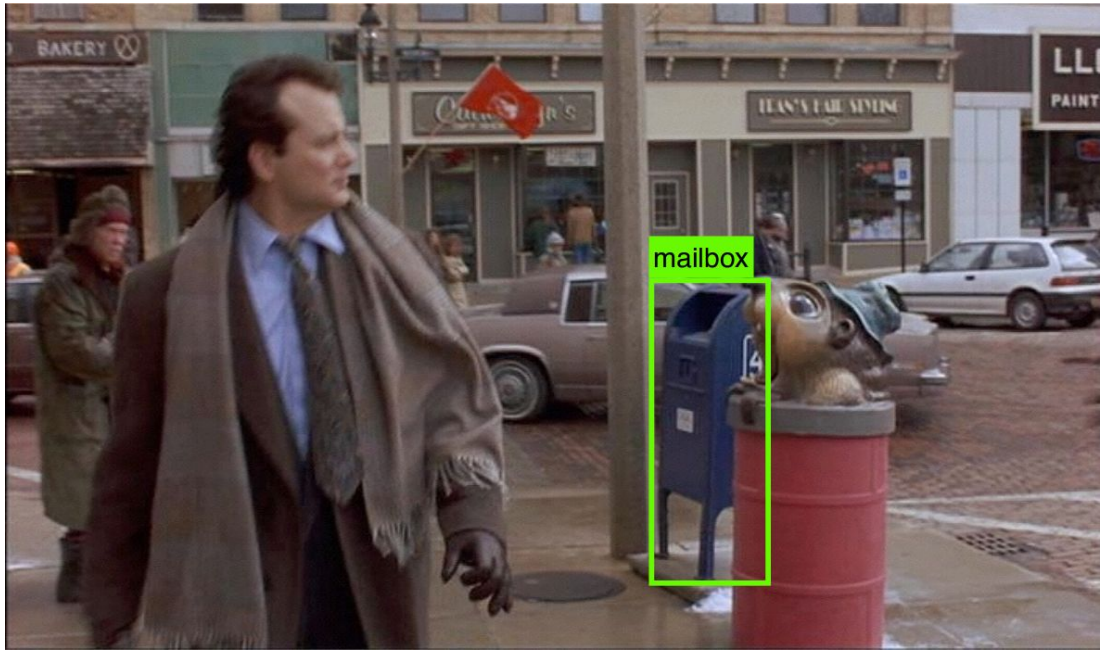


Fig. 2. Detections obtained with a 2 component bicycle model. These examples illustrate the importance of deformations mixture models. In this model the first component captures sideways views of bicycles while the second component captures frontal and near frontal views. The sideways component can deform to match a "wheelie".

Felzenszwalb et al. 2012 (IEEE TPAMI)

Challenges in Computer Vision

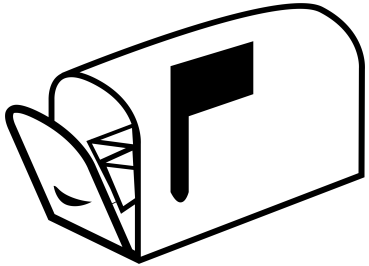
Object detection under different: **pose**, **color**, **illumination**, **occlusion**, **shape** and **instance**.

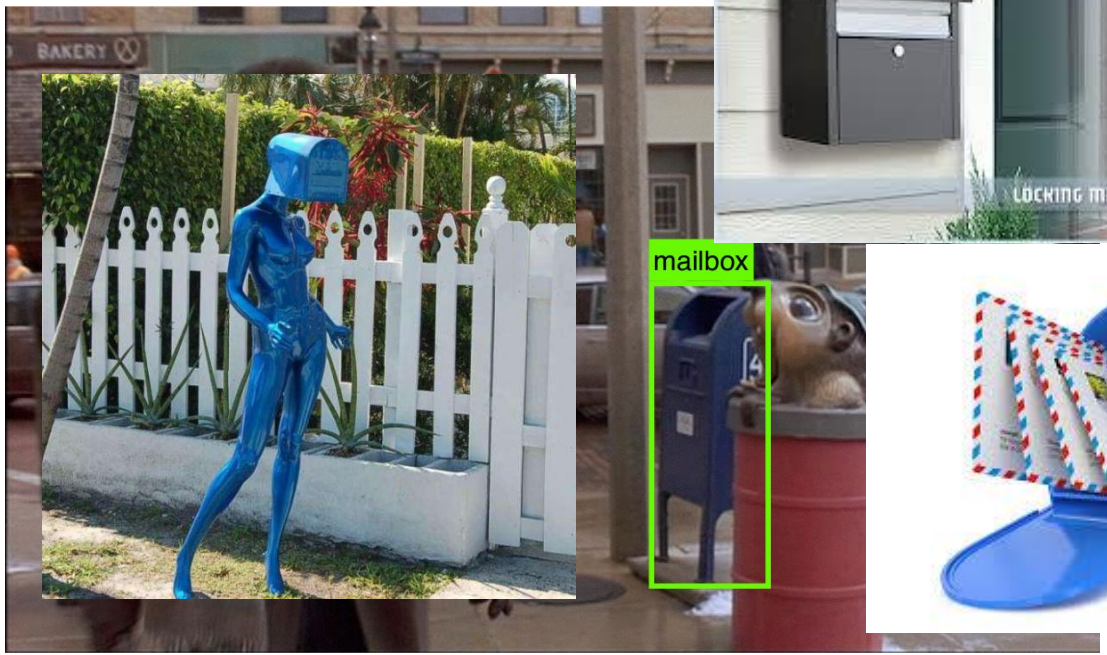


From [Vedaldi BMVC ‘14]

Challenges in Computer Vision

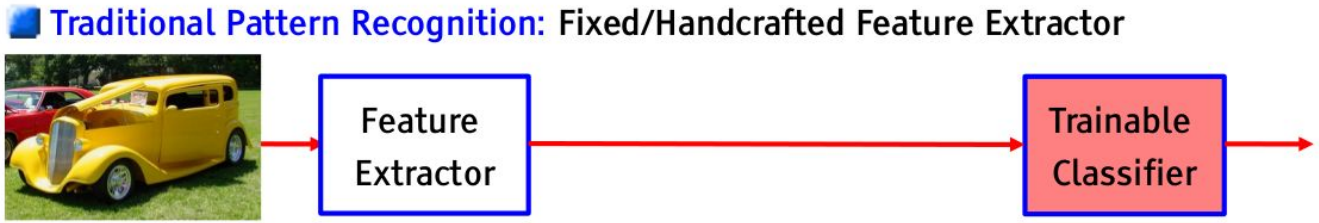
Object detection under different: **pose**, **color**, **illumination**, **occlusion**, **shape** and **instance**.





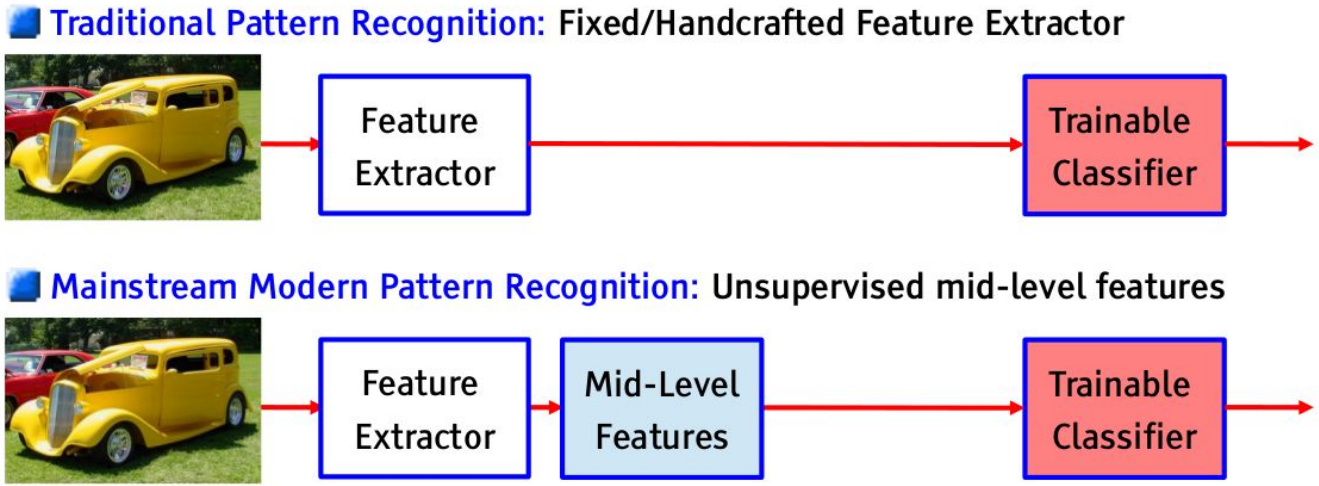
From [Vedaldi BMVC ‘14]

