Representation Learning and Deep Learning

Fabio A. González
Univ. Nacional de Colombia

Some history

https://www.youtube.com/watch?v=cNxadbrN_aI

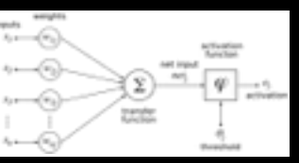
Rosenblatt's Perceptron (1957)

- Input: 20x20 photocells array
- Weights implemented with potentiometers
- Weight updating performed by electric motors

Neural networks time line

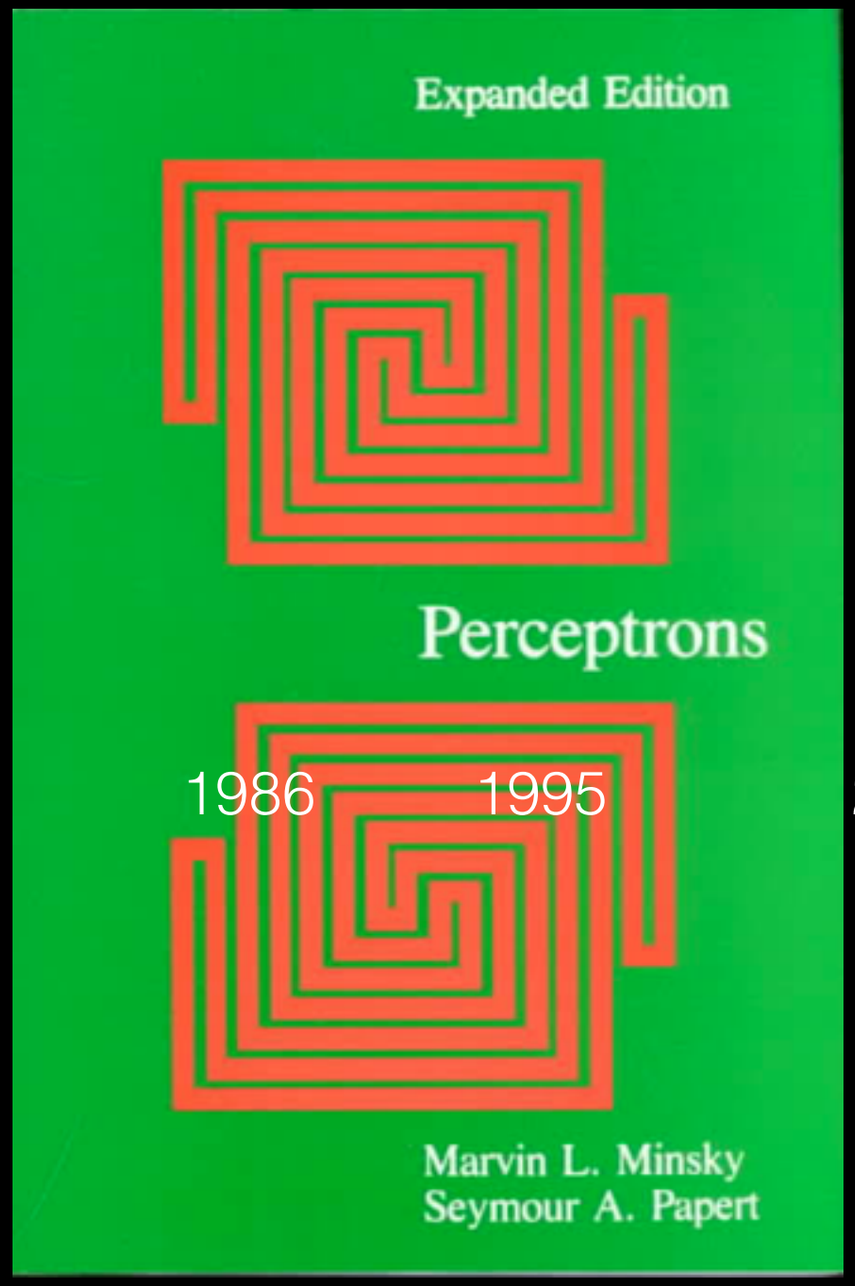
1943 1957 1969 1986 1995 2007 2012 2016

Neural networks time line



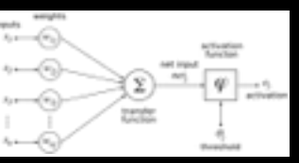
1943 1957 1969 1986 1995 2007 2012 2016

Neural networks time line

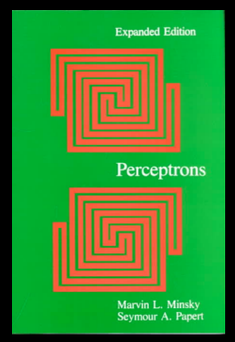


1943 1957 1969 1986 1995 2007 2012 2016

Neural networks time line

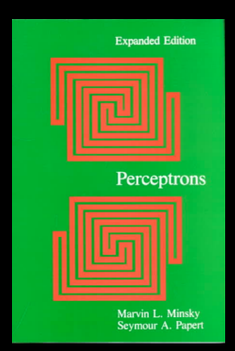


1943 1957 1969 1986 1995 2007 2012 2016

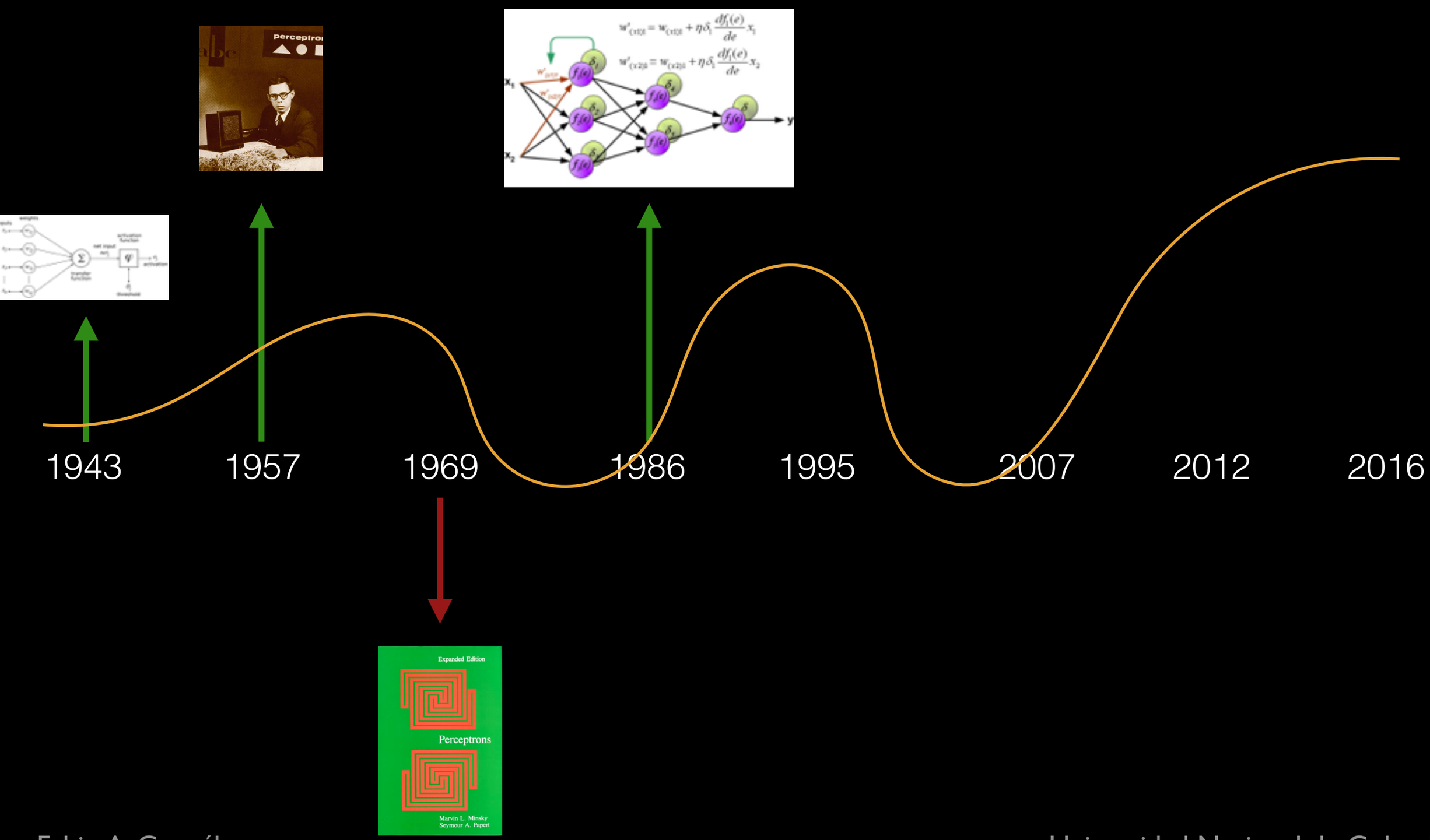


Neural networks time line

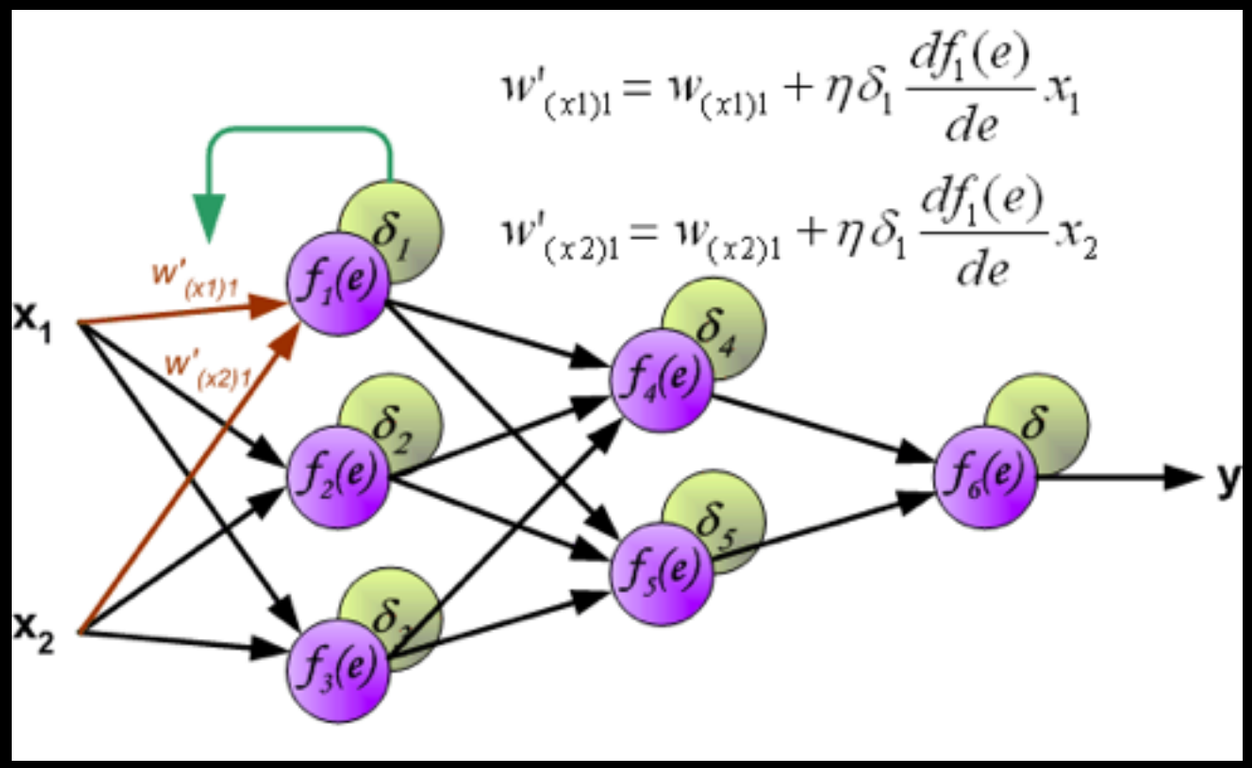
1943 1957 1969 1986 1995 2007 2012 2016



Neural networks time line

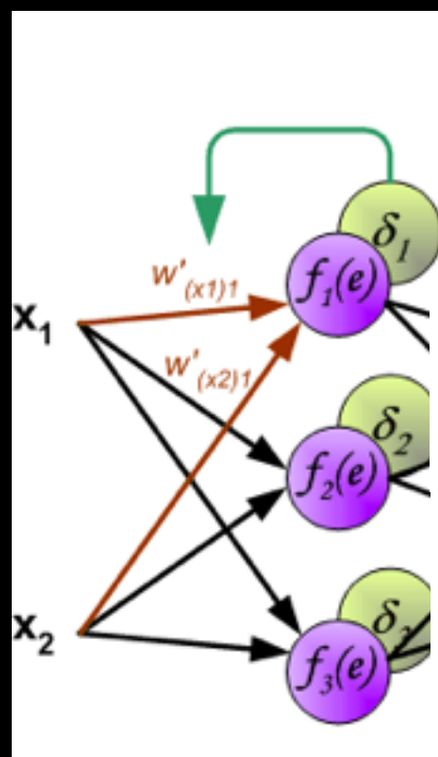


Backpropagation



Source: http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html

Backpropagation



$$w'_{(x1)1} = w_{(x1)1} + \eta \delta_1 \frac{df_1(e)}{de} x_1$$

$$w'_{(x2)1} = w_{(x2)1} + \eta \delta_1 \frac{df_1(e)}{de} x_2$$

letters to nature

nature

Nature **323**, 533 - 536 (09 October 1986); doi:10.1038/323533a0

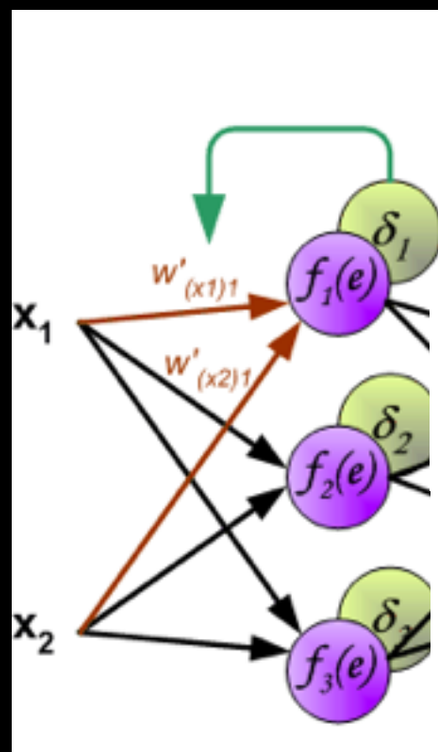
Learning representations by back-propagating errors

David E. Rumelhart*, Geoffrey E. Hinton† & Ronald J. Williams*

* Institute for Cognitive Science, C-015, University of California, San Diego, La Jolla, California 92093, USA
† Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Philadelphia 15213, USA

Source: http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html

Backpropagation



$$w'_{(x1)1} = w_{(x1)1} + \eta \delta_1 \frac{df_1(e)}{de} x_1$$

$$w'_{(x2)1} = w_{(x2)1} + \eta \delta_1 \frac{df_1(e)}{de} x_2$$

letters to nature

nature

Nature **323**, 533 - 536 (09 October 1986); doi:10.1038/323533a0

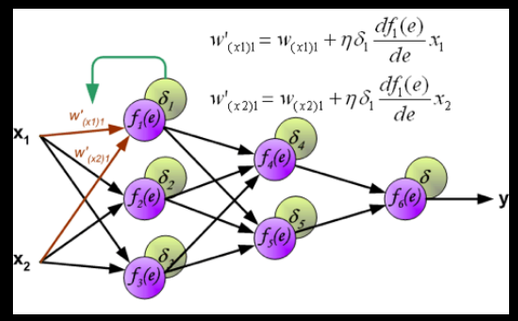
Learning representations by back-propagating errors

David E. Rumelhart*, Geoffrey E. Hinton† & Ronald J. Williams*

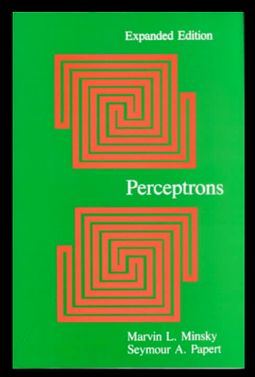
* Institute for Cognitive Science, C-015, University of Californi
San Diego, La Jolla, California 92093, USA
† Department of Computer Science, Carnegie-Mellon Universit
Pittsburgh, Philadelphia 15213, USA

