

Computer Vision and Cnn

Raghavendra Singh

Ingredients of Neural Networks

- * Data

- * Learning Concepts **represented** by Data

- * Problem

- * Converted to a **differential** loss function

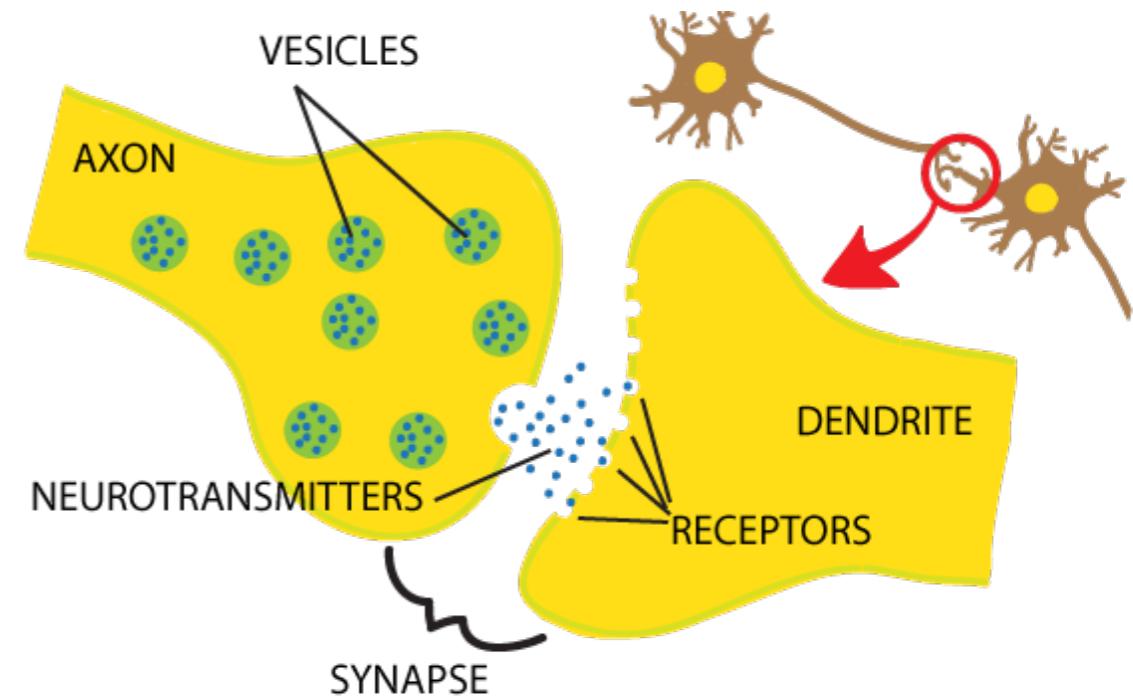
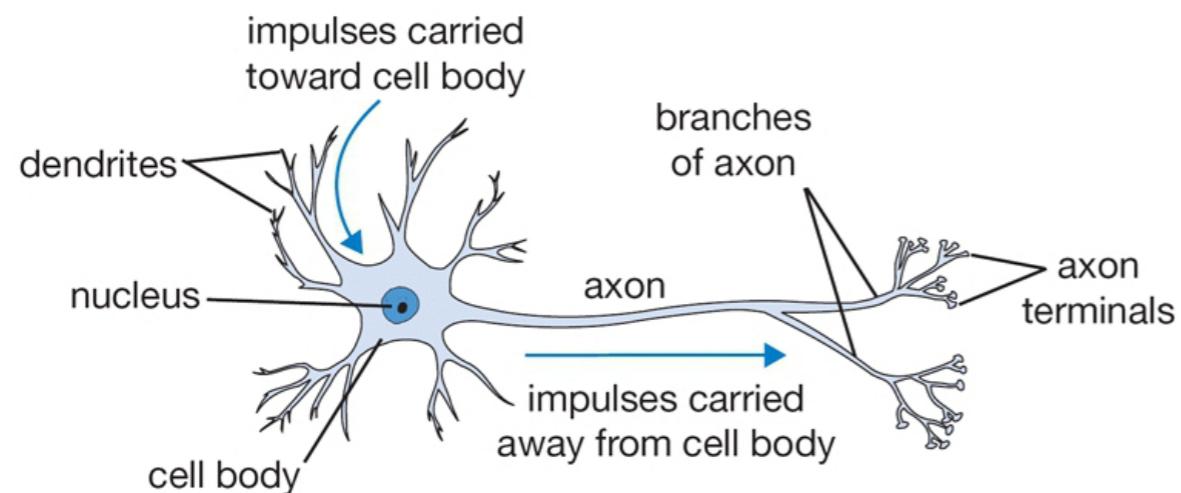
- * Neurons

- * Non Linearity

- * Architecture

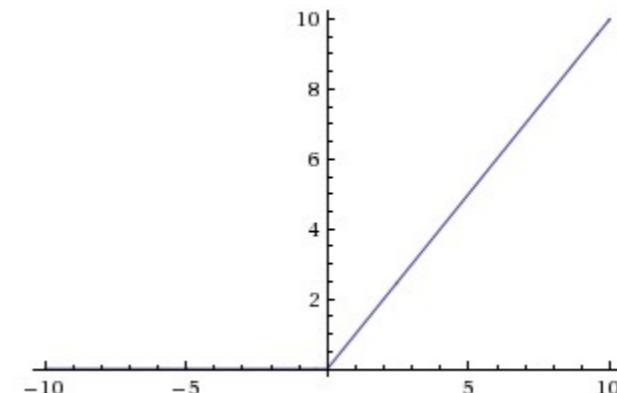
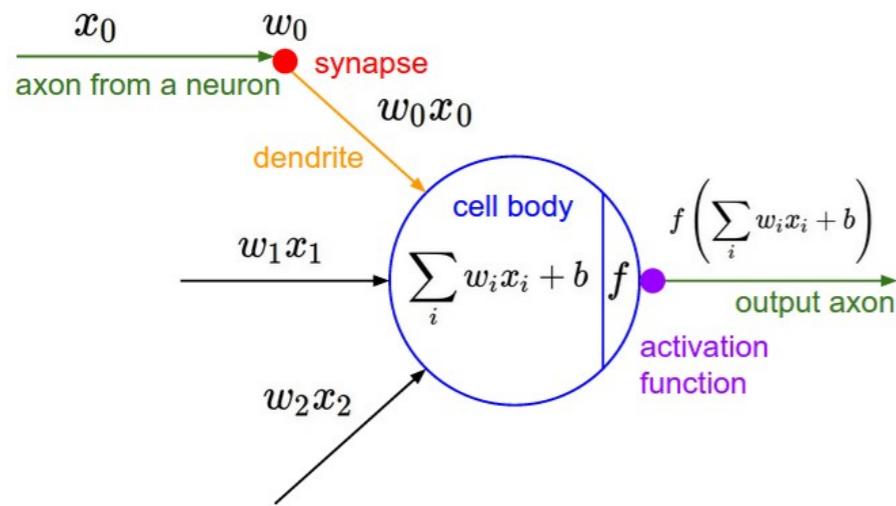
- * How are neurons interconnected