Computer Vision [**Pre**-Deep Learning era] pipeline



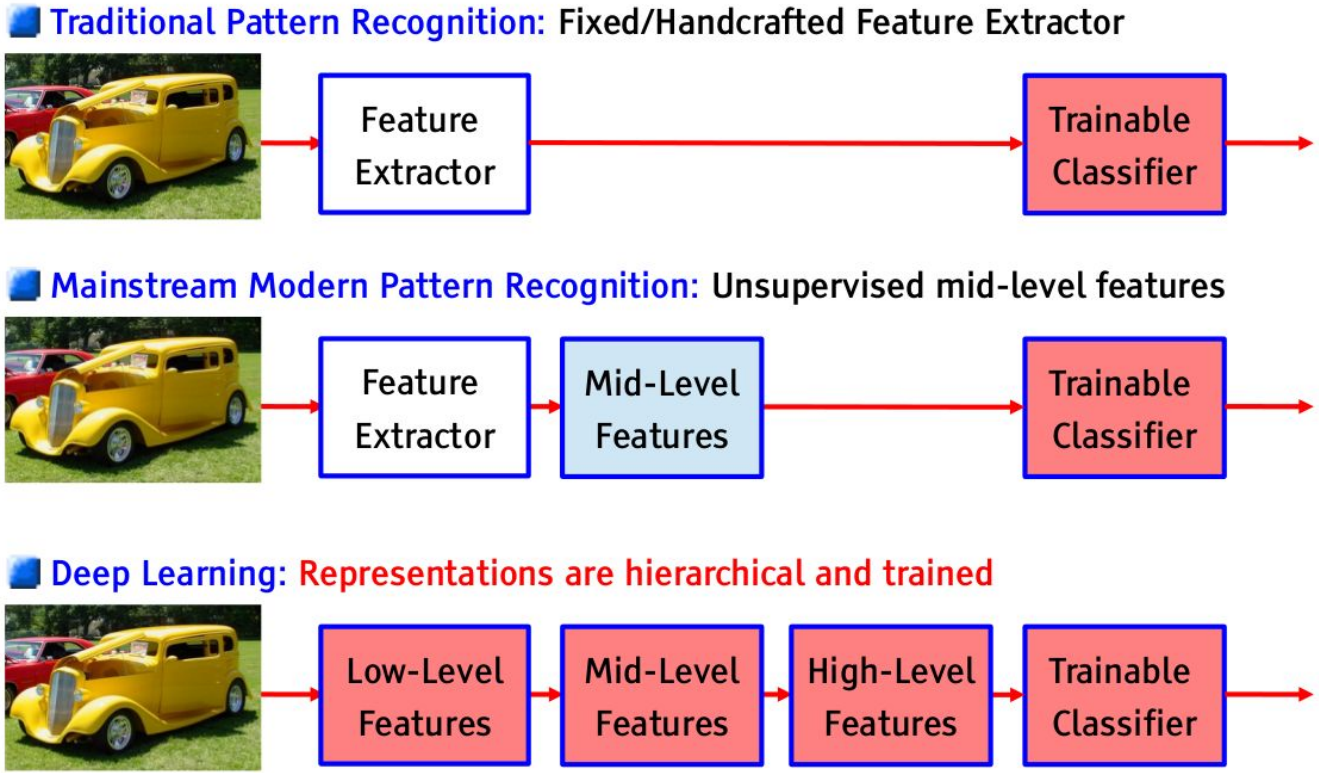
LeCun, 2015

Computer Vision [**Pre**-Deep Learning era] pipeline



LeCun, 2015

Computer Vision [**Post**-Deep Learning era] pipeline

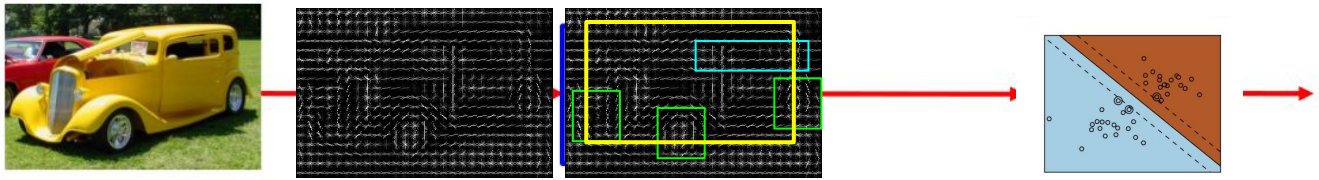


LeCun, 2015

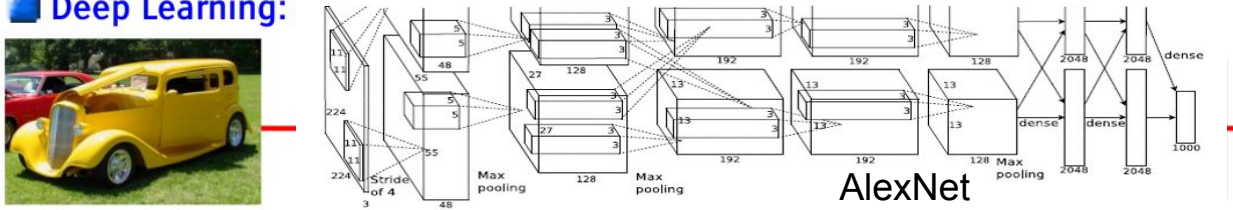
Computer Vision [**Post**-Deep Learning era] pipeline

Traditional Pattern Recognition: Fixed/Handcrafted Feature Extractor

Mainstream Modern Pattern Recognition: Unsupervised mid-level features

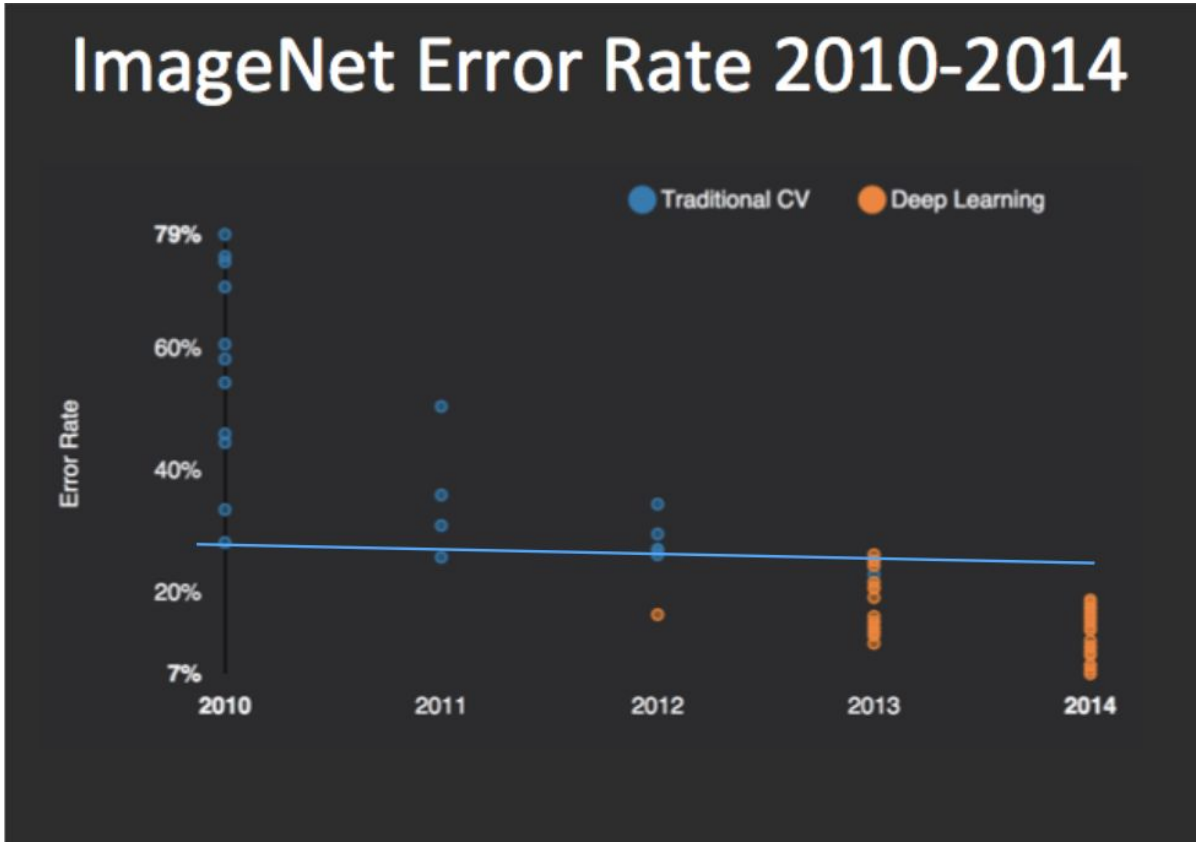


Deep Learning:



LeCun, 2015

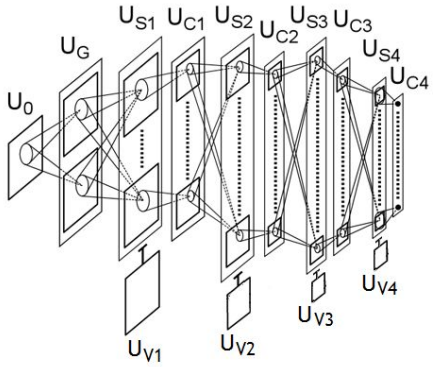
Deep Successes by the numbers



Provided by Matt Zeiler @ Clarifai

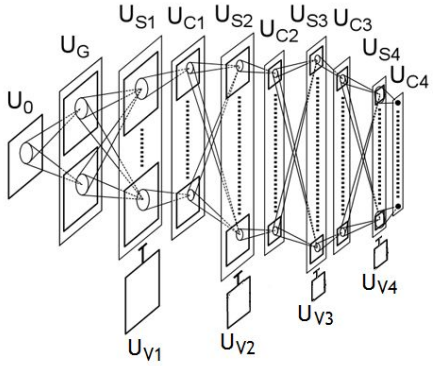
Deep Architectures

Neurocognitron (Fukushima, 1980)

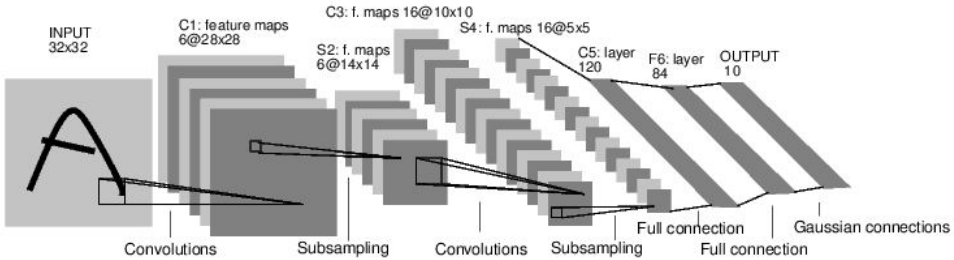


Deep Architectures

Neurocognitron (Fukushima, 1980)