Source: http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html

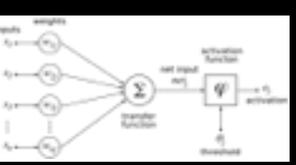
Neural networks time line

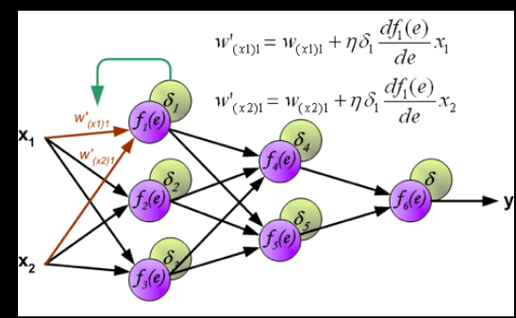


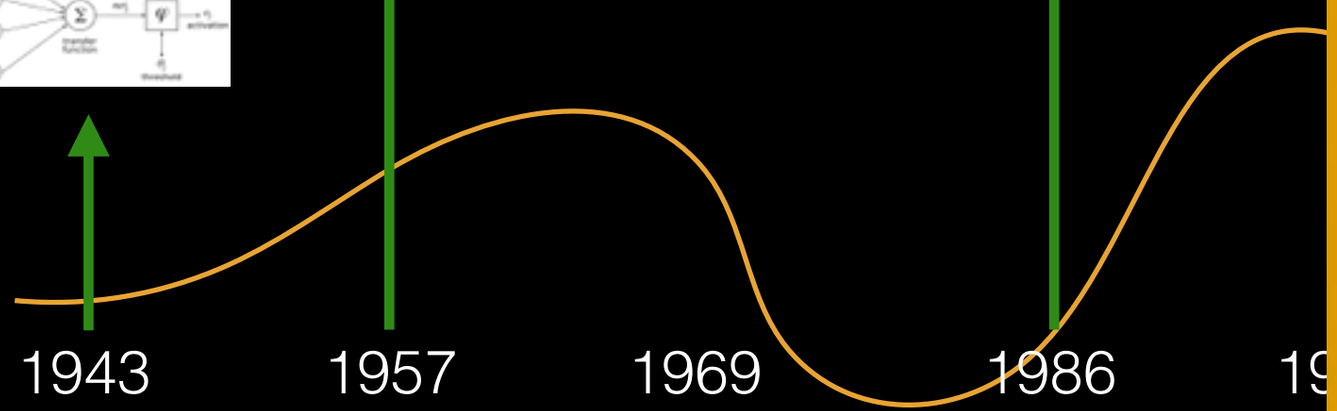
1943 1957 1969 1986 1995 2007 2012 2016

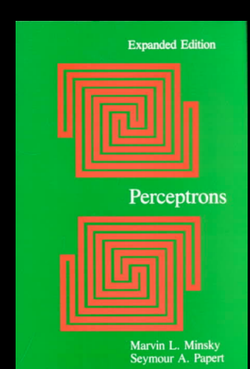


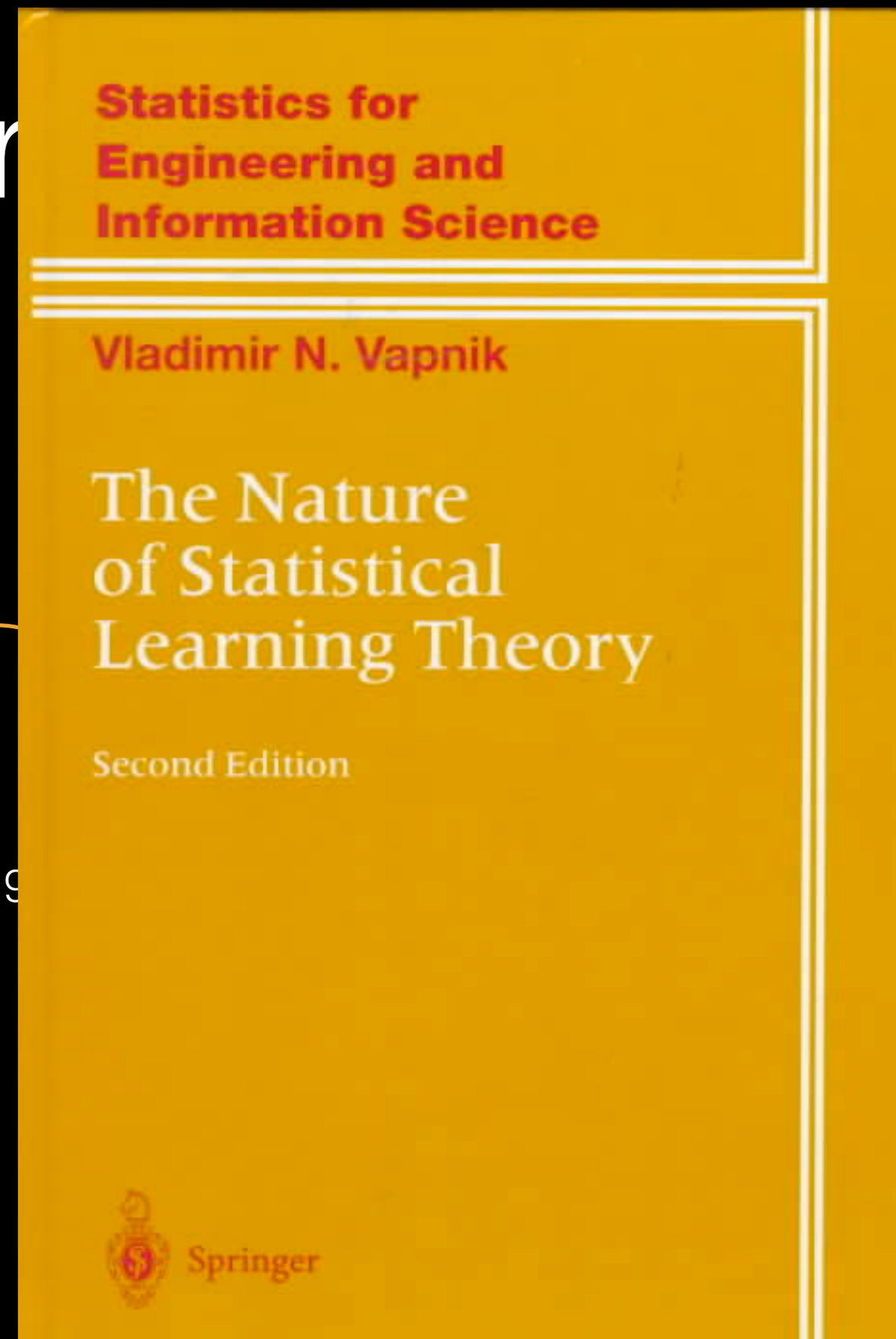
Neural networ



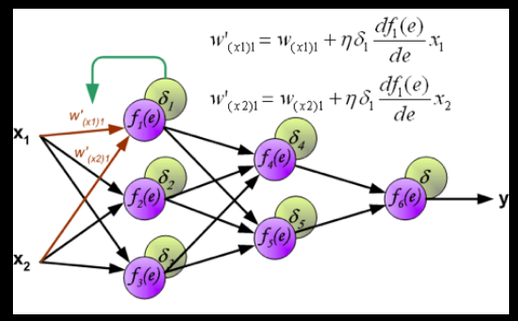




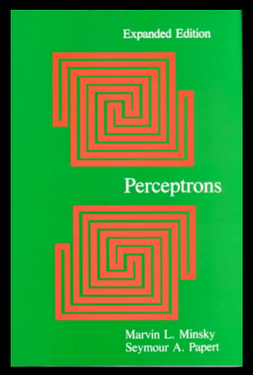


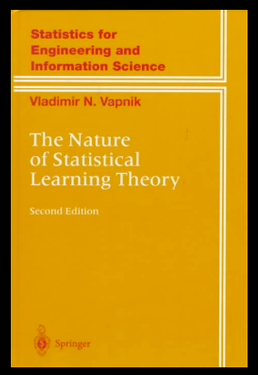


Neural networks time line

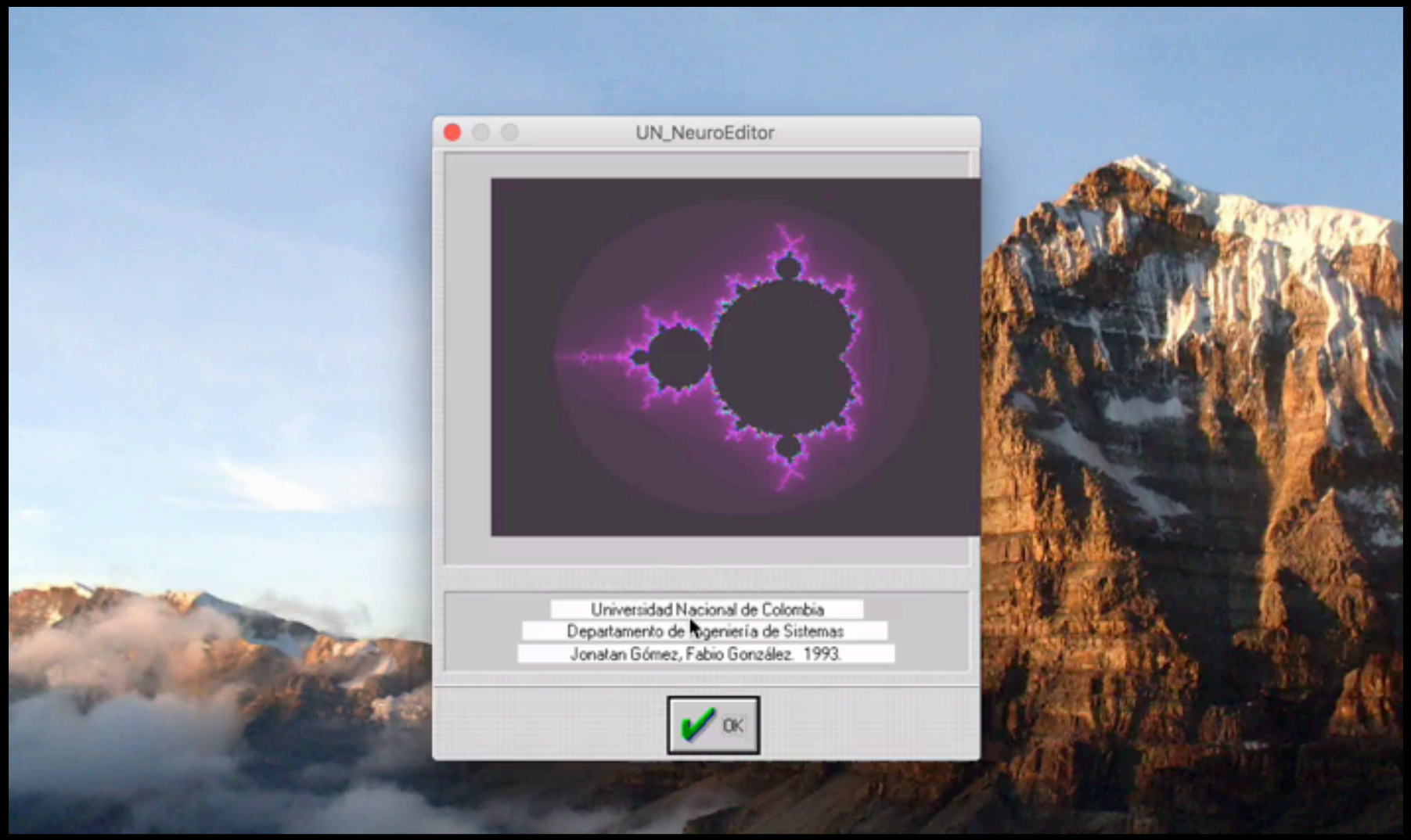