Biological Neurons



- * Integrate signals on dendrites
- * If result greater than a threshold send out an **impulse** along axon
 - * Leak potential if no incoming signal
- * Impulse carries **time stamp**
- * Neurotransmitters in **synapse** modulate signal strength
 - * Neurotransmitter properties change as brain “learns”

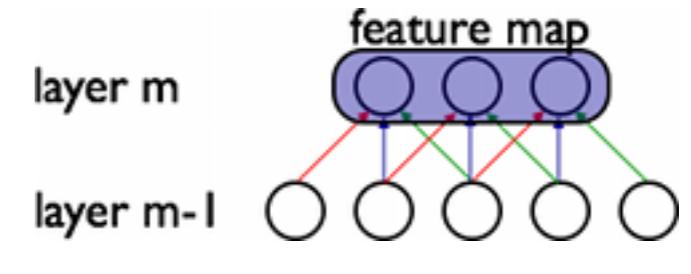
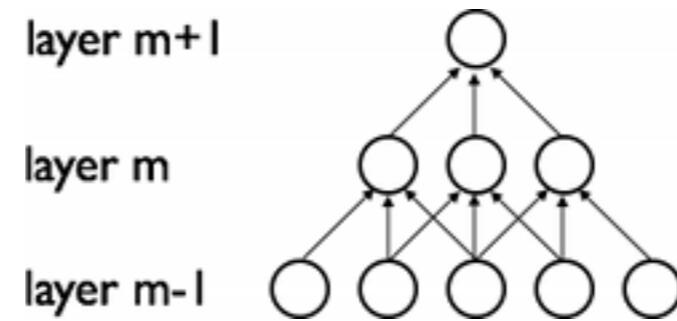
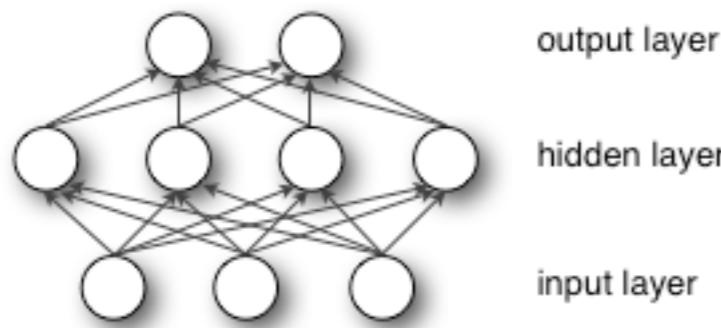
Artificial Neurons



ReLU

- * Sum weighted inputs
- * Activation function is a non-linear function, e.g. ReLU
- * No time stamp
- * Synapse — weights that multiply input
 - * Weights and biases are “learned” from data

Feedforward Architectures



* MLP

- * All neurons in this layer **connected to all** neurons in the next layer

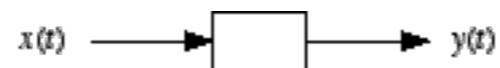
* Sparse MLP

- * Some neurons in this layer **connected to a** neuron in next layer
 - * Some usually defined by concept of locality

* Sparse + Shared MLP == CNN

- * Connections have **shared** weights

Linear Time Invariant Systems



$$\delta(t) = \begin{cases} 1 & t = 0 \\ 0 & t \neq 0 \end{cases}$$

$$x(t) \equiv \sum_j \delta(t - \tau_j) x(\tau_j)$$

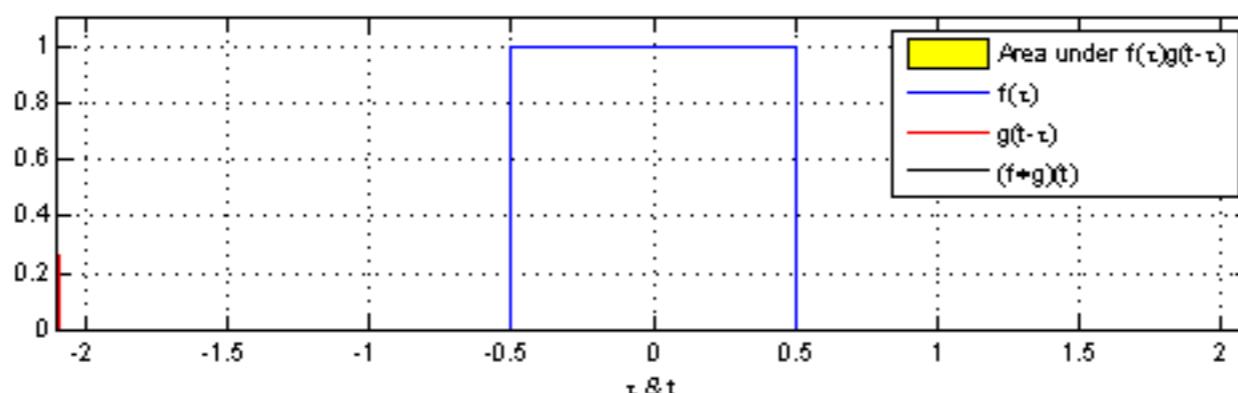
linearity -principle
of superposition

$h(t)$
Impulse response

$$\begin{aligned}\delta(t) &\rightarrow h(t) \\ \delta(t - \tau_j) &\rightarrow h(t - \tau_j) \\ x(\tau_j) \delta(t - \tau_j) &\rightarrow x(\tau_j) h(t - \tau_j) \\ \sum_j x(\tau_j) \delta(t - \tau_j) &\rightarrow \sum_j x(\tau_j) h(t - \tau_j)\end{aligned}$$