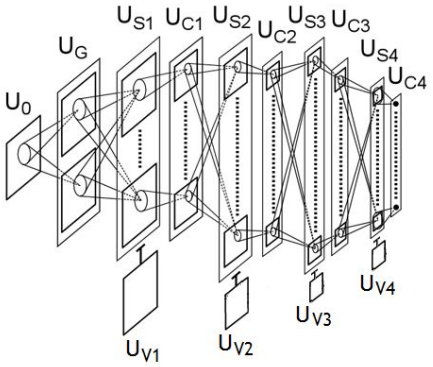


LeNet,1989

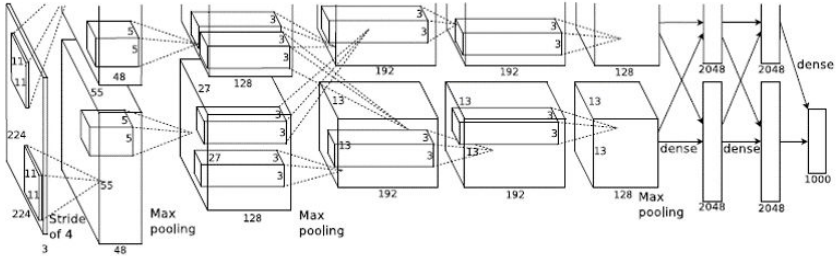


Deep Architectures

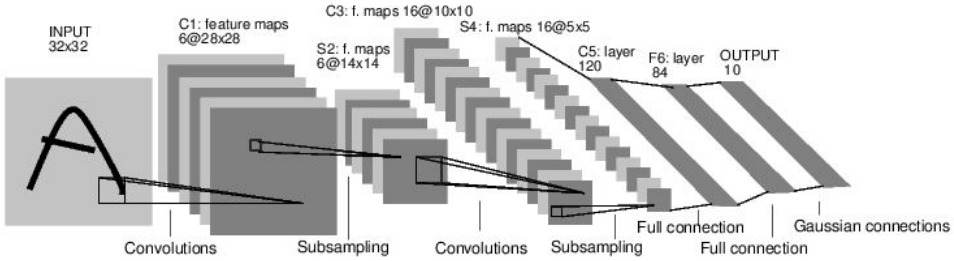
Neurocognitron (Fukushima, 1980)



AlexNet, 2011



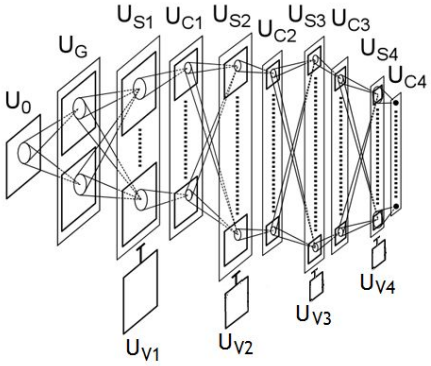
LeNet,1989



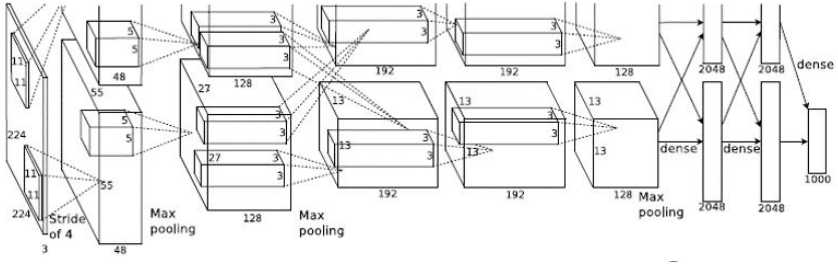
Deep Architectures

VGGNet (Simonyan & Zisserman, 2014)

Neurocognitron (Fukushima, 1980)

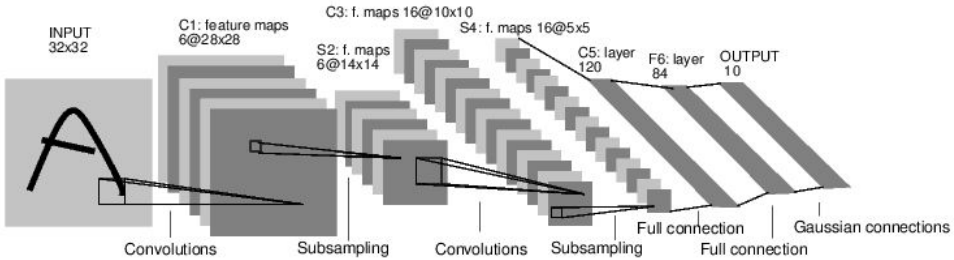


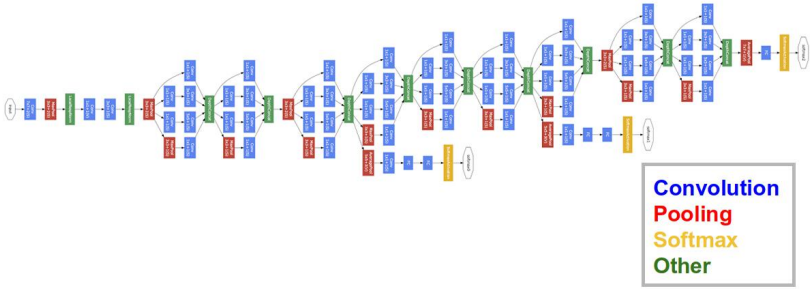
AlexNet, 2011



GoogLeNet, 2014

LeNet,1989

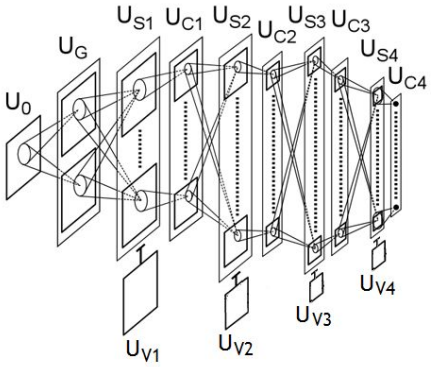




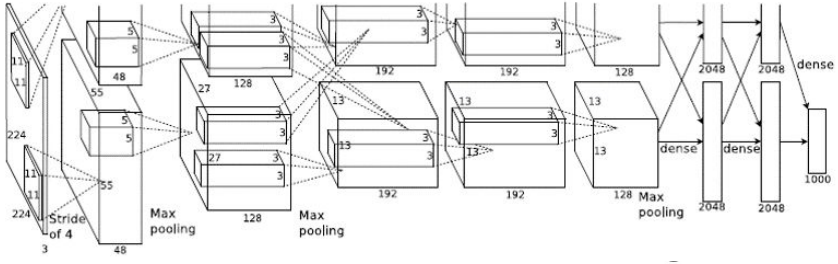
Deep Architectures

VGGNet (Simonyan & Zisserman, 2014)

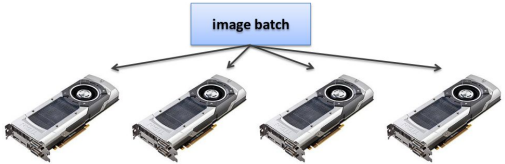
Neurocognitron (Fukushima, 1980)



AlexNet, 2011

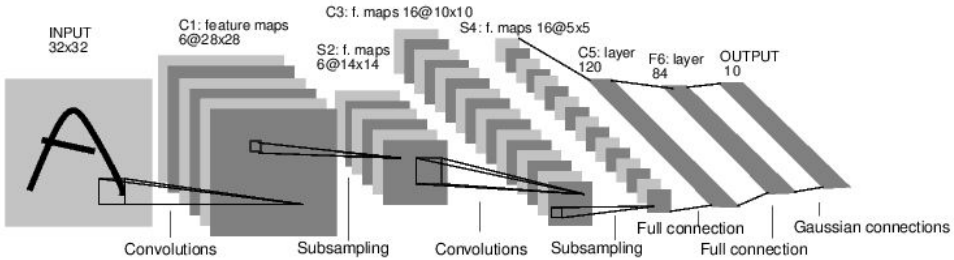


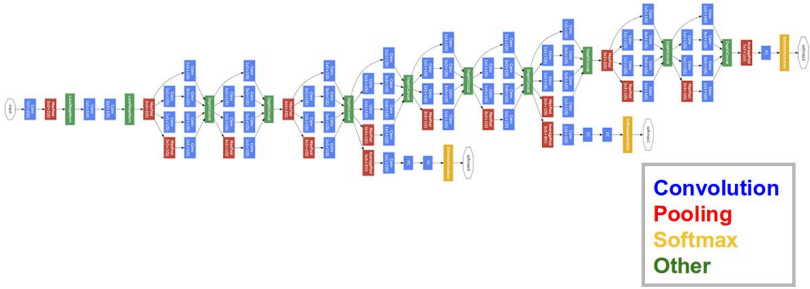
(Ridiculously huge amount of parameters)