1943 1957 1969 1986 1995 2007 2012 2016



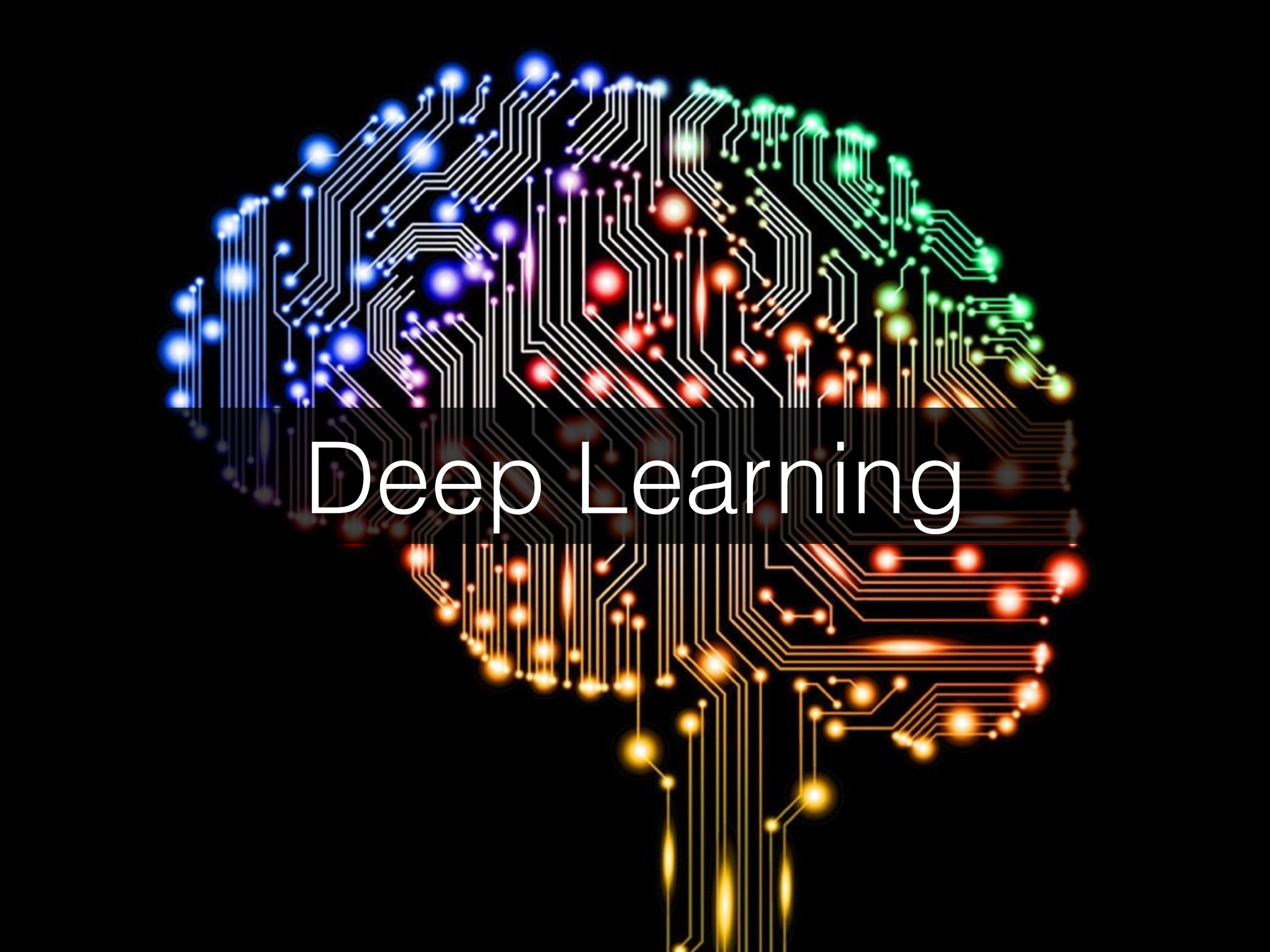


My own history with NN (circa 1993)



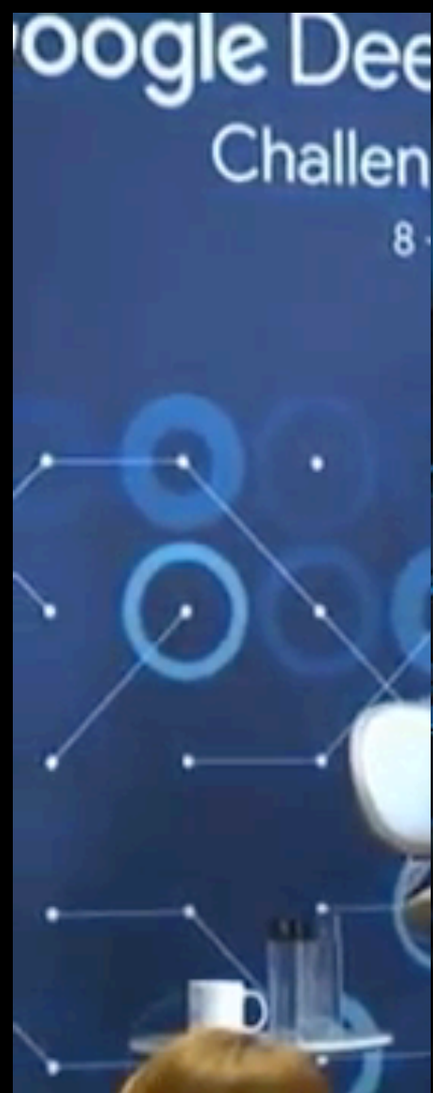
Quick and Dirty Introduction to Keras

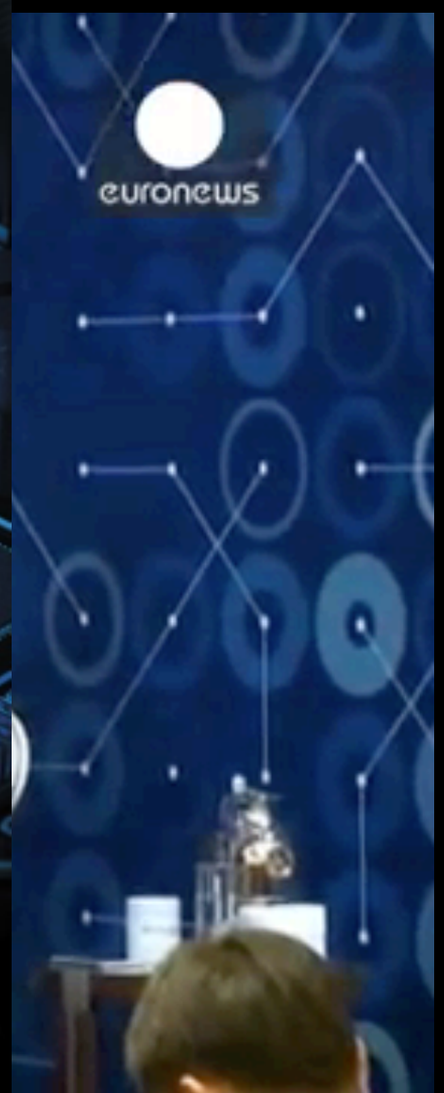
Interactive Demo


Deep Learning

Deep learning boom

Deep learning boom

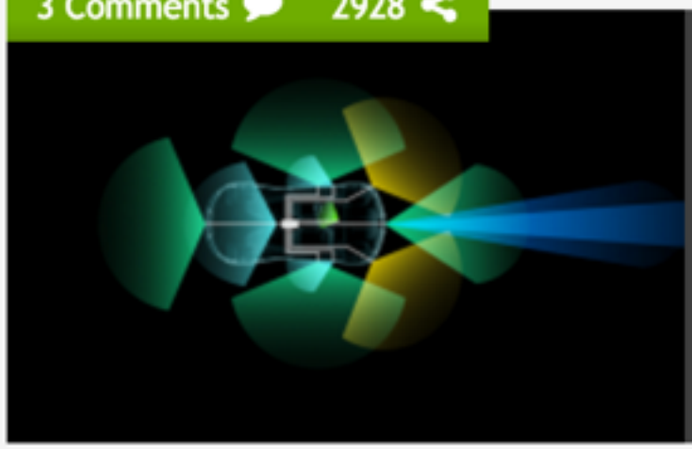




Deep learning boom

Deep learning boom

3 Comments 2928



DRIVING

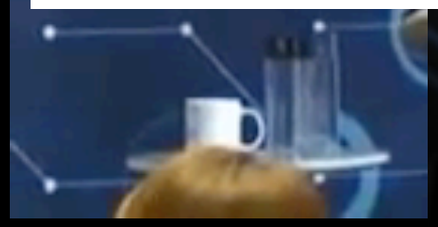
Here's How Deep Learning Will Accelerate Self-Driving Cars