$$y(t) \equiv \sum_j x(\tau_j) h(t - \tau_j)$$

Convolution



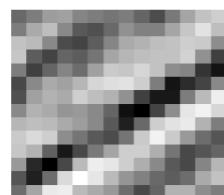
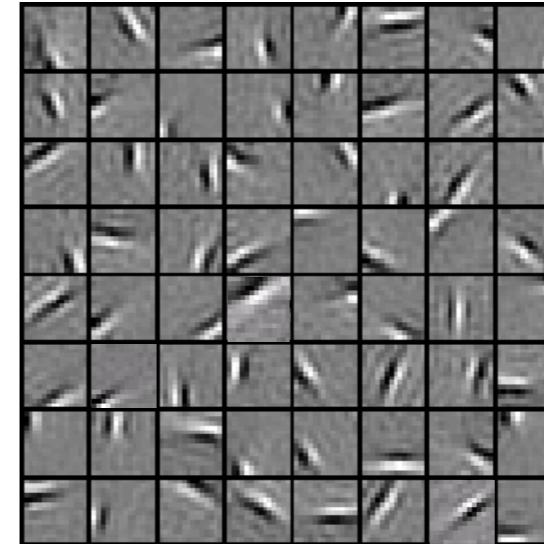
Images == LSI Signals

- * Images often modelled as linear shift invariant signals

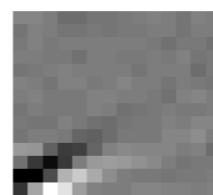
Natural Images



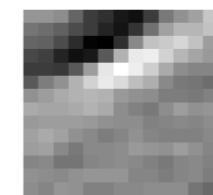
Learned Basis



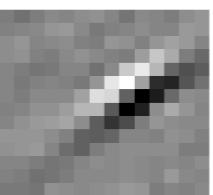
$\approx 0.8 * \quad$



$+ 0.3 * \quad$



$+ 0.5 * \quad$



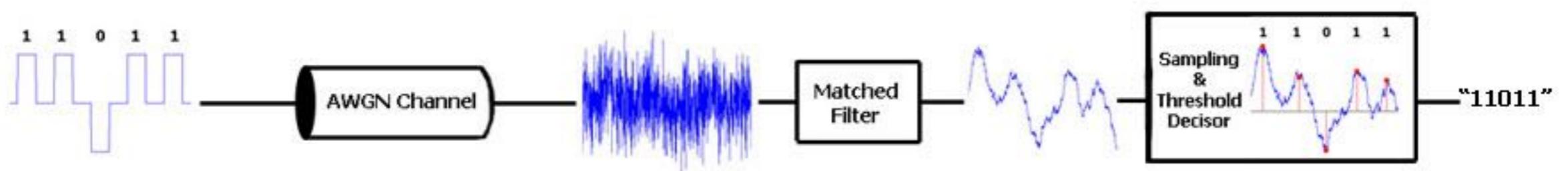
Decomposition

Image Filtering

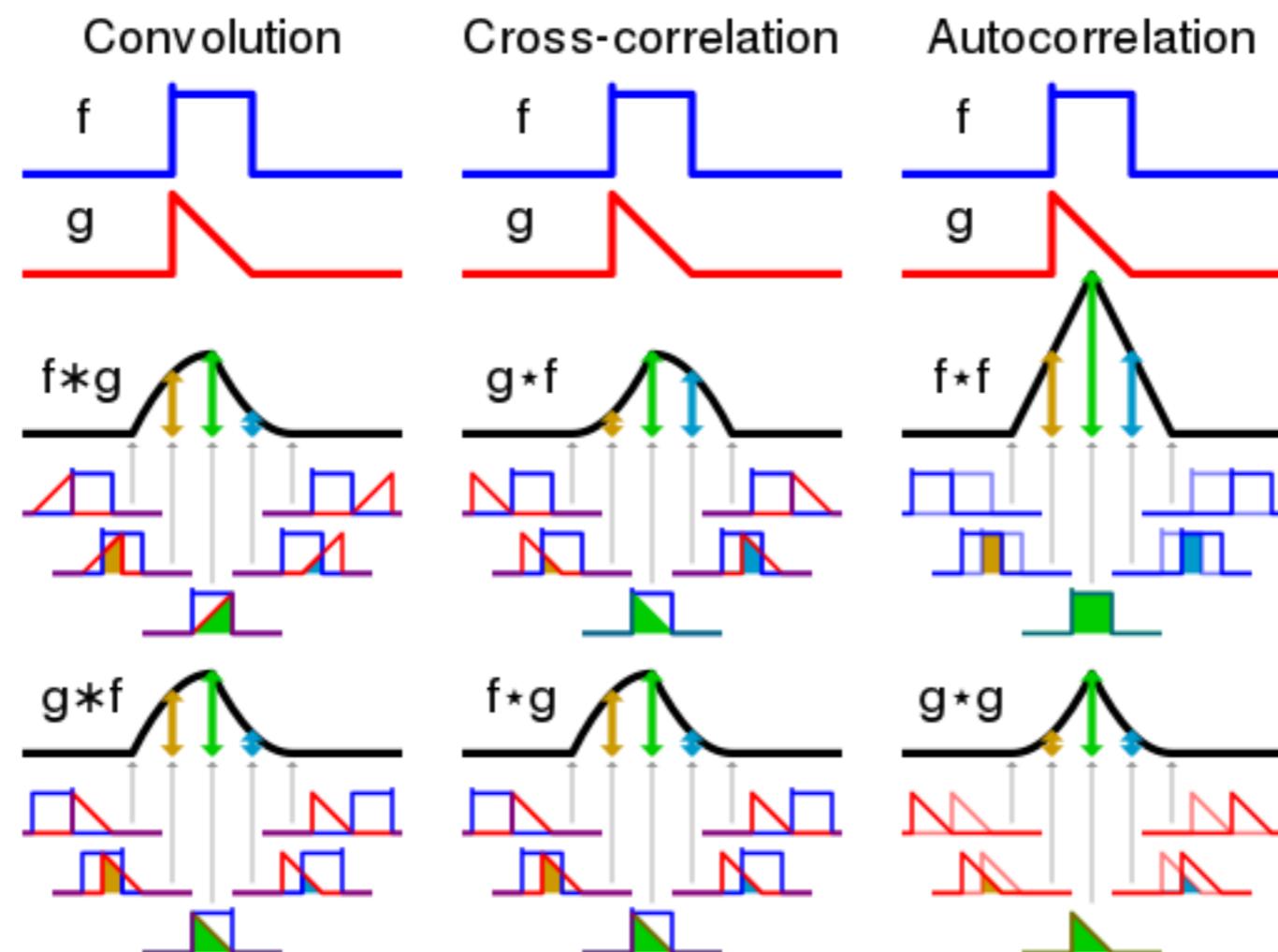
- * To “filter” an image, or “transform” an image often involves convolution operation