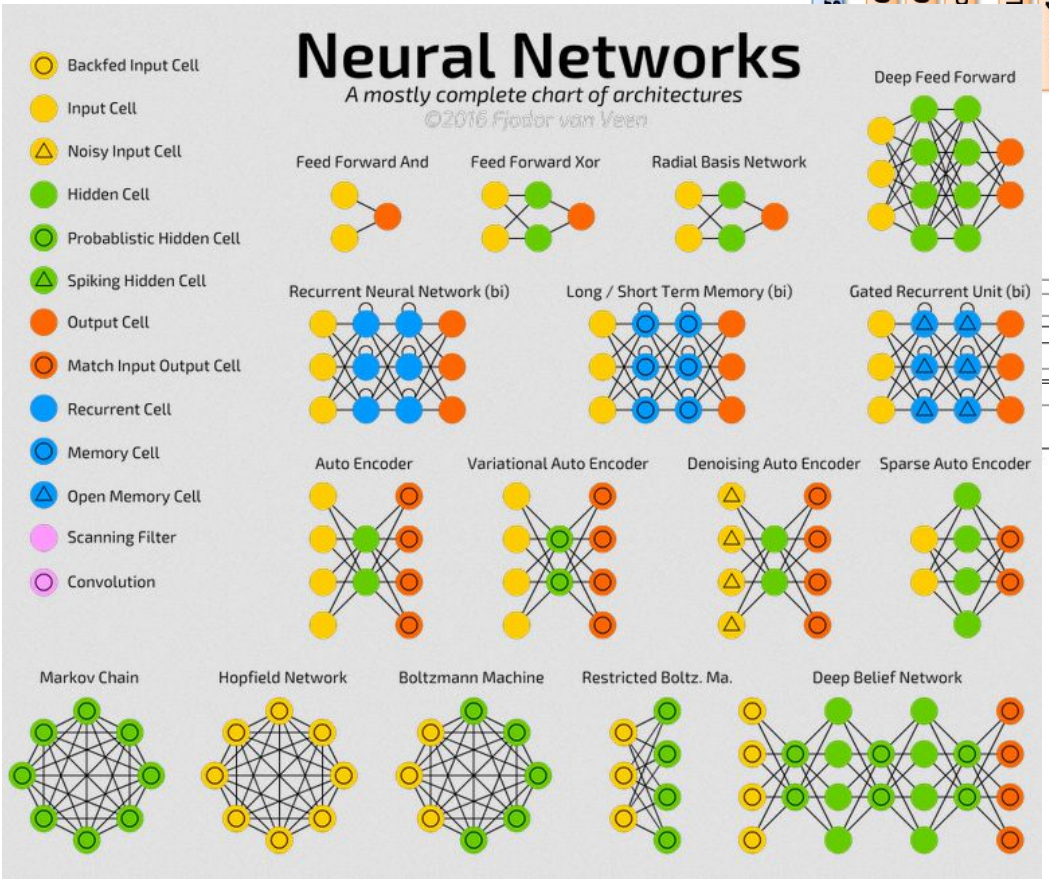
GoogLeNet, 2014

LeNet,1989



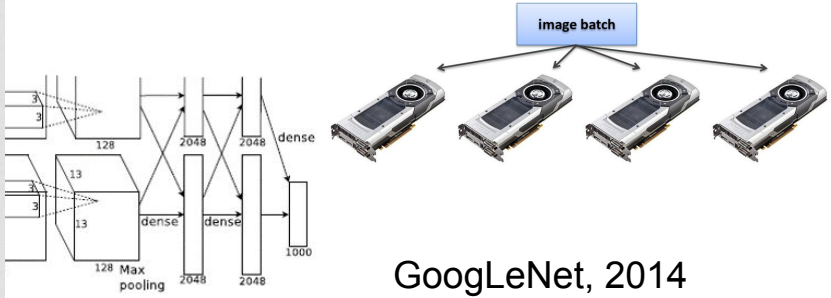


Deep Architectures

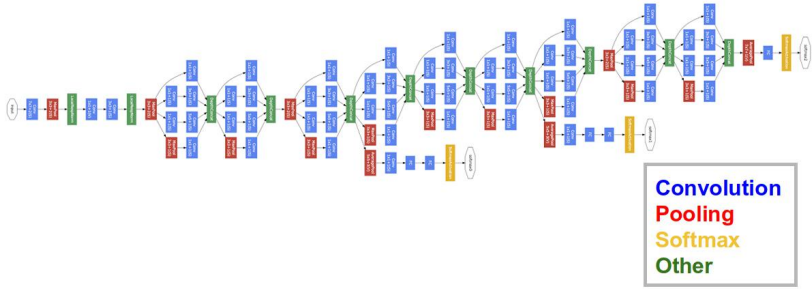


VGGNet (Simonyan & Zisserman, 2014)

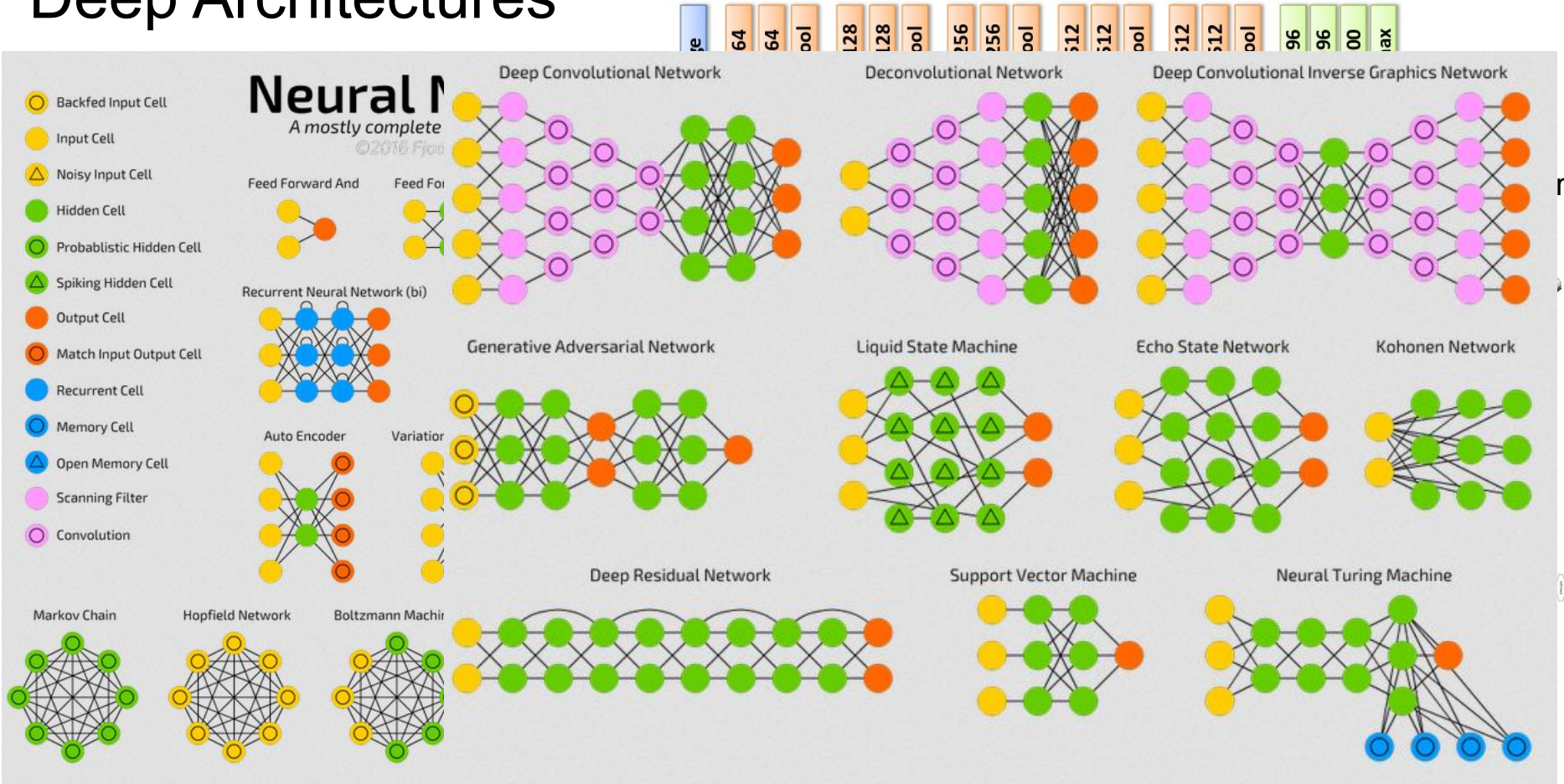
(Ridiculously huge amount of parameters)