By Danny Shapiro on February 24, 2015

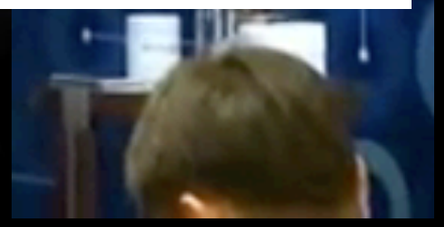
nature16961

Mastering the game of Go with deep neural networks and tree search

David Silver[1]*, Aja Huang[1]*, Chris J. Maddison[1], Arthur Guez[1], Laurent Sifre[1], George van den Driessche[1], Julian Schrittwieser[1], Ioannis Antonoglou[1], Veda Panneershelvam[1], Marc Lanctot[1], Sander Dieleman[1], Dominik Grewe[1], John Nham[2], Nal Kalchbrenner[1], Ilya Sutskever[2], Timothy Lillicrap[1], Madeleine Leach[1], Koray Kavukcuoglu[1], Thore Graepel[1] & Demis Hassabis[1]



ALL SYSTEMS GO



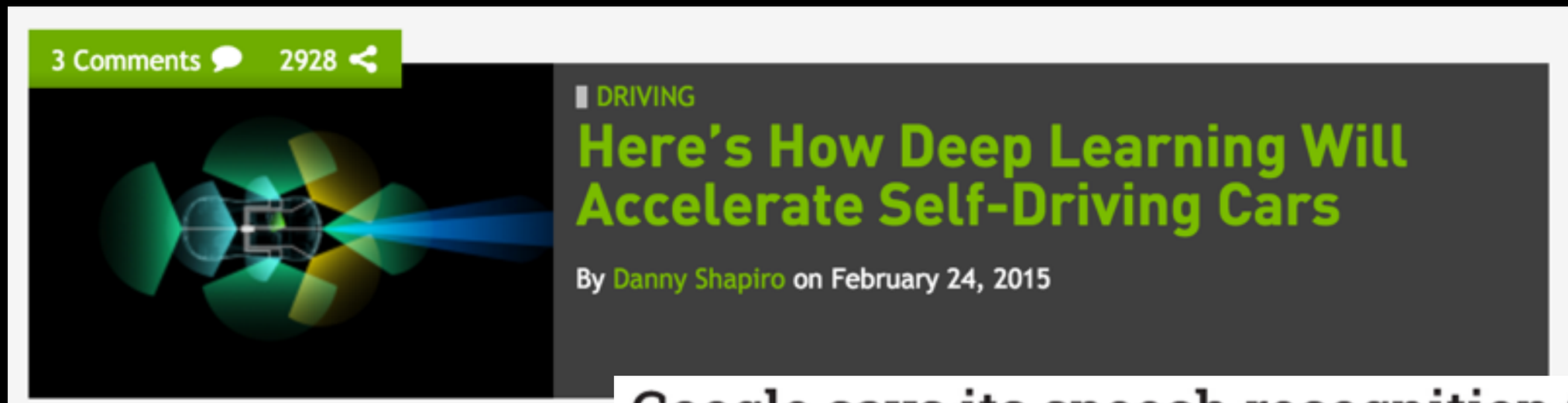
CONSERVATION
SONGBIRDS À LA CARTE
Illegal harvest of millions of Mediterranean birds
PAGE 452

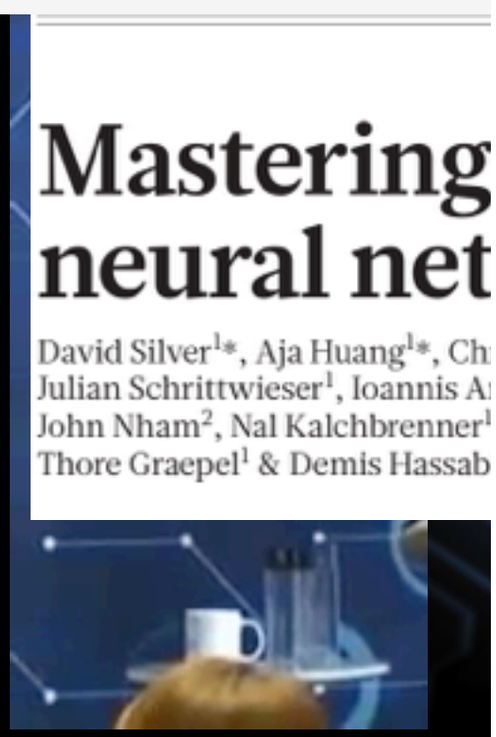
RESEARCH ETHICS
SAFEGUARD TRANSPARENCY
Don't let openness backfire on individuals
PAGE 459

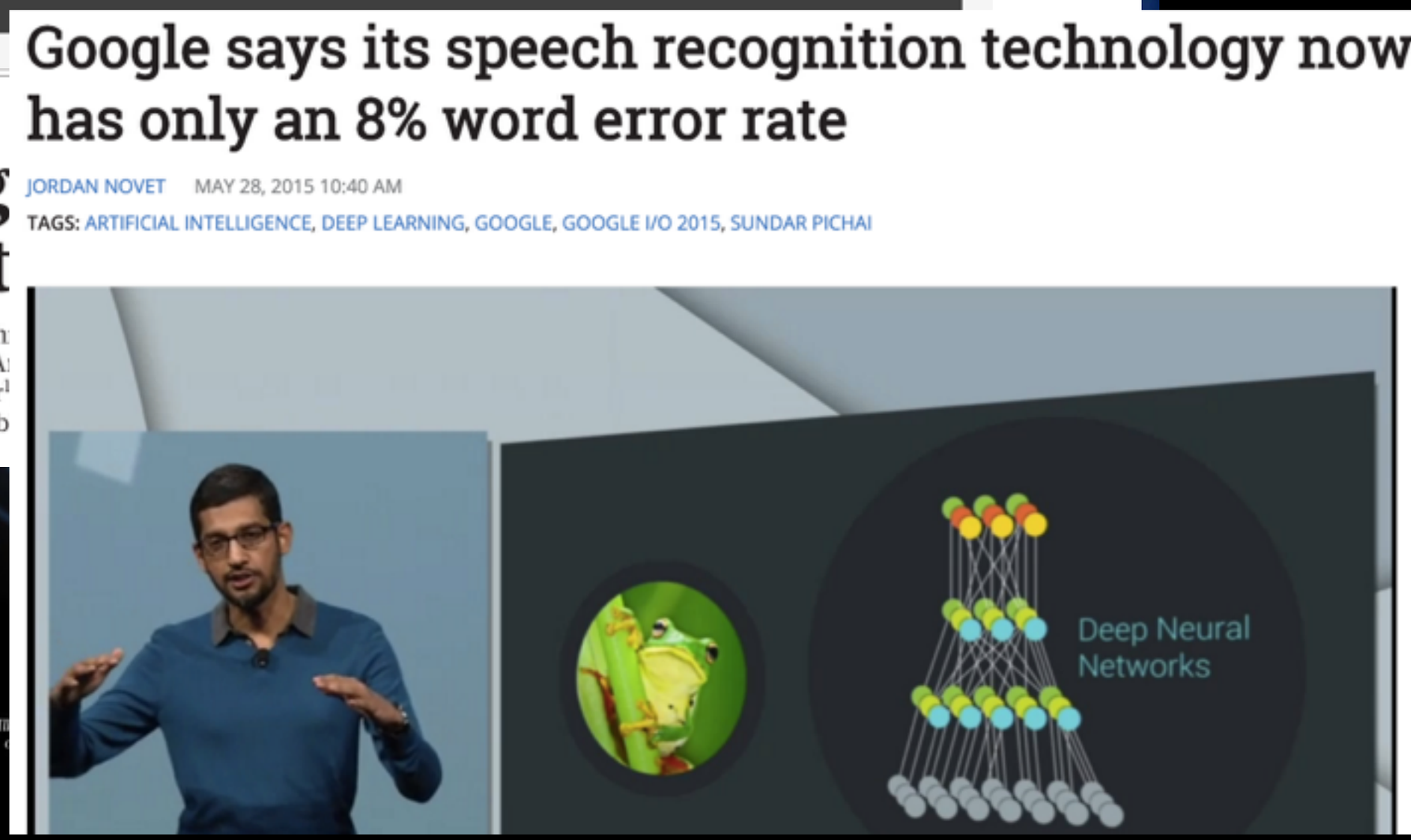
POPULAR SCIENCE
WHEN GENES GOT 'SELFISH'
Dawkins's calling card forty years on
PAGE 462

NATURE.COM/NATURE
28 January 2016 £10
Vol. 529, No. 7587

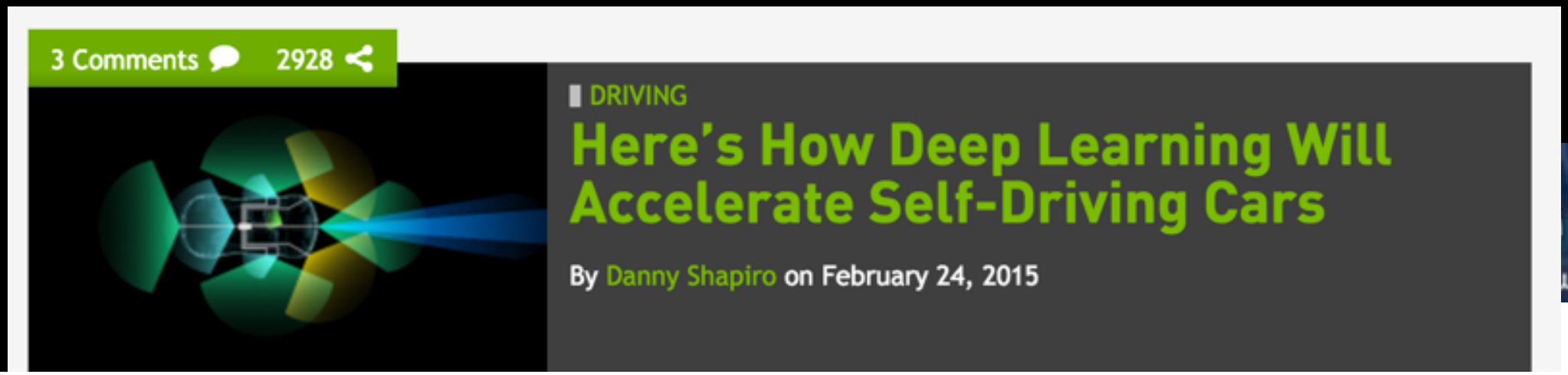
Deep learning boom







Deep learning boom



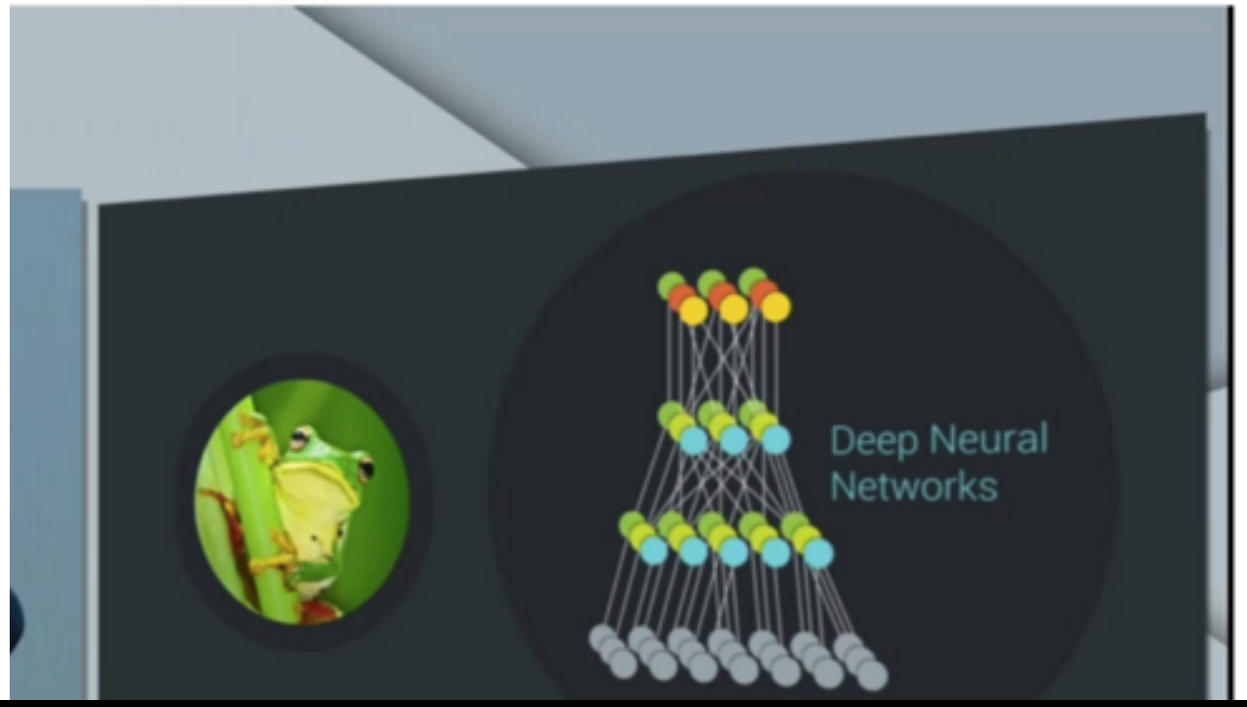
ROBERT MCMILLAN BUSINESS 03.13.13 6:30 AM