Matching Filter

- * Matched filter: To maximise SNR at receiver

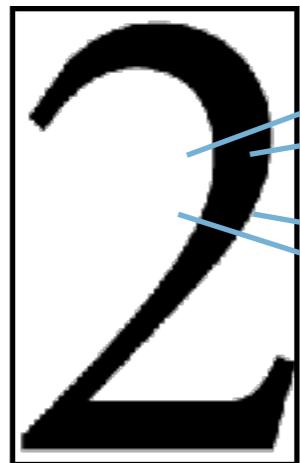


Convolution and Correlation



Computer “Vision”

Image



95	98	88	97	95	15	99	80	00	70	98	00	07	75	94	22	05	77	32	98	
98	98	88	93	12	91	18	97	90	97	97	40	45	50	05	35	20	45	23	94	45
52	70	95	29	89	59	11	99	49	56	96	98	56	75	97	01	02	06	98		
22	31	18	74	96	97	95	97	95	96	96	34	59	22	10	10	22	64	33	32	95
98	87	97	62	44	28	84	37	49	75	33	50	78	39	86	20	35	21	22	96	
93	99	81	29	88	29	87	10	24	36	40	47	60	54	70	46	28	38	44	70	
97	29	19	92	95	98	98	11	10	30	63	94	99	63	26	10	39	44	18	31	23
78	99	94	39	88	78	98	74	97	17	79	78	98	74	98	34	99	87	32		
31	86	28	59	75	59	78	81	20	48	35	14	50	45	39	97	54	35	33	95	
73	17	59	29	22	75	91	57	35	96	98	80	98	62	16	31	03	95	66	98	
18	99	29	81	98	99	95	91	87	98	98	88	24	00	37	94	24	24	25	88	97
19	98	93	83	84	71	88	37	10	44	37	46	93	25	11	33	94	21	38		
19	99	91	91	95	98	87	59	80	79	98	98	96	52	37	77	91	65	65	98	
28	98	95	95	97	95	99	19	97	97	97	98	20	20	20	78	32	21	32	96	
94	98	94	97	97	97	97	77	100	98	98	97	97	97	97	97	97	93	93	98	
38	82	18	79	21	51	93	11	34	94	72	18	98	46	29	33	42	43	74	94	
29	99	95	81	72	99	89	99	98	98	98	98	98	98	98	98	91	61	66	94	
32	79	95	29	73	91	92	91	79	35	49	73	95	94	93	34	22	27	62	94	
31	73	98	71	98	91	98	94	35	100	98	93	45	92	25	95	24	51	49		

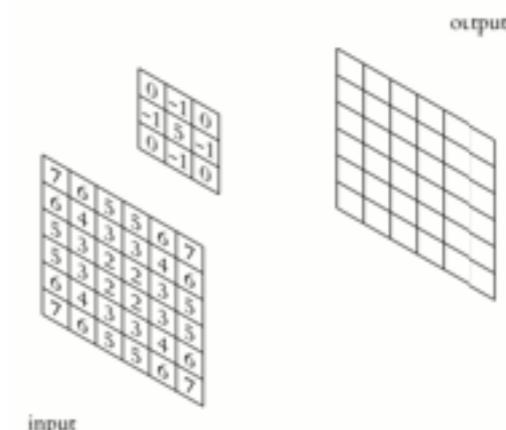
Computer sees a matrix of numbers

Classifying an image is to label the image

2

Understanding an image is to describe the image
numeral 2
in black on white
background

Processing an image is to mathematically operate on it



Why is it hard?