GoogLeNet, 2014

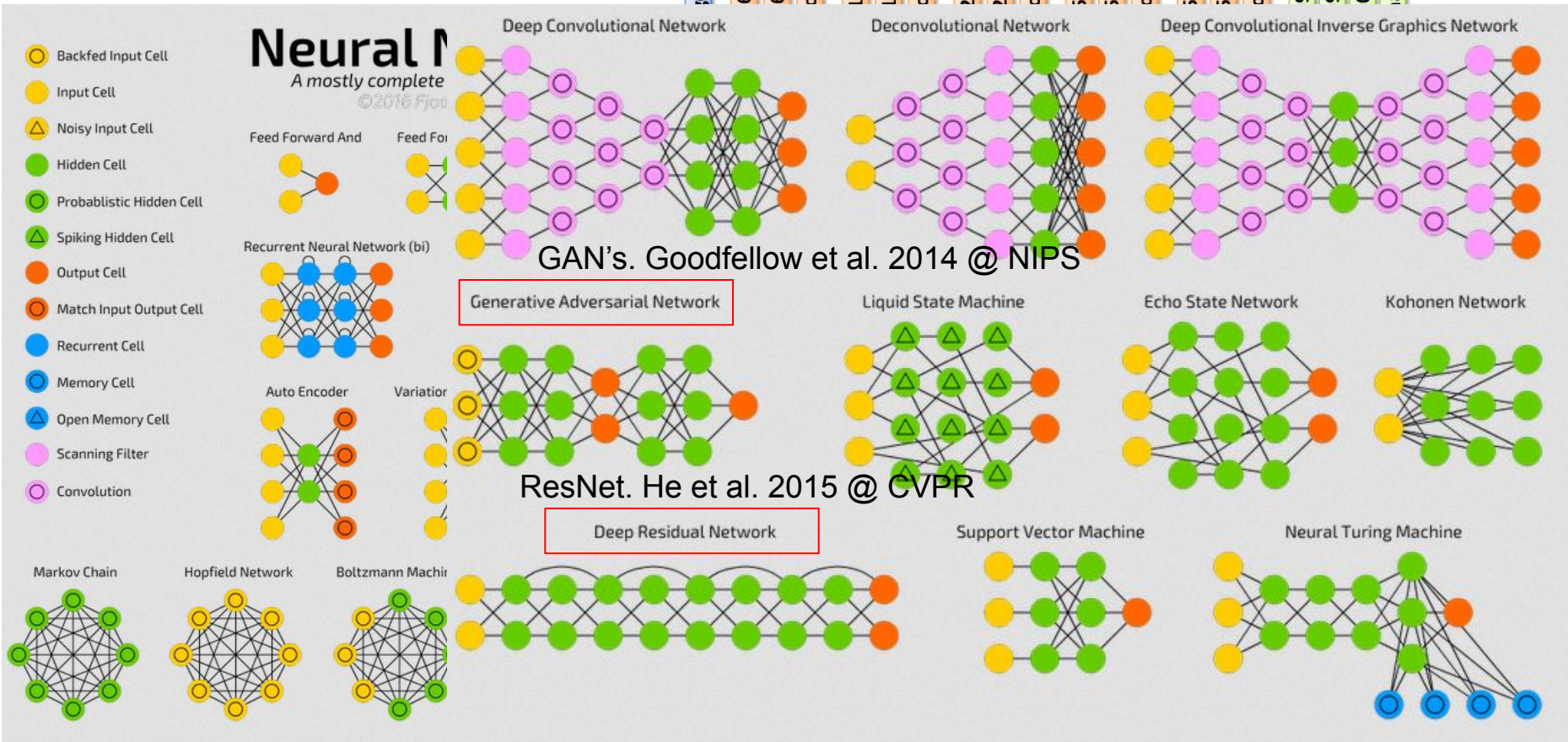


Deep Architectures



Deep Architectures

VGGNet (Simonyan & Zisserman, 2014)



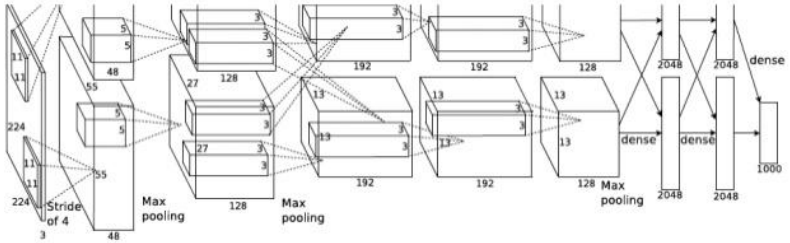
rs)

What are the networks learning?

This argument has been promoted by the vision science community, but still seems as if it is going back to the first AlexNet paper (Krizhevsky, Sutskever & Hinton @ NIPS, 2012)