GOOGLE HIRES BRAINS THAT HELPED SUPERCHARGE MACHINE LEARNING

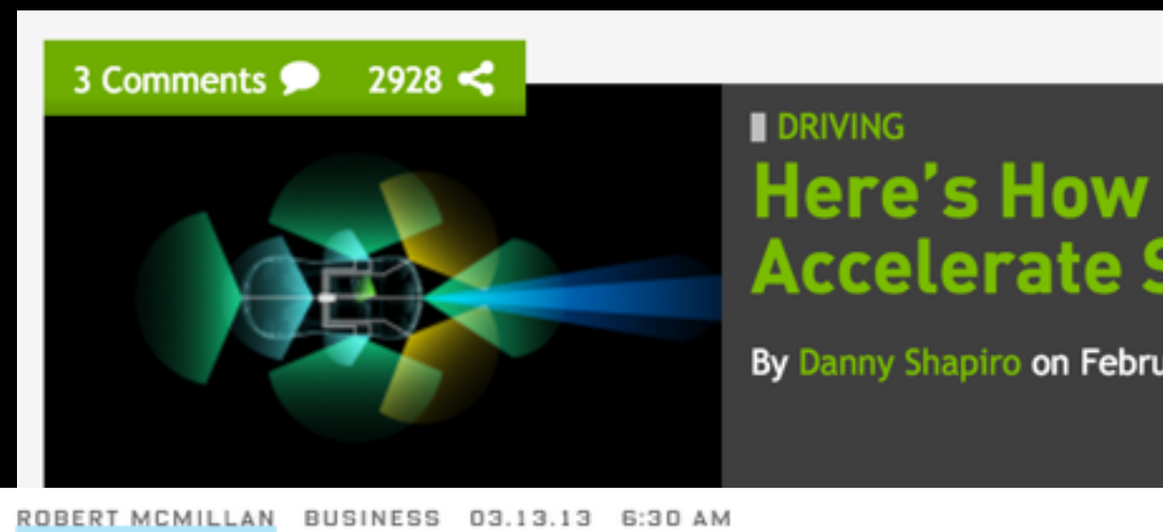
speech recognition technology now

word error rate

ING, GOOGLE, GOOGLE I/O 2015, SUNDAR PICHAI



Deep learning boom



ROBERT MCMILLAN BUSINESS 03.13.13 6:30 AM

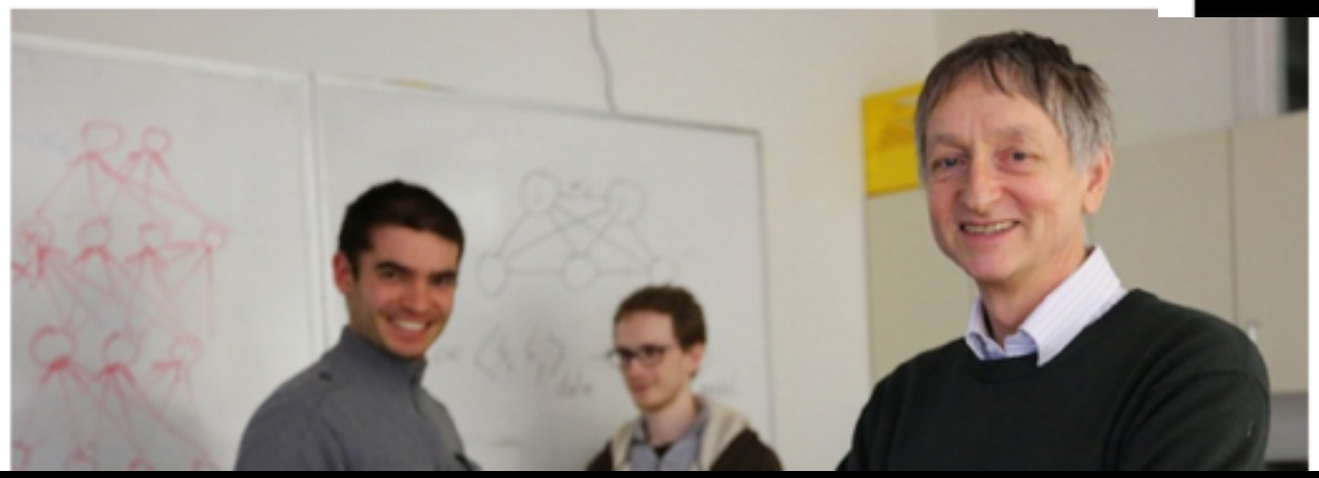
GOOGLE HIRES BRAINS THAT HELPED SUPERCHARGE MACHINE LEARNING

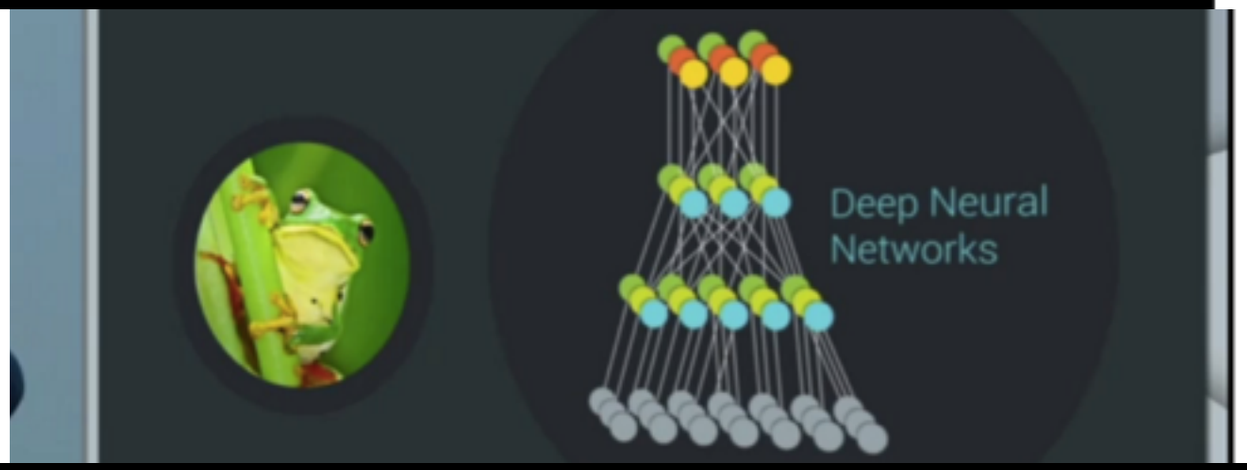
MIT Technology Review

Baidu 百度

Computing

Chinese Search Giant Baidu Hires Man Behind the "Google Brain"





Deep learning boom

DRIVING
Here's How

MIT Technology Review

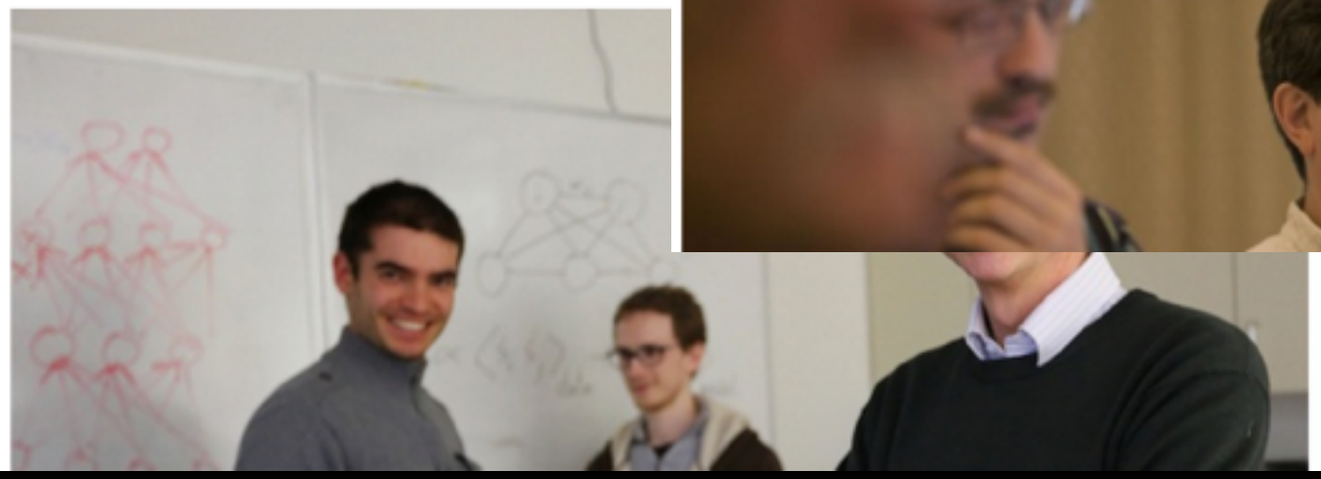
CADE METZ BUSINESS 12.09.13 3:14 PM

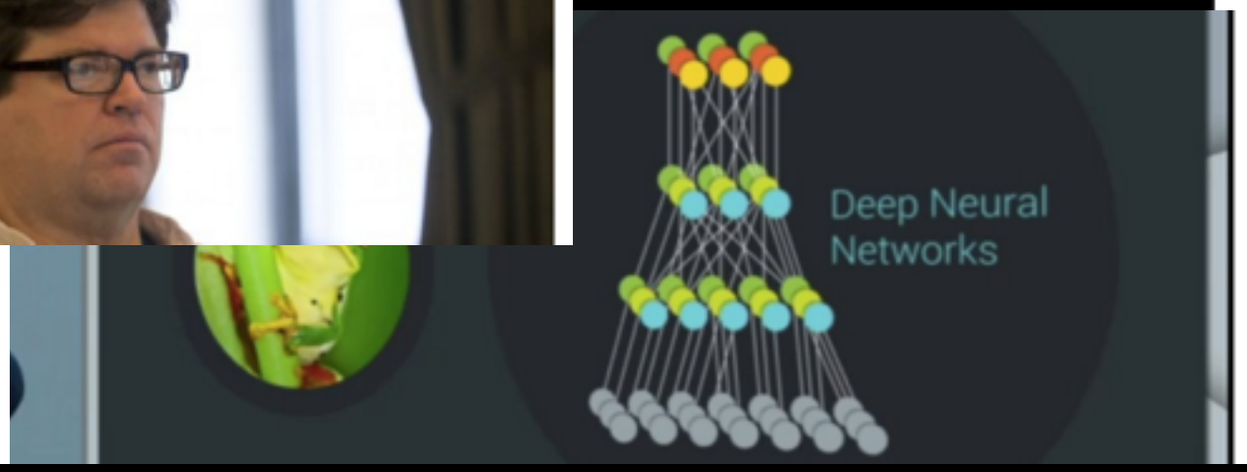
FACEBOOK TAPS 'DEEP LEARNING' GIANT FOR NEW AI LAB