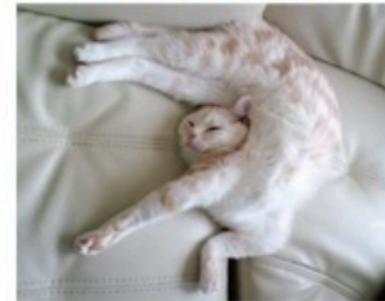
Viewpoint variation



Scale variation



Deformation



Occlusion



Illumination conditions



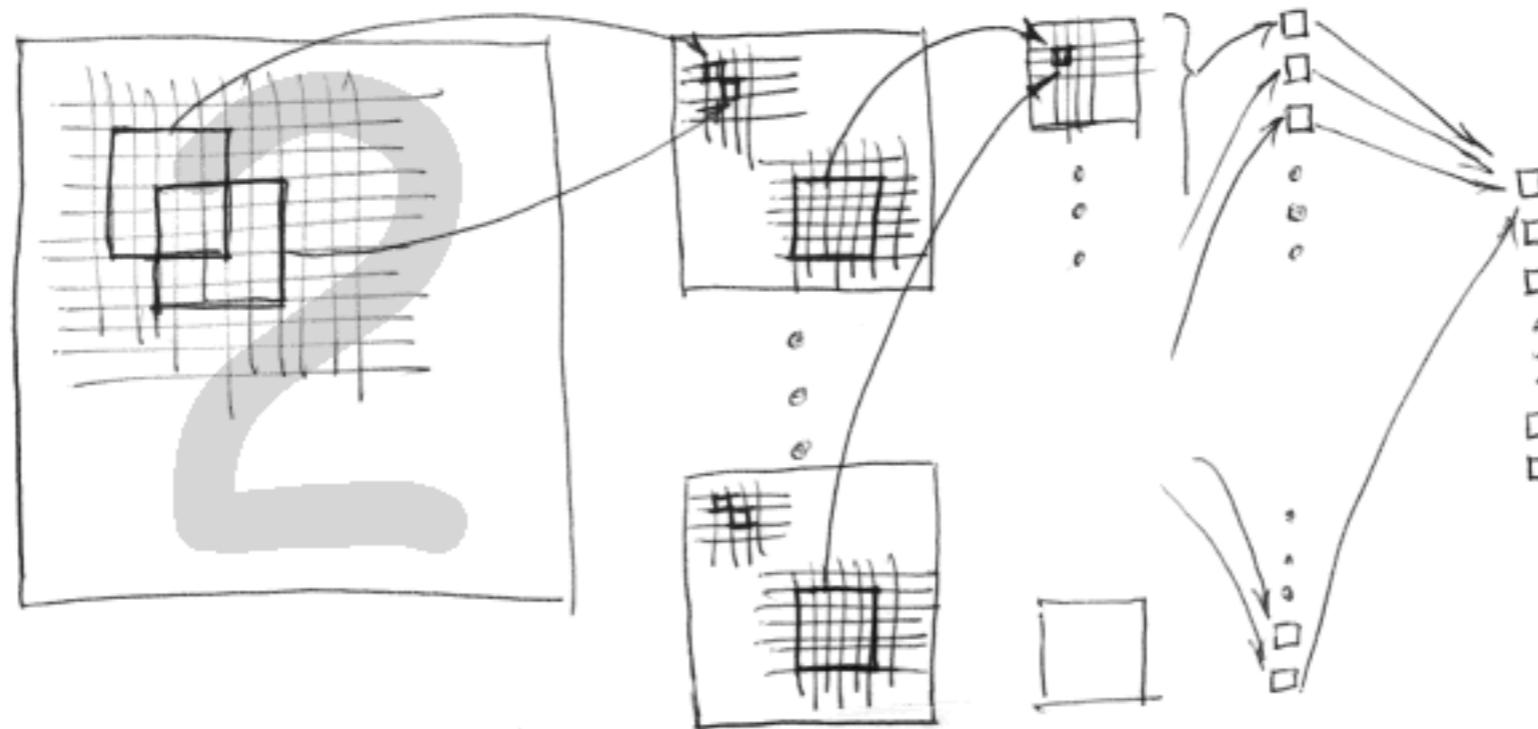
Background clutter



Intra-class variation

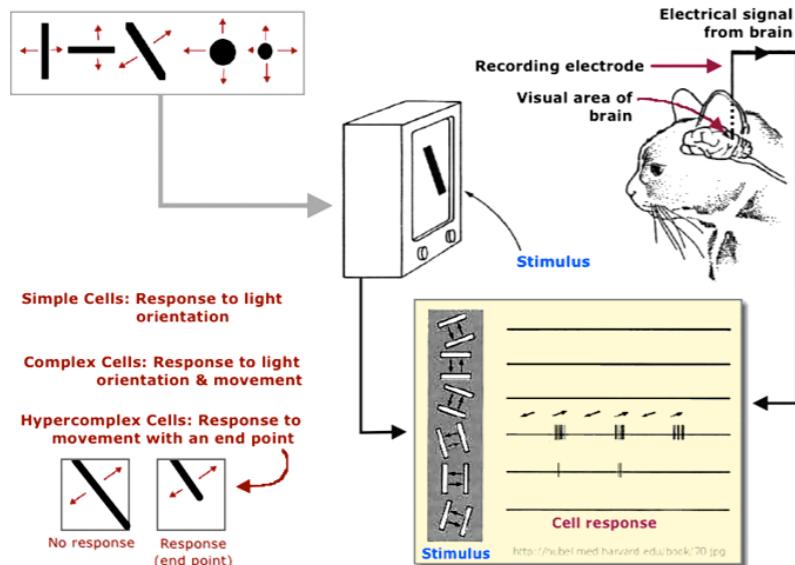


Convolutional Neural Networks



- * Layers of operations
- * Each operation is local — *Convolution*
- * Uses common learned parameters —
- * Each operation by a artificial neuron