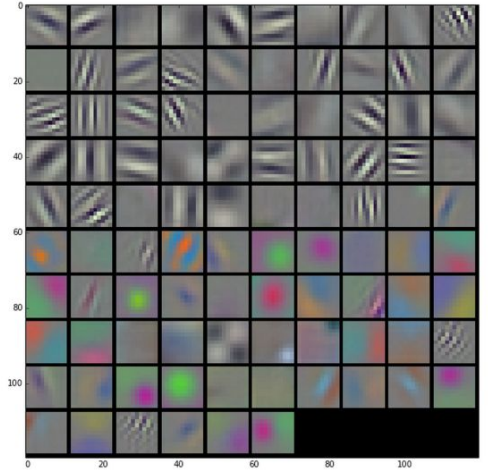
AlexNet

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



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

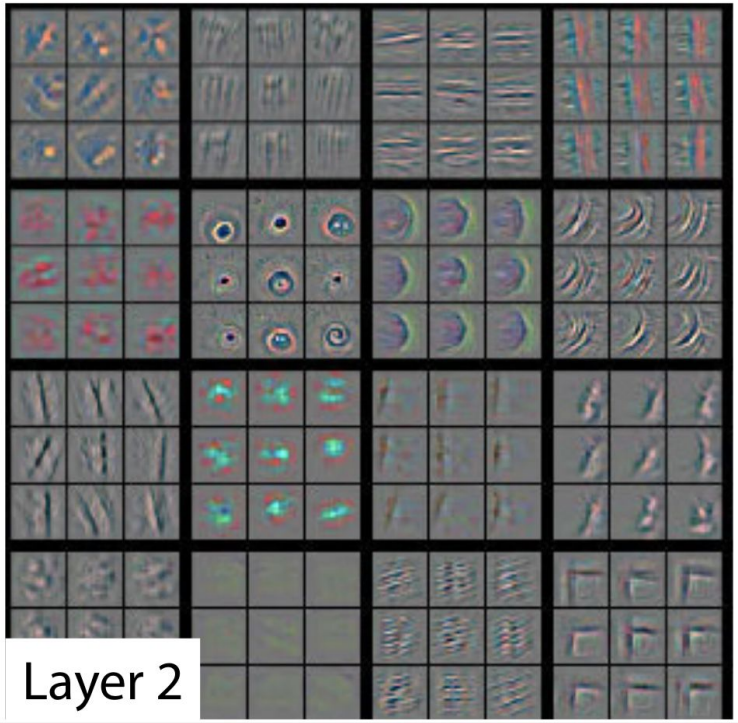
1st layer of AlexNet



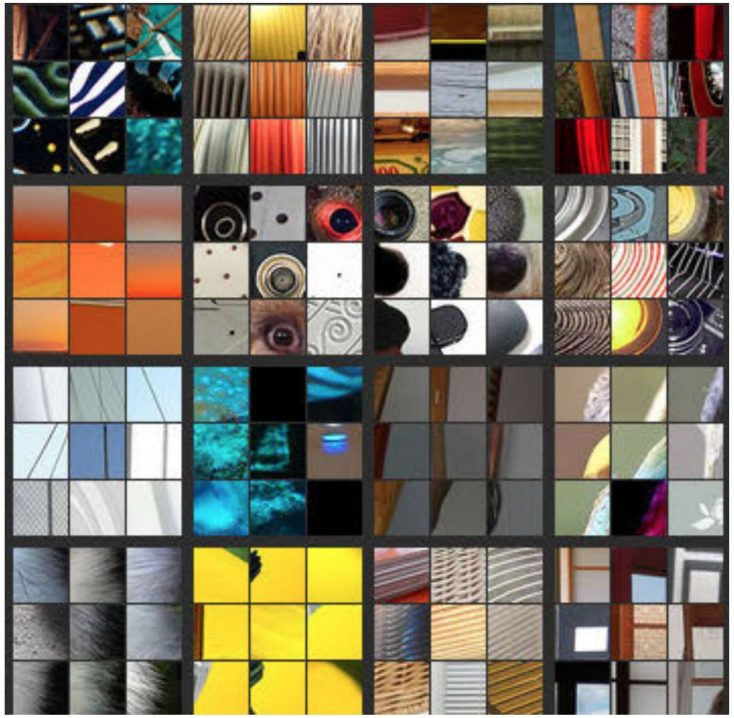
Visualizing Deep Networks

Visualization of AlexNet architecture

Layer 1

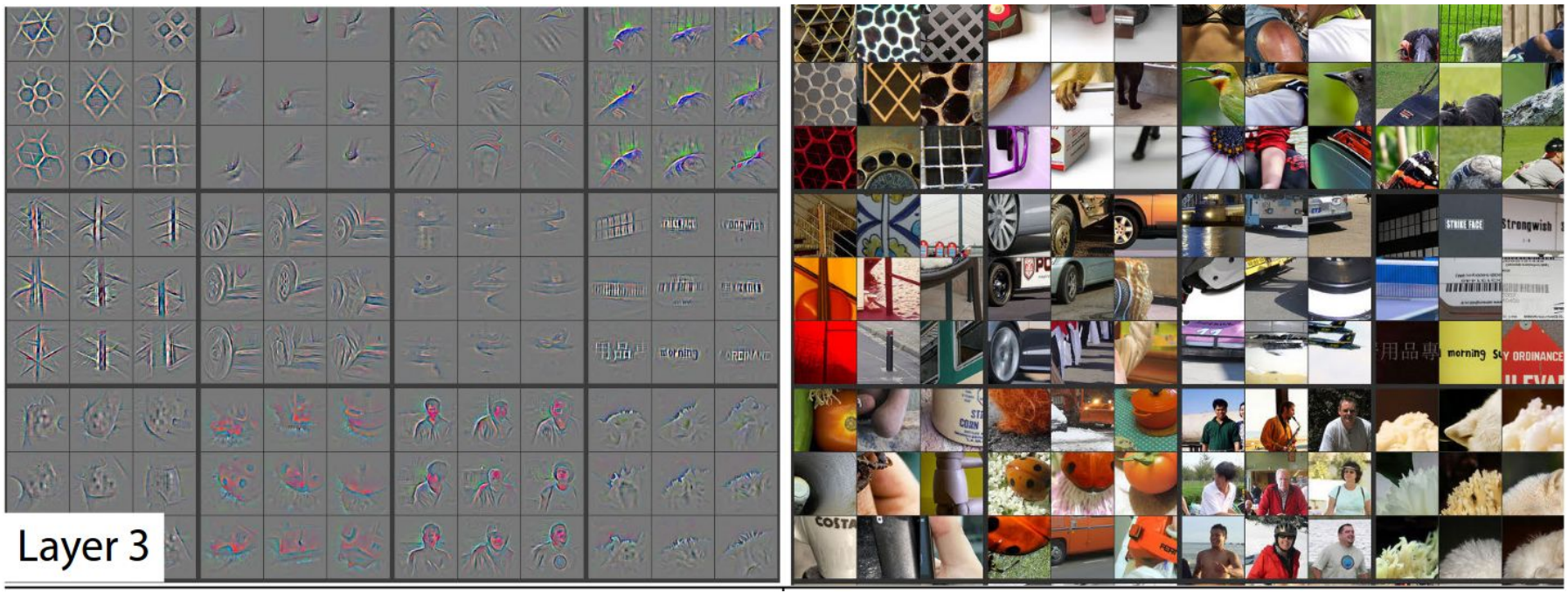


Layer 2



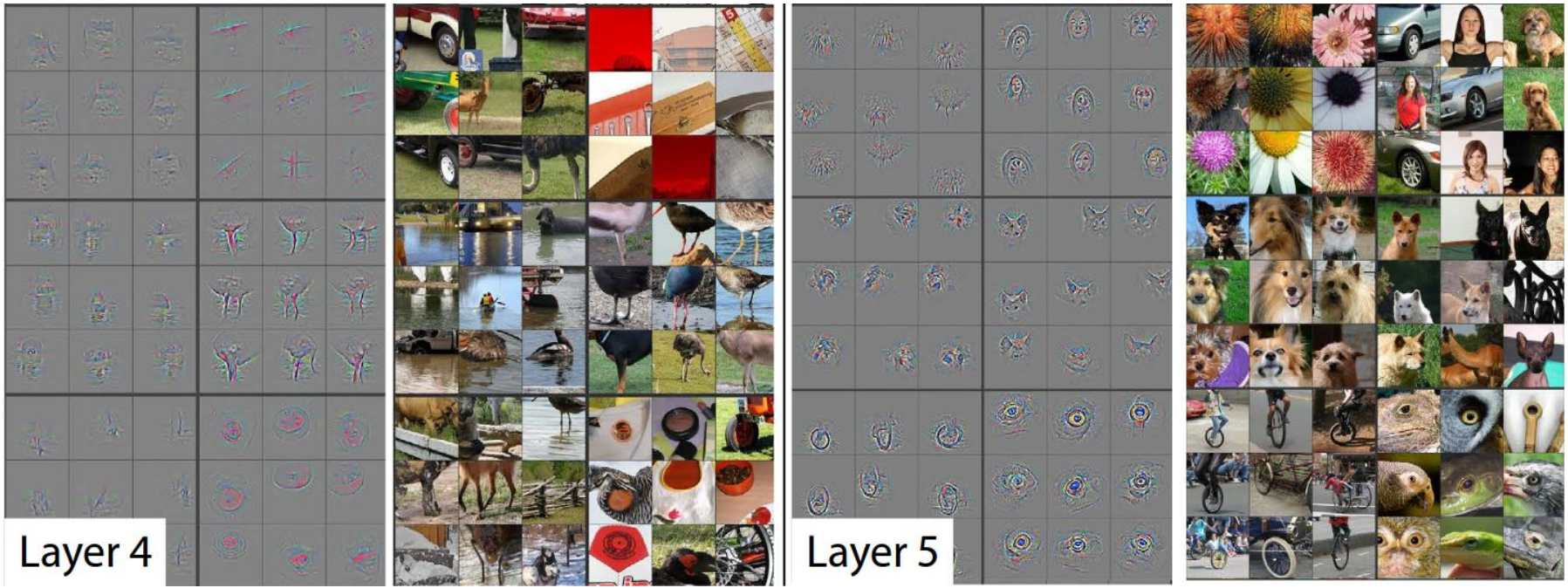
Zeiler & Fergus , 2013

Visualizing Deep Networks



Zeiler & Fergus , 2013

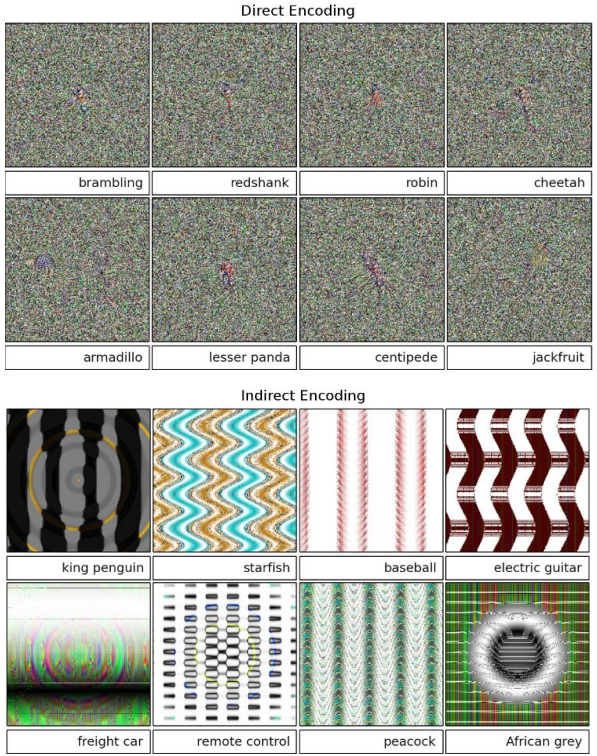
Visualizing Deep Networks



Zeiler & Fergus , 2013

But Deep Networks are still fooled

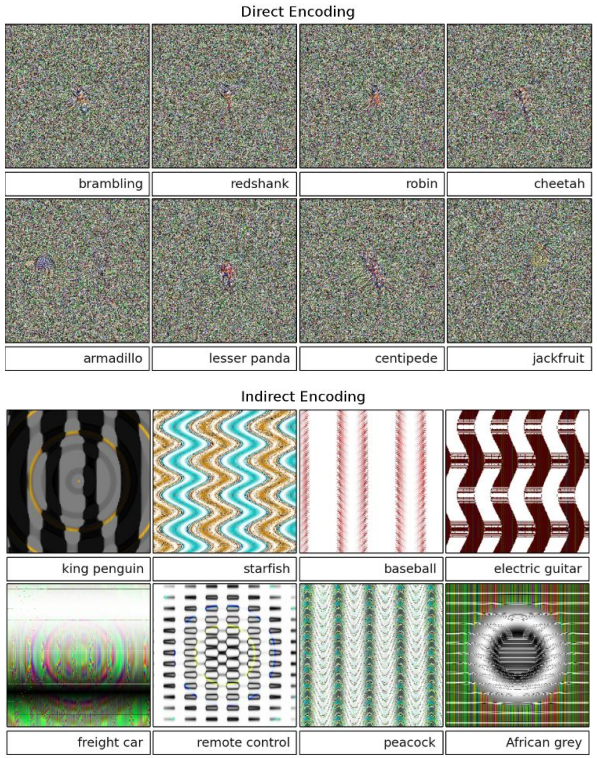
“Deep Confusion”



Nguyen et. al @ CVPR 2015

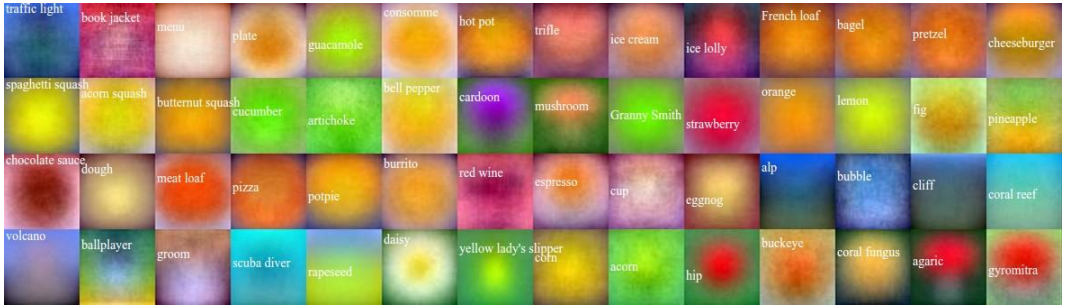
But Deep Networks are still fooled

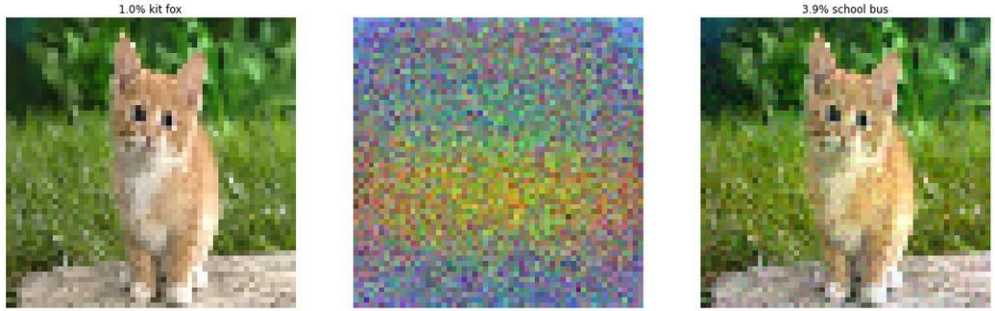
“Deep Confusion”



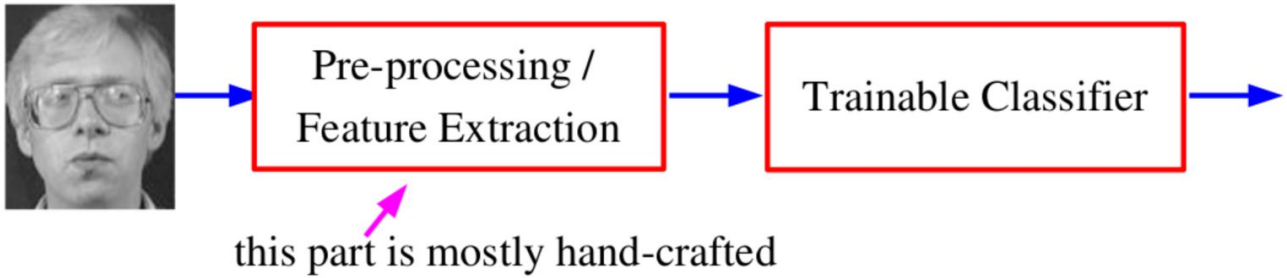
Nguyen et. al @ CVPR 2015

But you can also fool a linear classifier !
(Karpathy’s blog notes, 2015)