ROBERT MCMILLAN BUSINESS 03.13.13 6:30

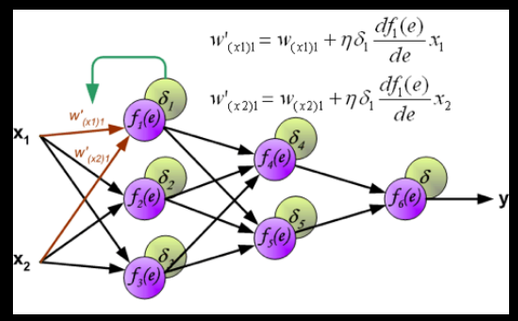
GOOGLE HIRES B
HELPED SUPERC
MACHINE LEARN

rch Giant
Man Behind
Brain"

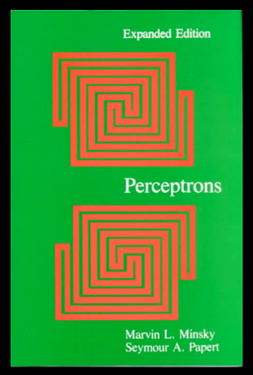


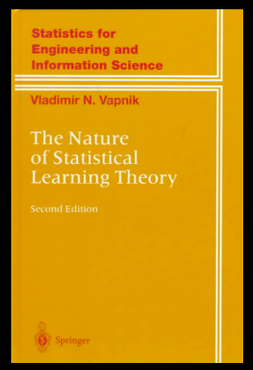


Deep learning time line

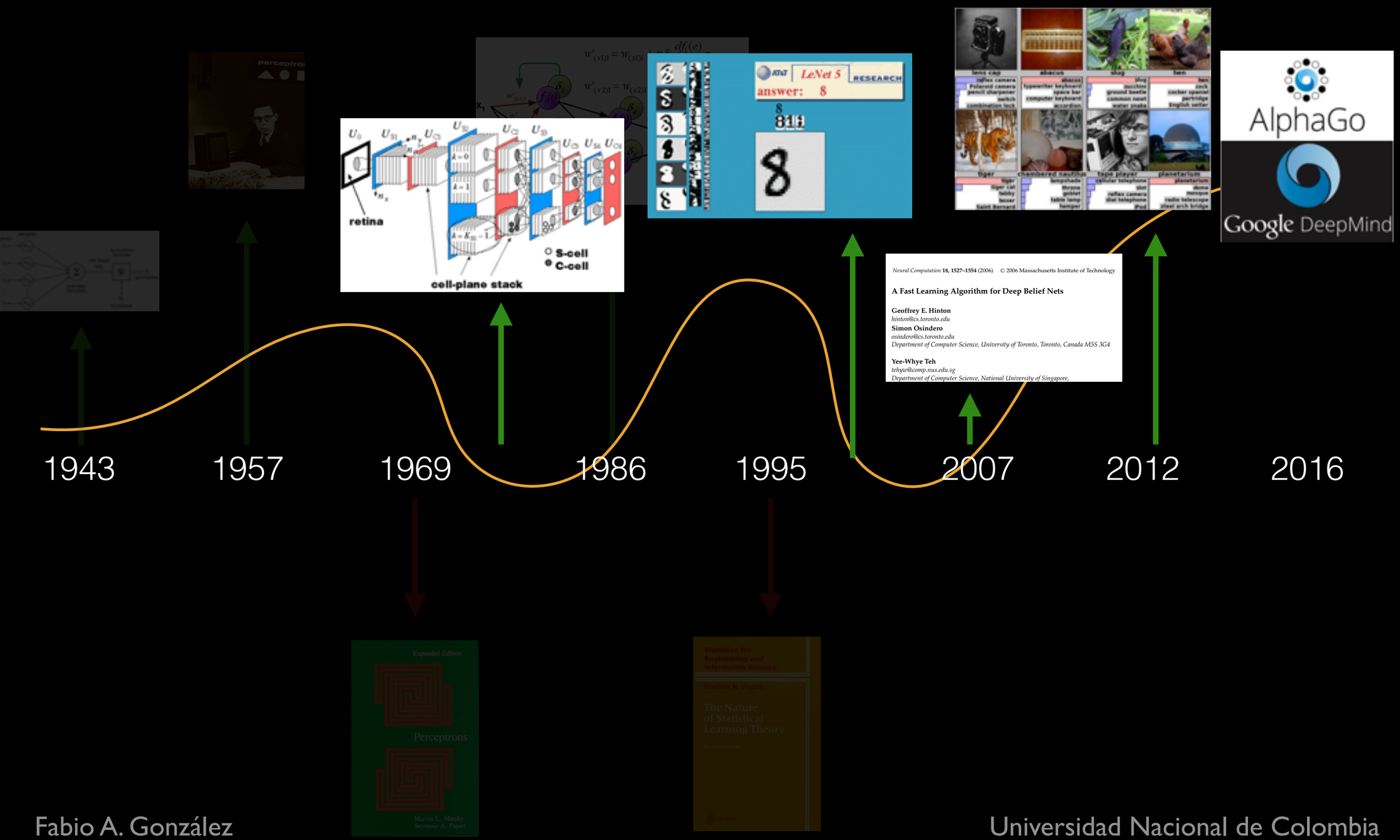


1943 1957 1969 1986 1995 2007 2012 2016



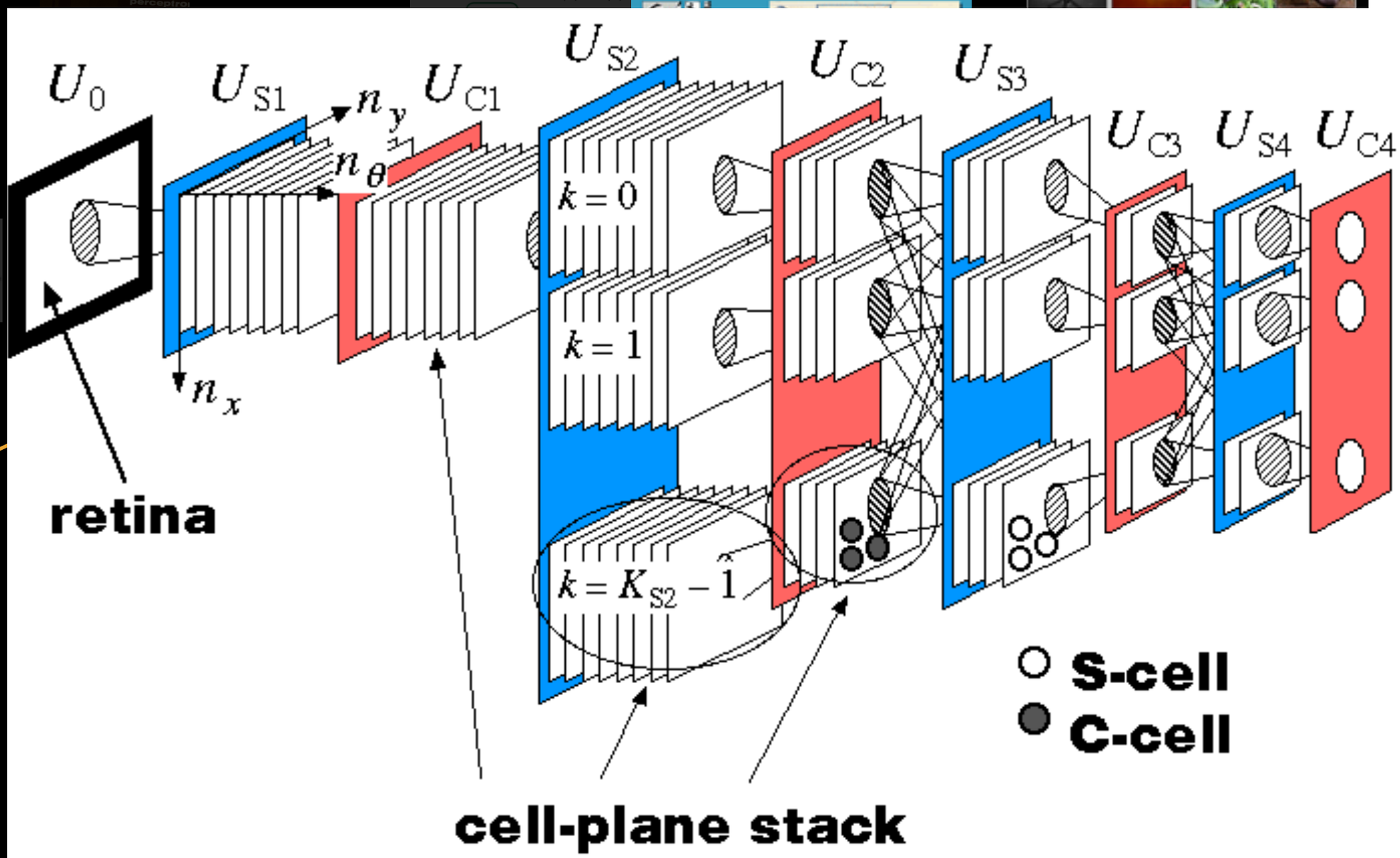


Deep learning time line



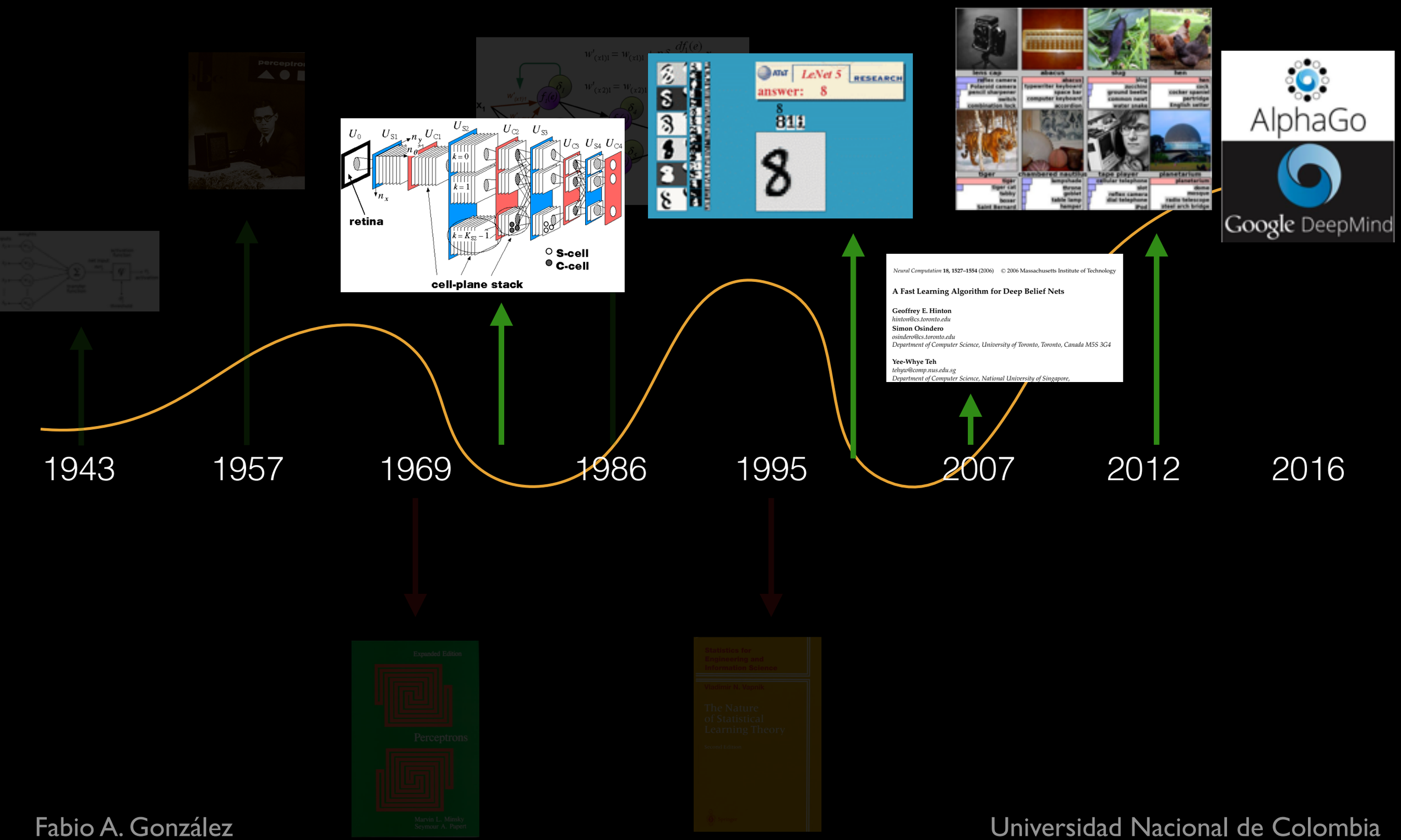
Deep learning time line

1943



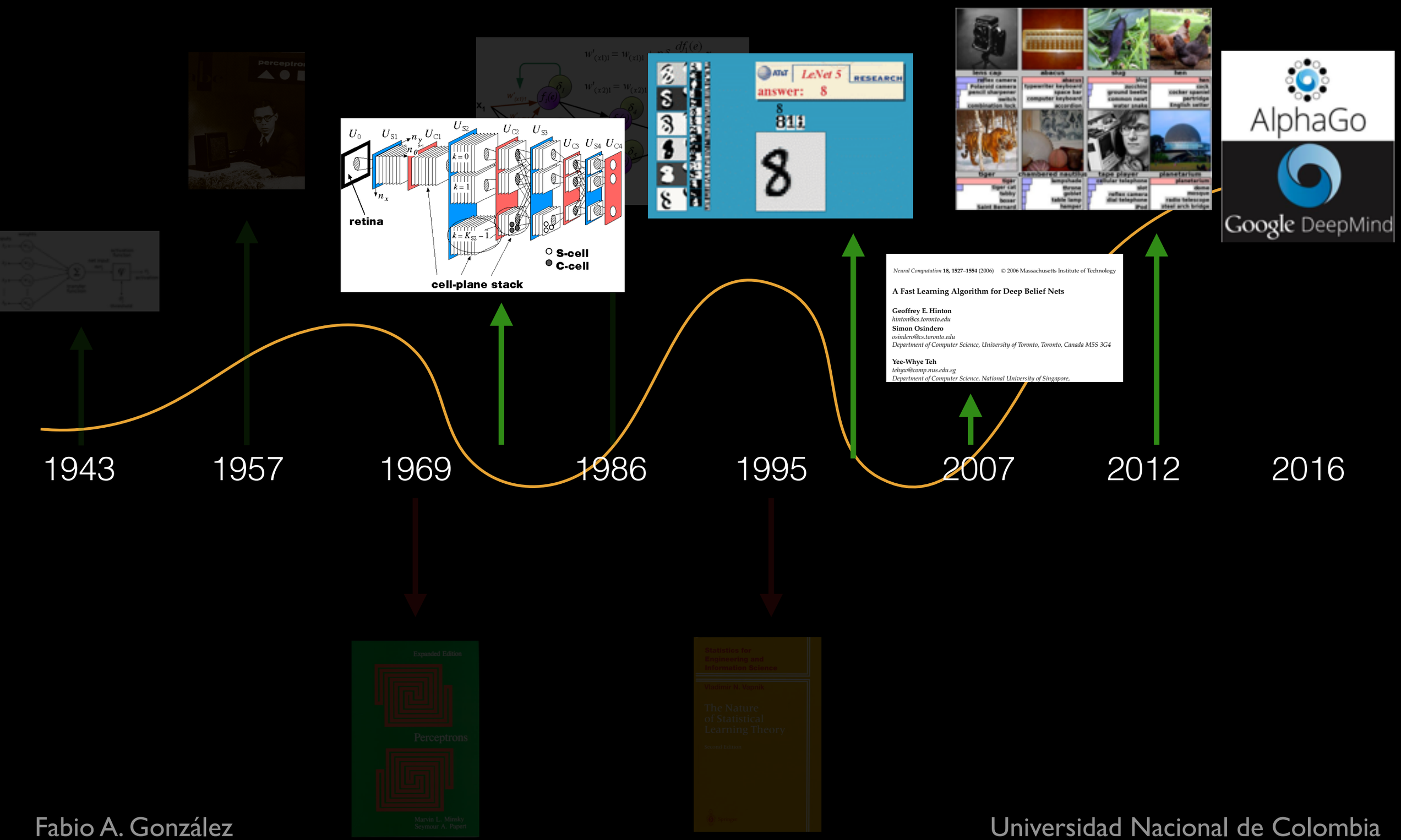
2016

Deep learning time line