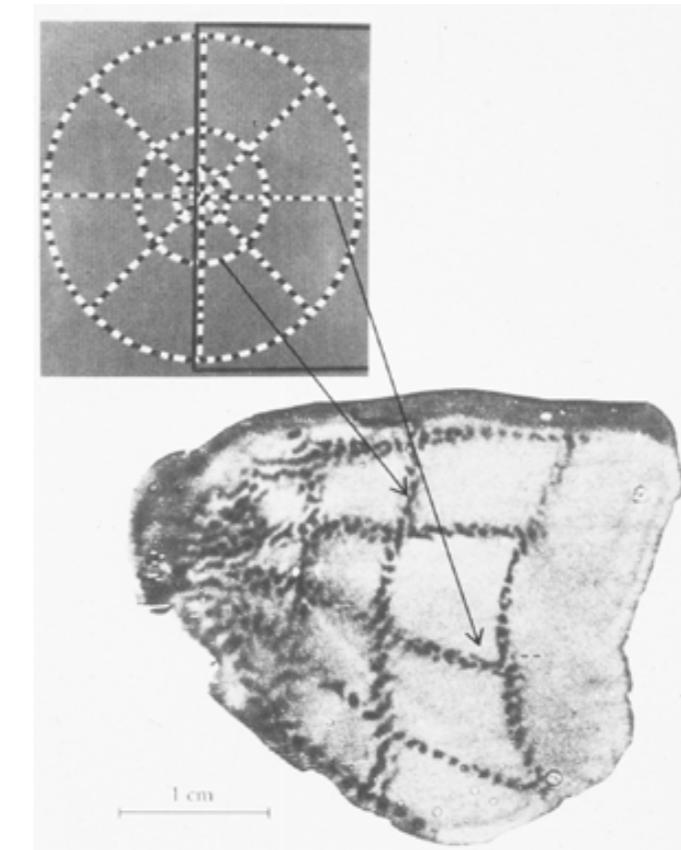
Why Convolution



Hubel & Wiesel, 1959

- * Receptive fields of visual system are local

- * Filters are oriented



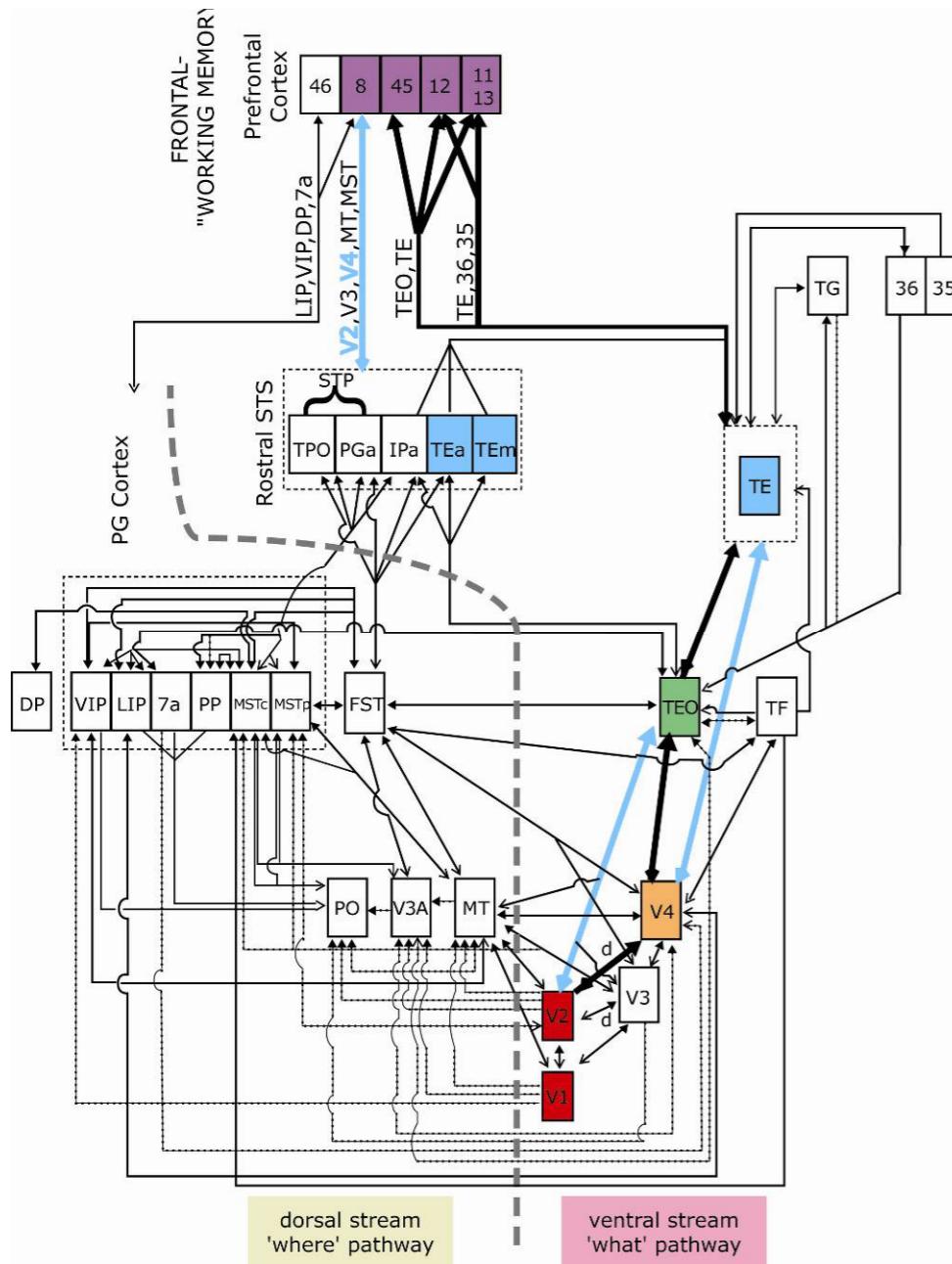
- * Topographic Mapping — nearby cells in retina processed by nearby regions in the brain