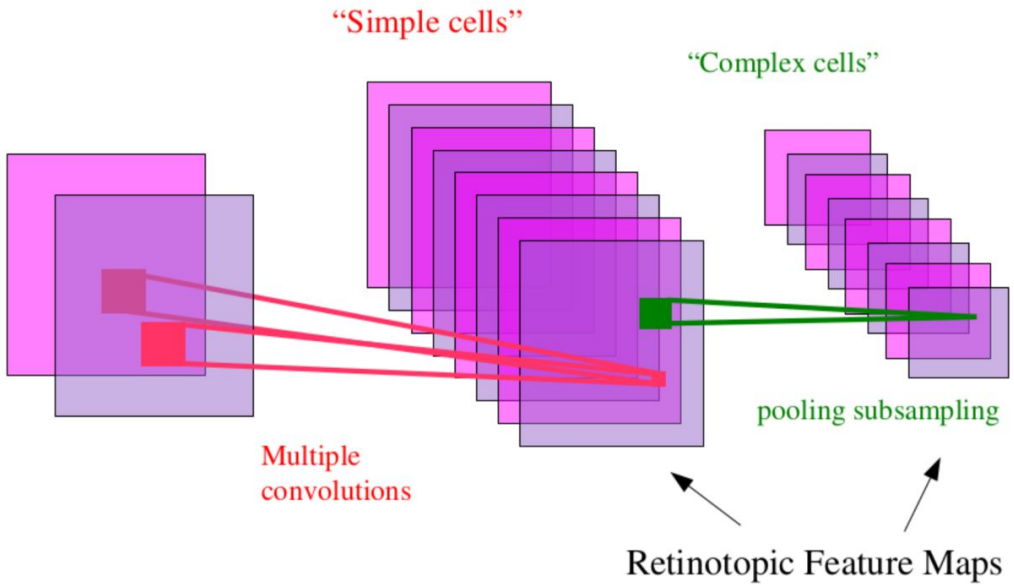
The Traditional Architecture for Recognition



- The raw input is pre-processed through a hand-crafted feature extractor
- The trainable classifier is often generic (task independent)

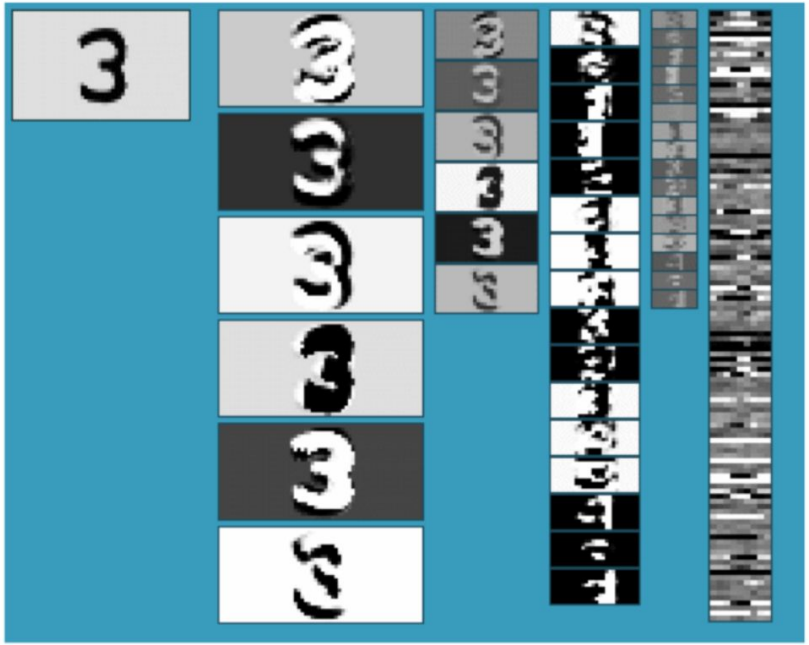
An Old Idea for Local Shift Invariance

- **[Hubel & Wiesel 1962]:**
  - simple cells detect local features
  - complex cells "pool" the outputs of simple cells within a retinotopic neighborhood.

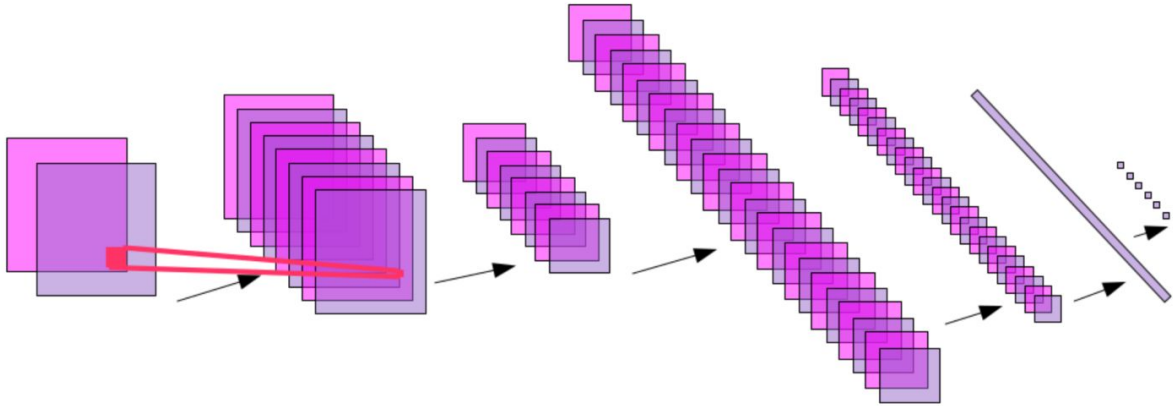


The Multistage Hubel-Wiesel Architecture

- **Building a complete artificial vision system:**
  - Stack multiple stages of simple cells / complex cells layers
  - Higher stages compute more global, more invariant features
  - Stick a classification layer on top
  - [Fukushima 1971-1982]
    - neocognitron
  - [LeCun 1988-2007]
    - convolutional net
  - [Poggio 2002-2006]
    - HMAX
  - [Ullman 2002-2006]
    - fragment hierarchy
  - [Lowe 2006]
    - HMAX
- **QUESTION: How do we find (or learn) the filters?**