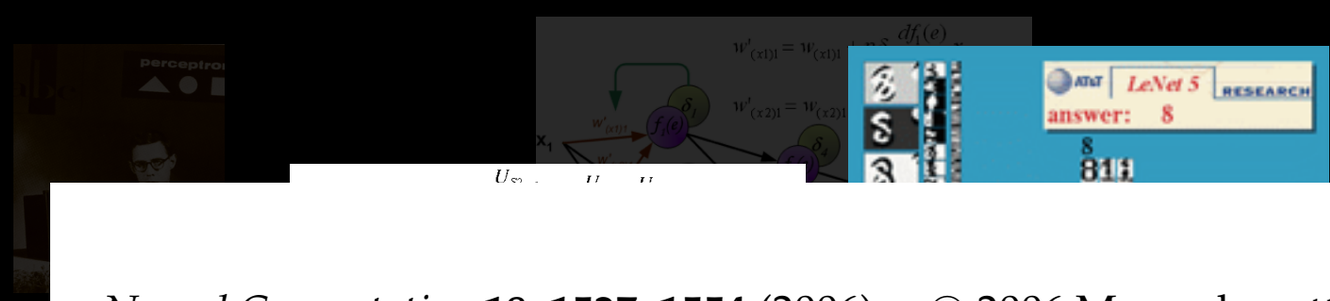


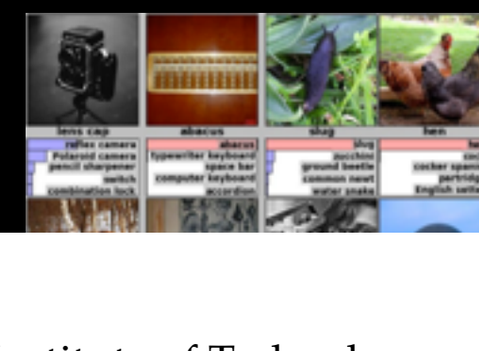
Deep learning time line

Deep learning time line



Deep learning time line





AlphaGo

Google DeepMind

Neural Computation **18, 1527–1554** (2006)

A Fast Learning Algorithm for Deep Belief Nets

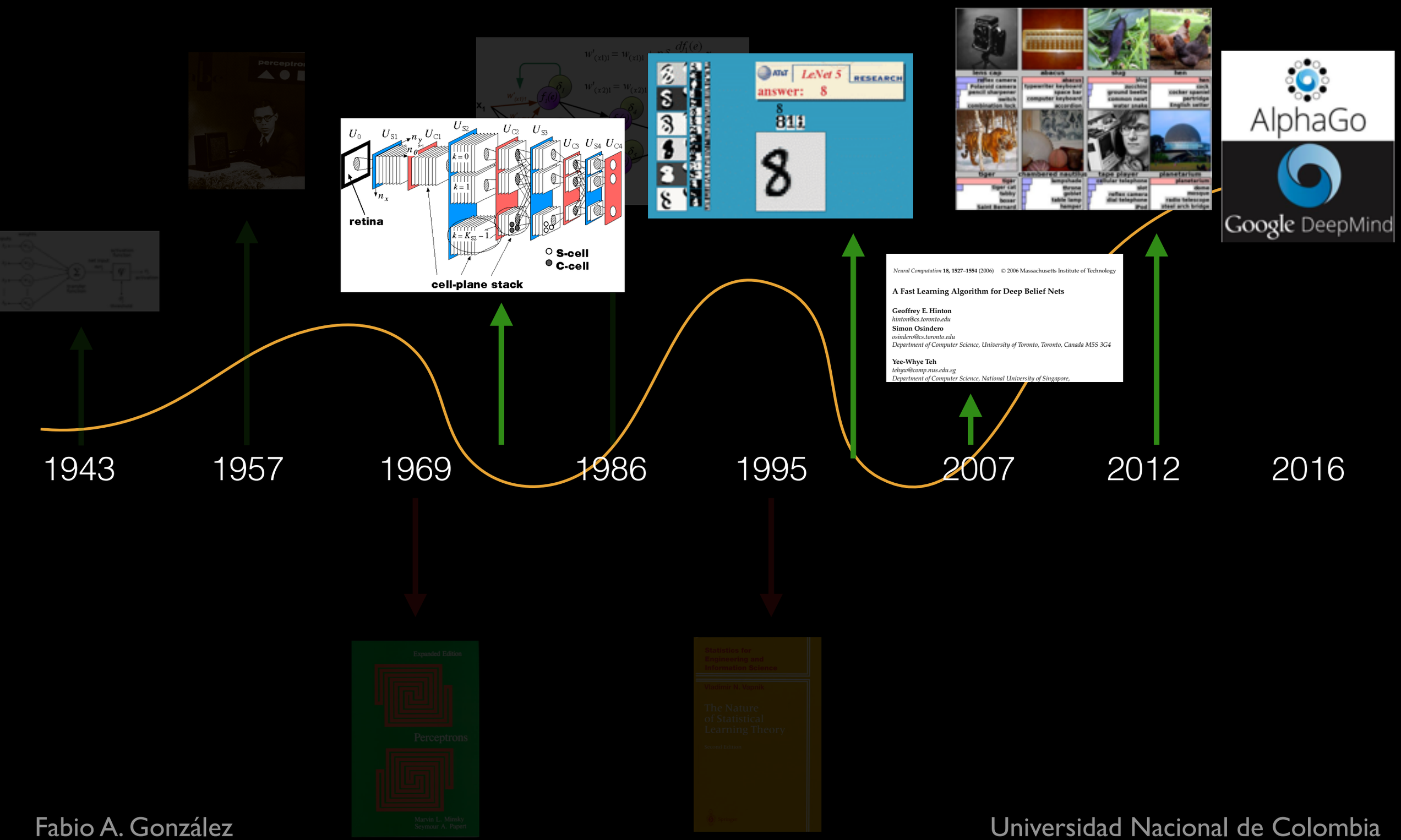
Geoffrey E. Hinton
hinton@cs.toronto.edu
Simon Osindero
osindero@cs.toronto.edu
Department of Computer Science, University of Toronto, Toronto, Canada M5S 3G4

Yee-Whye Teh
tehyw@comp.nus.edu.sg
Department of Computer Science, National University of Singapore,