Hubel & Wiesel

- * Won Nobel Prize for this work

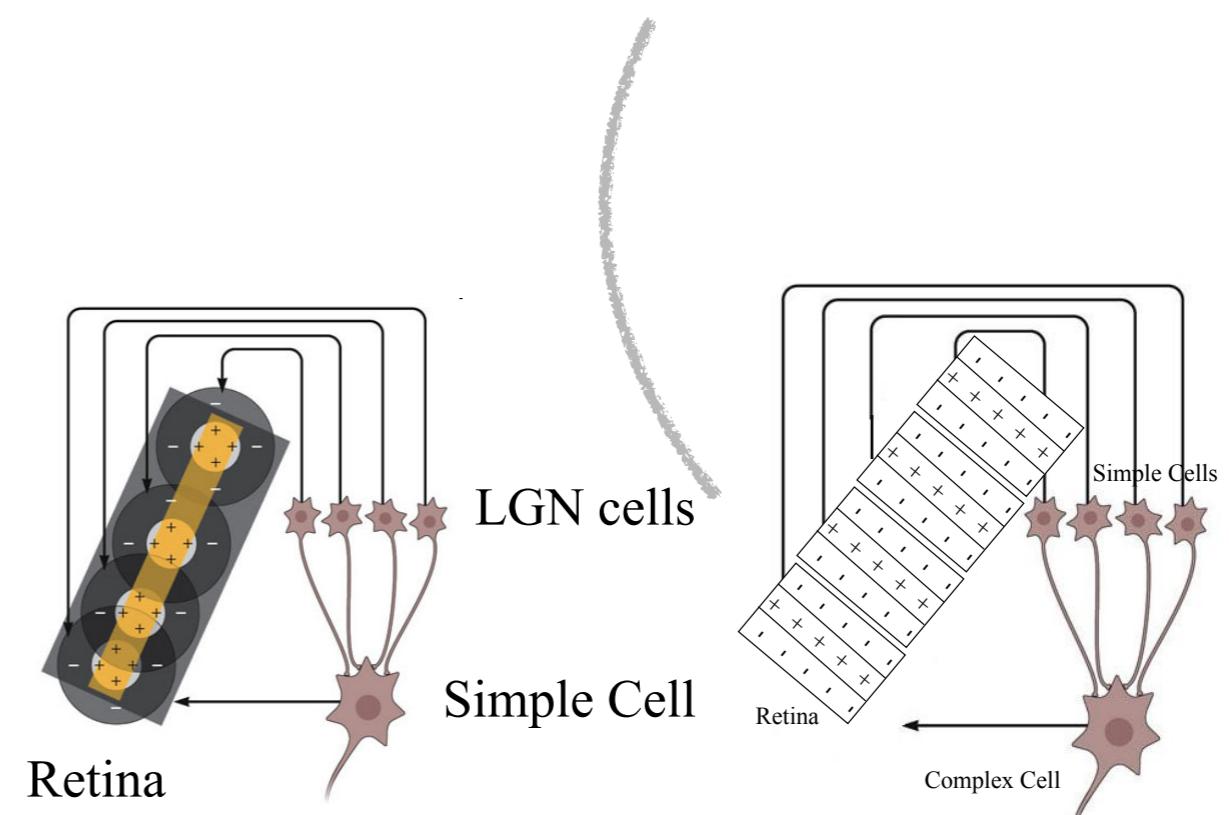


Why Layers



* Visual system has layers of processing

* Hierarchy of cells

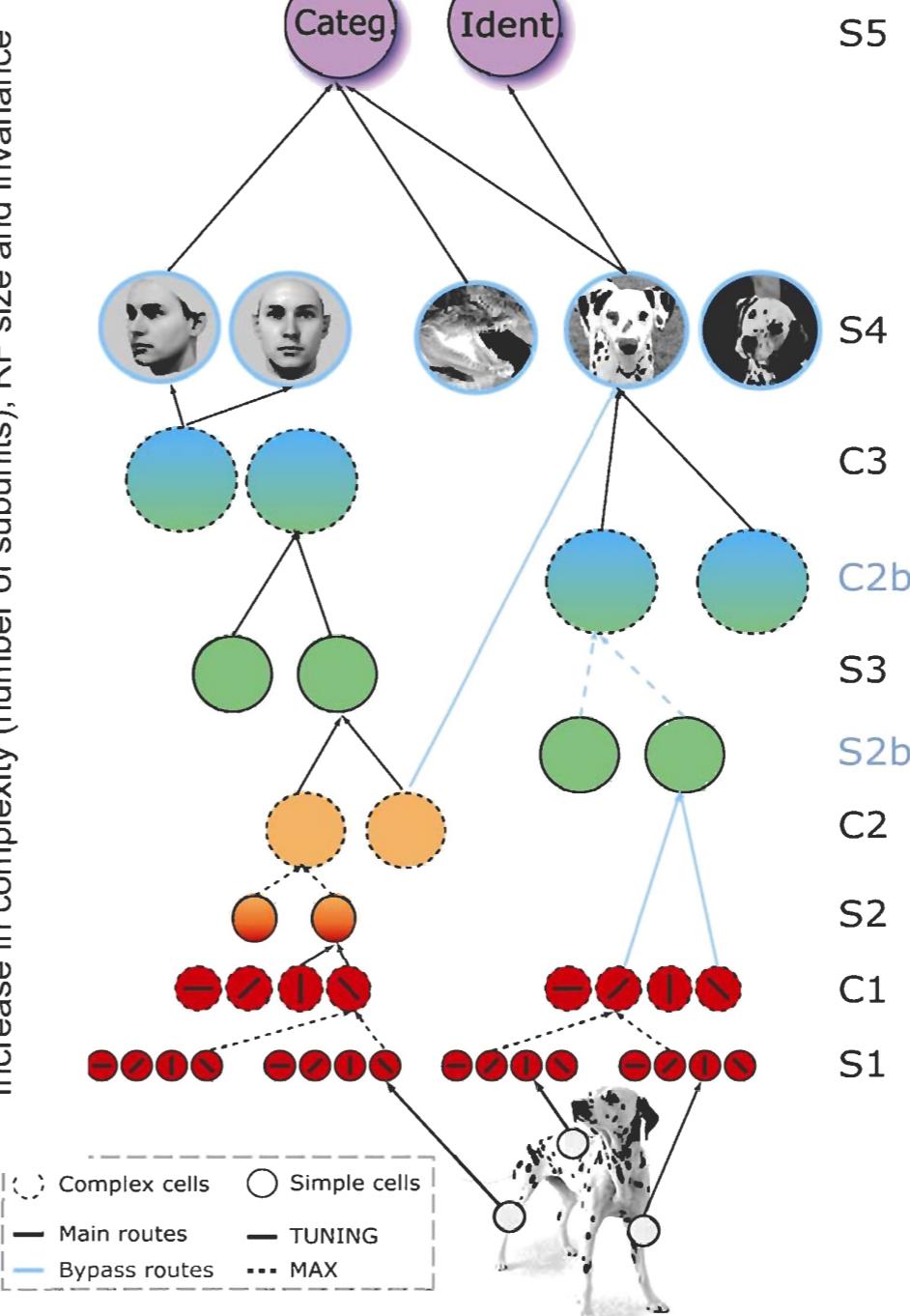


Modified from (Ungerleider & VanEssen)