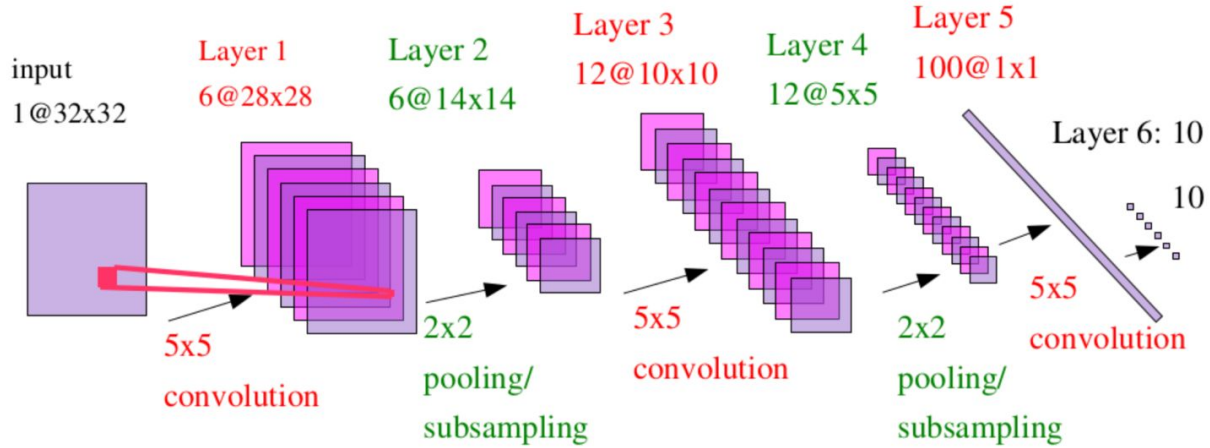


Getting Inspiration from Biology: Convolutional Network



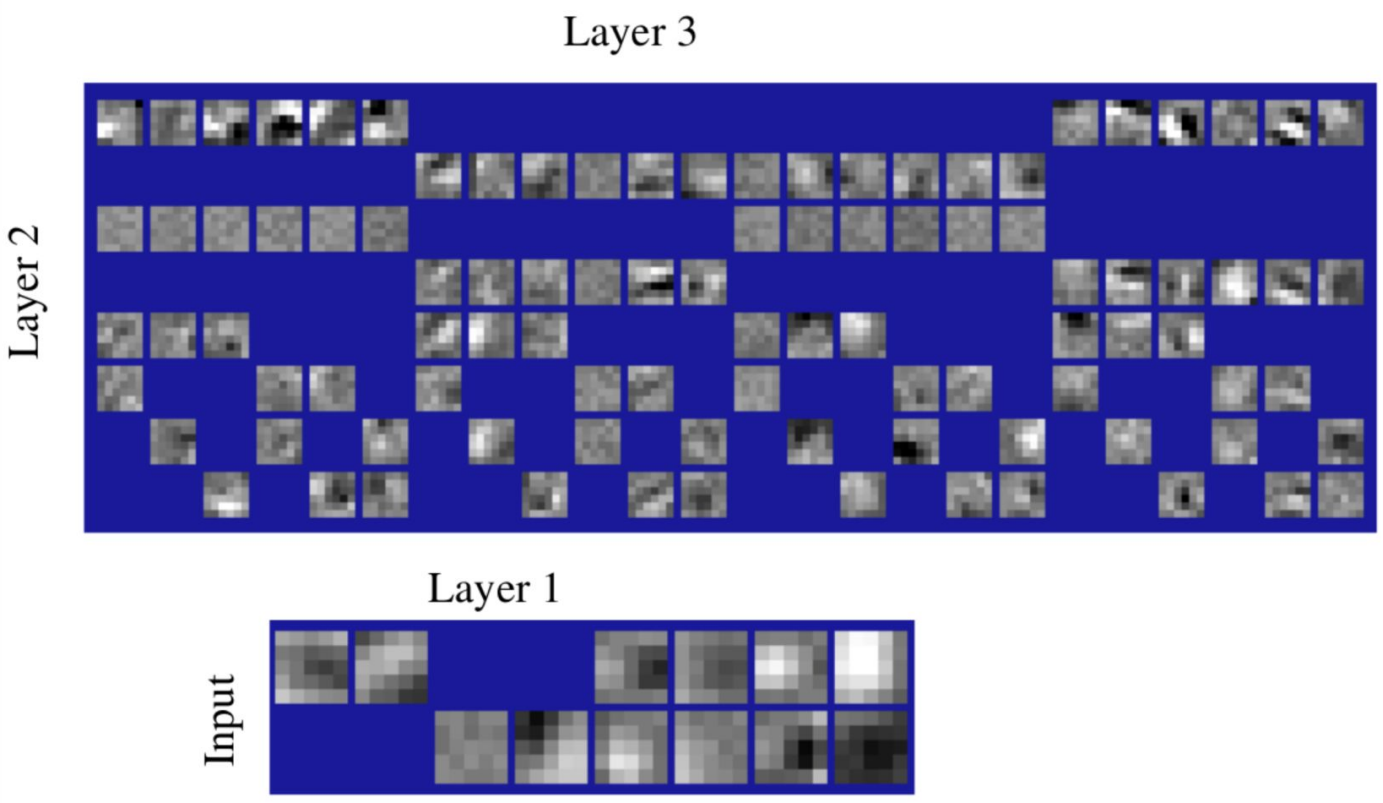
- **Hierarchical/multilayer:** features get progressively more global, invariant, and numerous
- **dense features:** features detectors applied everywhere (no interest point)
- **broadly tuned (possibly invariant) features:** sigmoid units are on half the time.
- **Global discriminative training:** The whole system is trained "end-to-end" with a gradient-based method to minimize a global loss function
- **Integrates segmentation, feature extraction, and invariant classification in one fell swoop.**

Convolutional Net Architecture

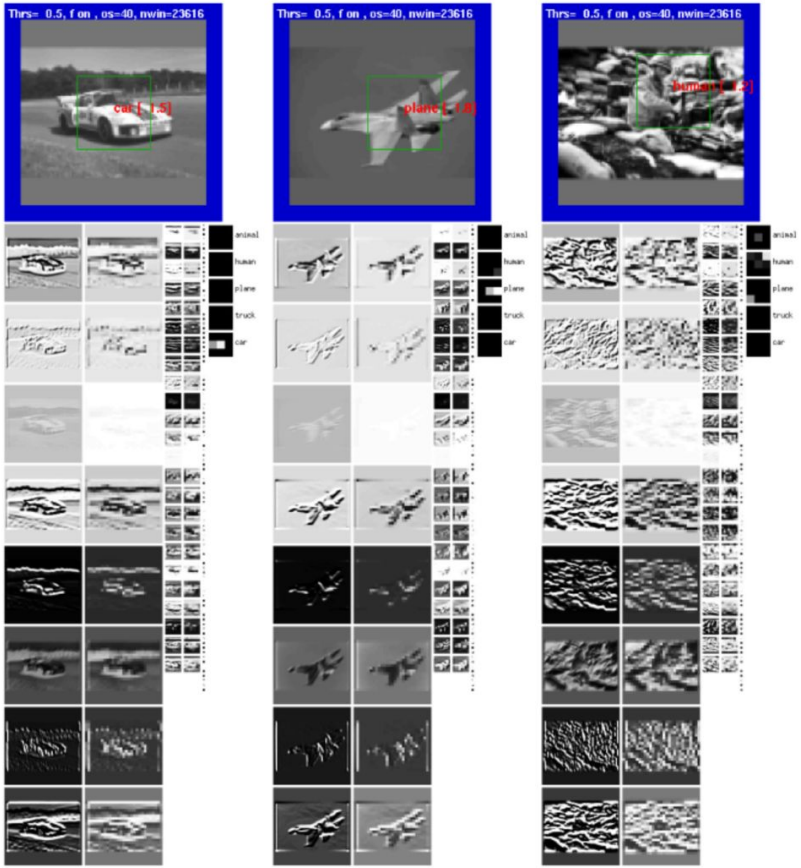


- **Convolutional net for handwriting recognition** (400,000 synapses)
- Convolutional layers (simple cells): all units in a feature plane share the same weights
- Pooling/subsampling layers (complex cells): for invariance to small distortions.
- **Supervised gradient-descent learning using back-propagation**
- **The entire network is trained end-to-end. All the layers are trained simultaneously**.

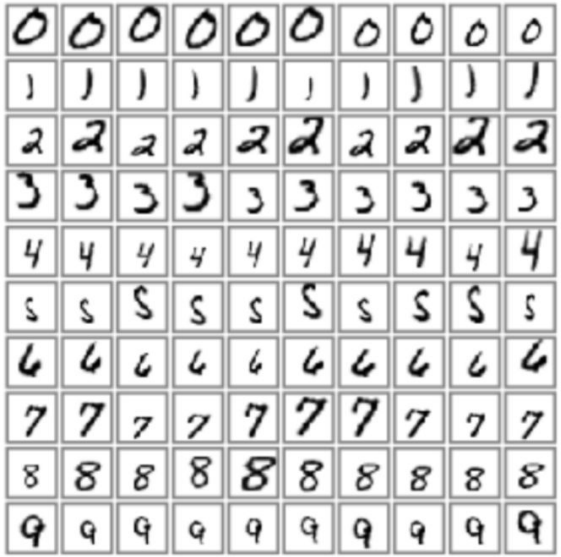
Learned Features



Natural Images (Monocular Mode)



MNIST Handwritten Digit Dataset



- Handwritten Digit Dataset MNIST: 60,000 training samples, 10,000 test samples

Results on MNIST Handwritten Digits

CLASSIFIER	DEFORMATION	PREPROCESSING	ERROR (%)	Reference
linear classifier (1-layer NN)		none	12.00	LeCun et al. 1998
linear classifier (1-layer NN)		deskewing	8.40	LeCun et al. 1998
pairwise linear classifier		deskewing	7.60	LeCun et al. 1998
K-nearest-neighbors, (L2)		none	3.09	Kenneth Wilder, U. Chicago
K-nearest-neighbors, (L2)		deskewing	2.40	LeCun et al. 1998
K-nearest-neighbors, (L2)		deskew, clean, blur	1.80	Kenneth Wilder, U. Chicago
K-NN L3, 2 pixel jitter		deskew, clean, blur	1.22	Kenneth Wilder, U. Chicago
K-NN, shape context matching		shape context feature	0.63	Belongie et al. IEEE PAMI 2002
40 PCA + quadratic classifier		none	3.30	LeCun et al. 1998
1000 RBF + linear classifier		none	3.60	LeCun et al. 1998
K-NN, Tangent Distance		subsamp 16x16 pixels	1.10	LeCun et al. 1998
SVM, Gaussian Kernel		none	1.40	
SVM deg 4 polynomial		deskewing	1.10	LeCun et al. 1998
Reduced Set SVM deg 5 poly		deskewing	1.00	LeCun et al. 1998
Virtual SVM deg-9 poly	Affine	none	0.80	LeCun et al. 1998
V-SVM, 2-pixel jittered		none	0.68	DeCoste and Scholkopf, MLJ 2002
V-SVM, 2-pixel jittered		deskewing	0.56	DeCoste and Scholkopf, MLJ 2002
2-layer NN, 300 HU, MSE		none	4.70	LeCun et al. 1998
2-layer NN, 300 HU, MSE,	Affine	none	3.60	LeCun et al. 1998
2-layer NN, 300 HU		deskewing	1.60	LeCun et al. 1998
3-layer NN, 500+150 HU		none	2.95	LeCun et al. 1998
3-layer NN, 500+150 HU	Affine	none	2.45	LeCun et al. 1998
3-layer NN, 500+300 HU, CE, reg		none	1.53	Hinton, unpublished, 2005
2-layer NN, 800 HU, CE		none	1.60	Simard et al., ICDAR 2003
2-layer NN, 800 HU, CE	Affine	none	1.10	Simard et al., ICDAR 2003
2-layer NN, 800 HU, MSE	Elastic	none	0.90	Simard et al., ICDAR 2003
2-layer NN, 800 HU, CE	Elastic	none	0.70	Simard et al., ICDAR 2003
Convolutional net LeNet-1		subsamp 16x16 pixels	1.70	LeCun et al. 1998
Convolutional net LeNet-4		none	1.10	LeCun et al. 1998
Convolutional net LeNet-5,		none	0.95	LeCun et al. 1998
Conv. net LeNet-5,	Affine	none	0.80	LeCun et al. 1998
Boosted LeNet-4	Affine	none	0.70	LeCun et al. 1998
Conv. net, CE	Affine	none	0.60	Simard et al., ICDAR 2003
Comv net, CE	Elastic	none	0.40	Simard et al., ICDAR 2003

Supervised Convolutional Nets: Pros and Cons

- **Convolutional nets can be trained to perform a wide variety of visual tasks.**
  - Global supervised gradient descent can produce parsimonious architectures
- **BUT: they require lots of labeled training samples**
  - 60,000 samples for handwriting
  - 120,000 samples for face detection
  - 25,000 to 350,000 for object recognition
- **Since low-level features tend to be non task specific, we should be able to learn them unsupervised.**
- **Hinton has shown that layer-by-layer unsupervised "pre-training" can be used to initialize "deep" architectures**
  - [Hinton & Shalakhutdinov, Science 2006]
- **Can we use this idea to reduce the number of necessary labeled examples.**