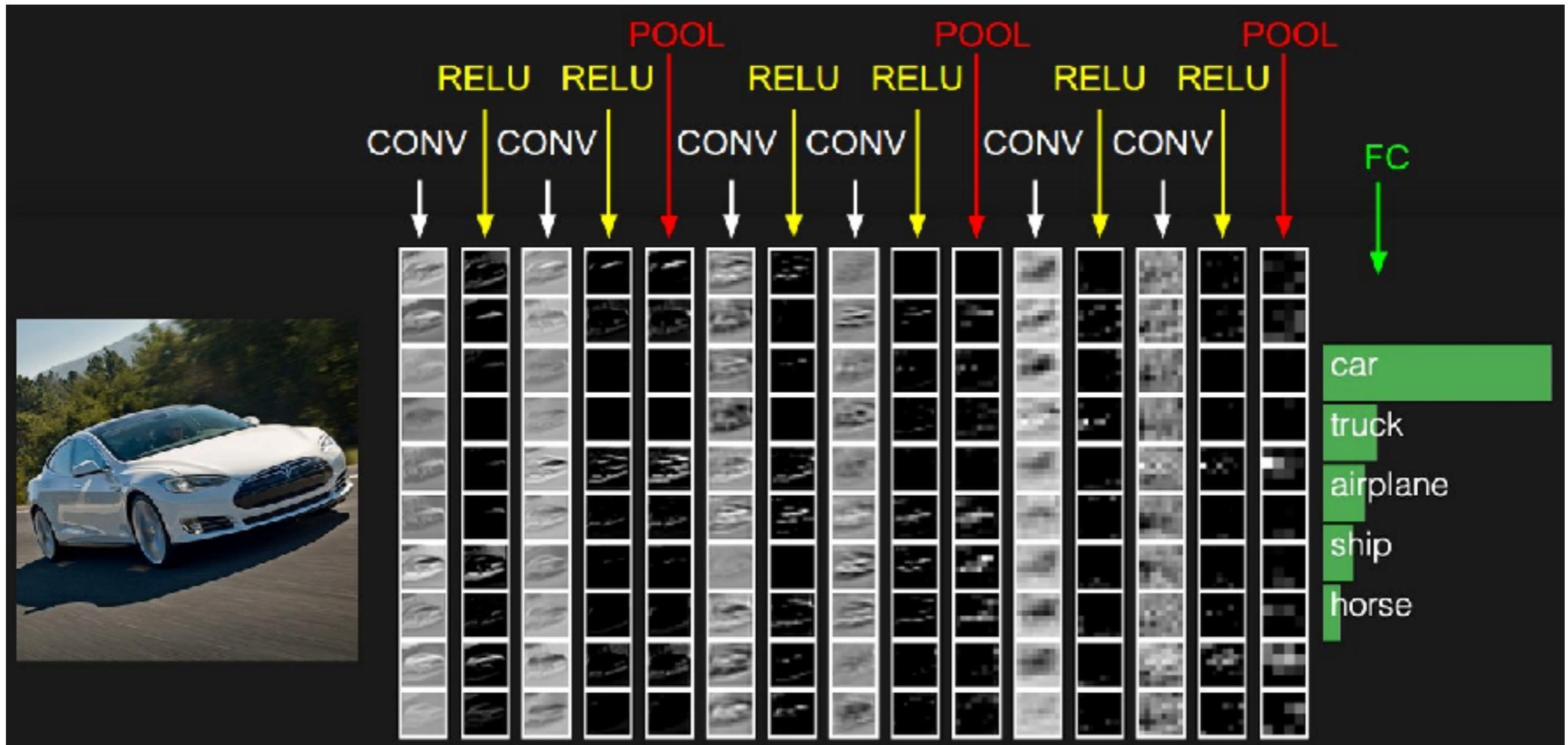
Visual Processing Model

Model layers	Corresponding brain area (tentative)	RF sizes	Number units
classifier	PFC	$1.0 \cdot 10^0$	
S4	AIT	 $>4.4^\circ$	$1.5 \cdot 10^2$ ~ 5,000 subunits
C3	PIT - AIT	 $>4.4^\circ$	$2.5 \cdot 10^3$
C2b	PIT	 $>4.4^\circ$	$2.5 \cdot 10^3$
S3	PIT	 $1.2^\circ - 3.2^\circ$	$7.4 \cdot 10^4$ ~ 100 subunits
S2b	V4 - PIT	 $0.9^\circ - 4.4^\circ$	$1.0 \cdot 10^7$ ~ 100 subunits
C2	V4	 $1.1^\circ - 3.0^\circ$	$2.8 \cdot 10^5$
S2	V2 - V4	 $0.6^\circ - 2.4^\circ$	$1.0 \cdot 10^7$ ~ 10 subunits
C1	V1 - V2	 $0.4^\circ - 1.6^\circ$	$1.2 \cdot 10^4$
S1	V1 - V2	 $0.2^\circ - 1.1^\circ$	$1.6 \cdot 10^6$

↑ Supervised task-dependent learning
↓ Unsupervised task-independent learning
Increase in complexity (number of subunits), RF size and invariance



Convolutional NN



- * Parameters of CONV layers are learned from data

Emergence of simple cell properties

- * Batch of images each with its labels (car, truck, airplane...)
- * Loss how well does predicted label match actual label
- * Minimize loss
- * Emergence of simple cell properties
 - * Hubel and Weisel showed that cat responded to oriented bars
 - * Learned visual fields of CNN are invariably oriented filters