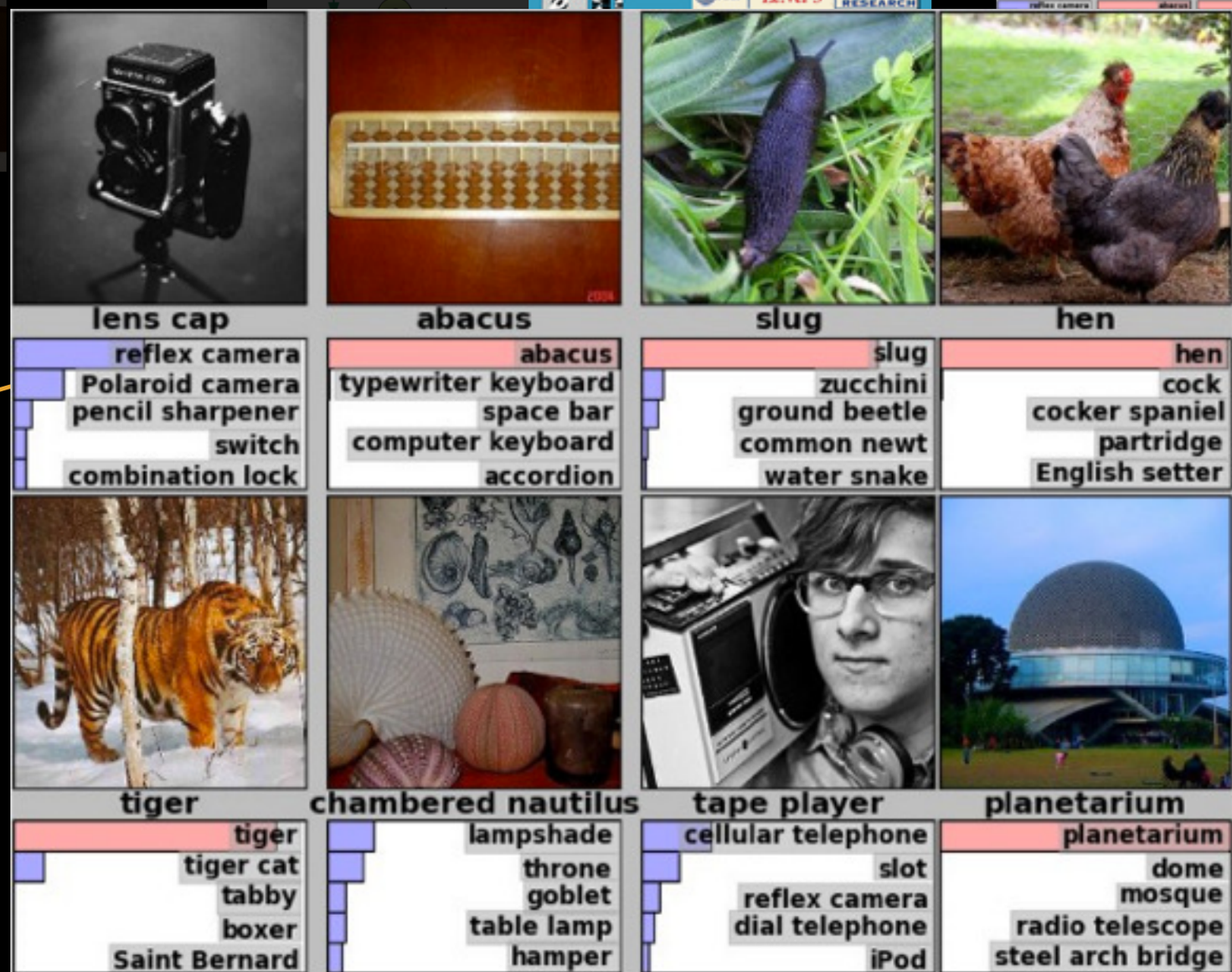
1943

2016

Deep learning time line

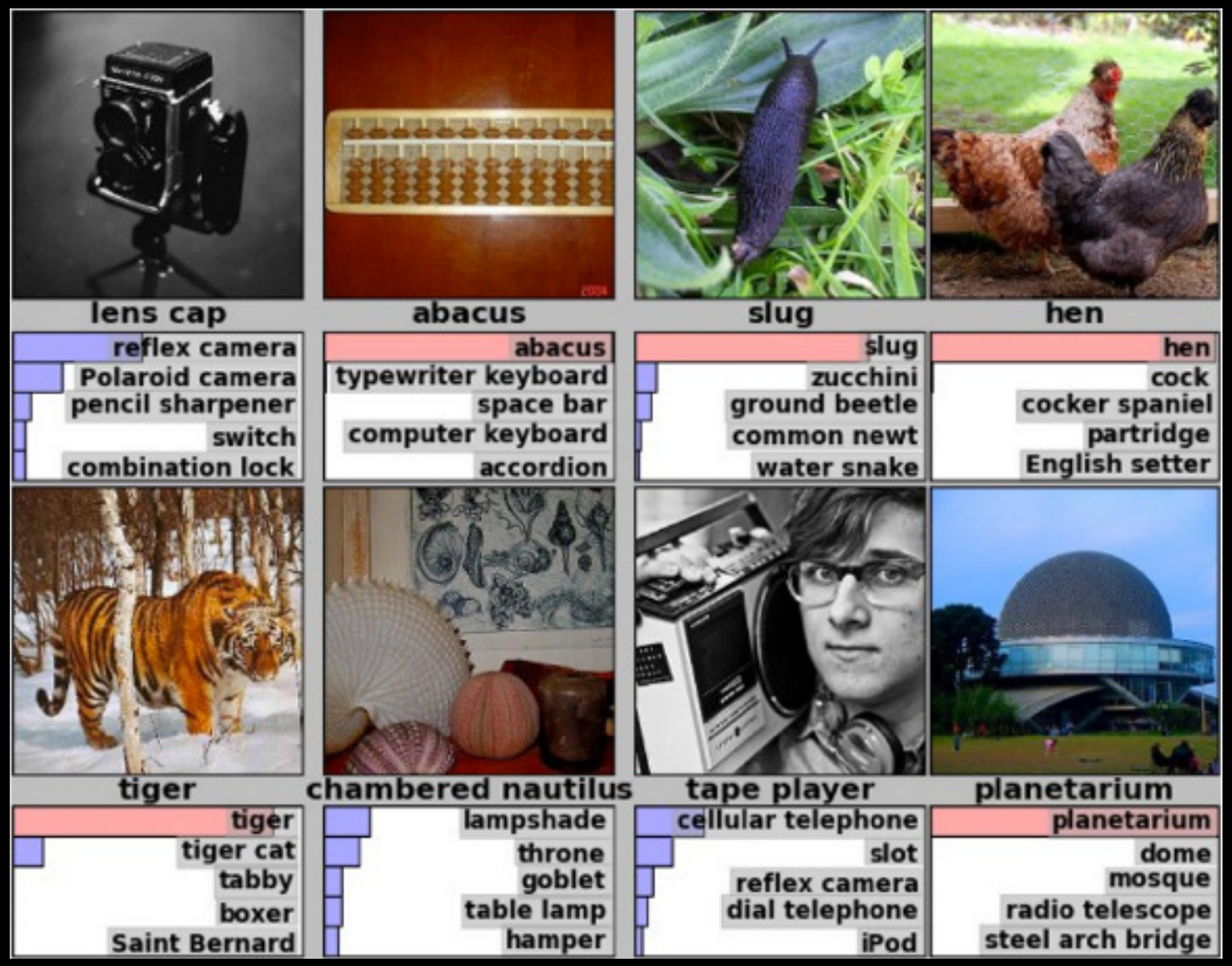


Deep learning time line



1943 1957 2012 2016

Deep learning model won ILSVRC 2012 challenge



Deep learning model won ILSVRC 2012 challenge

- ILSVRC 2012 (ImageNet Large Scale Visual Recognition)

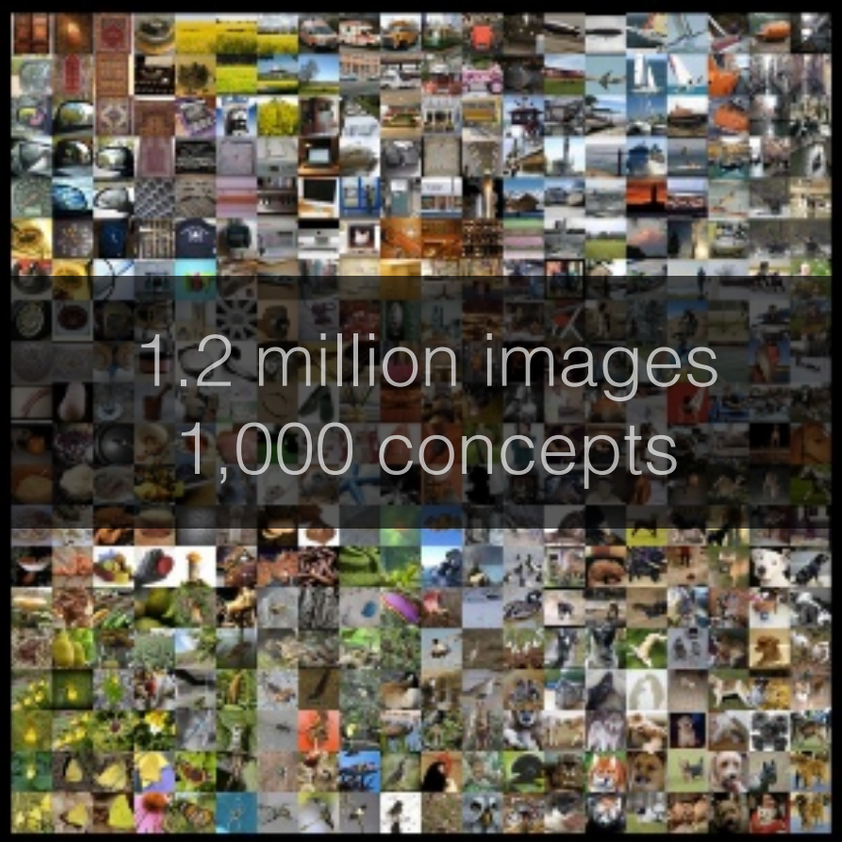
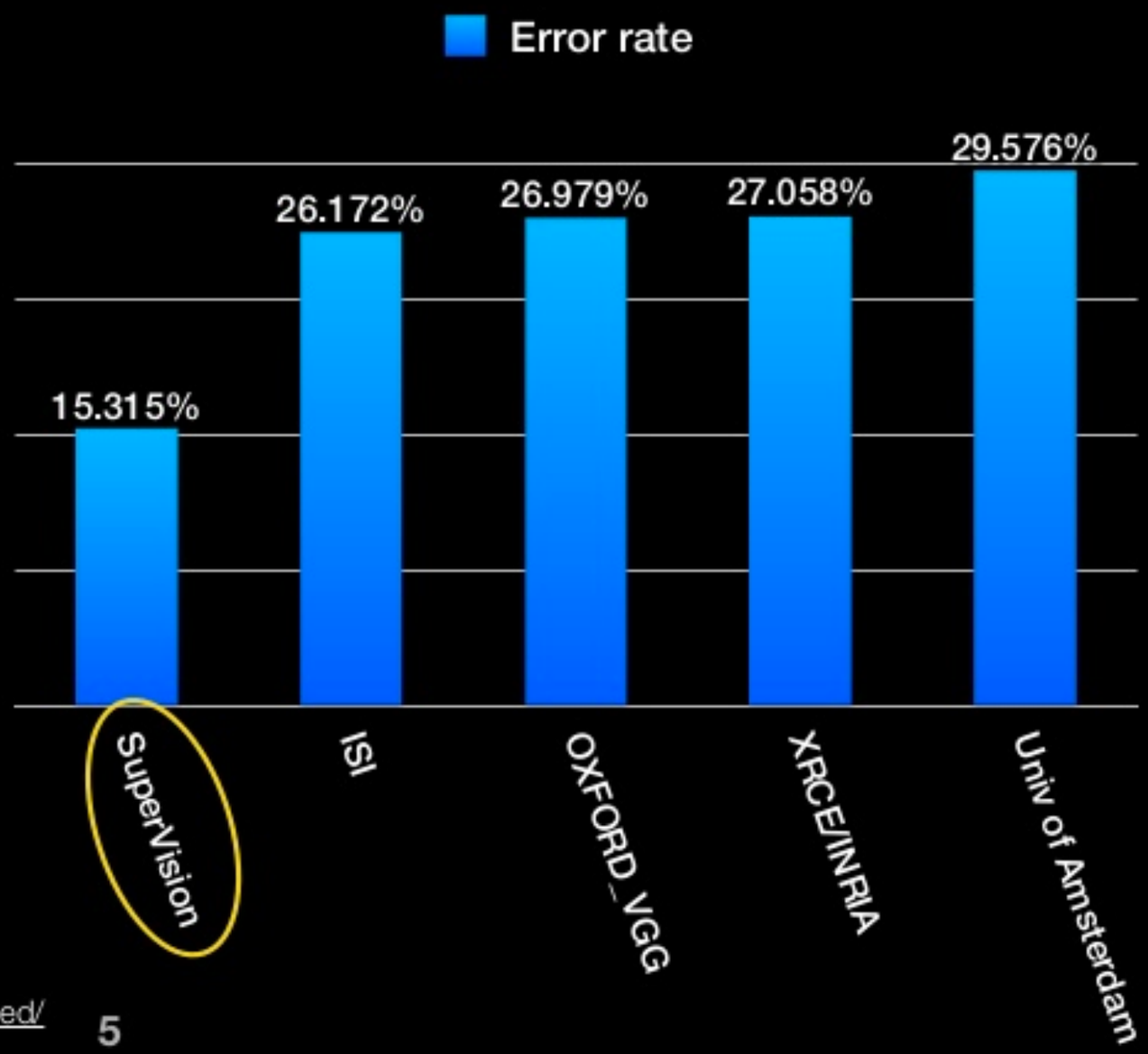


Image source: http://cs.stanford.edu/people/karpathy/cnnembed/



Deep learning model won ILSVRC 2012 challenge

- ILSV
  Larg
  Reco

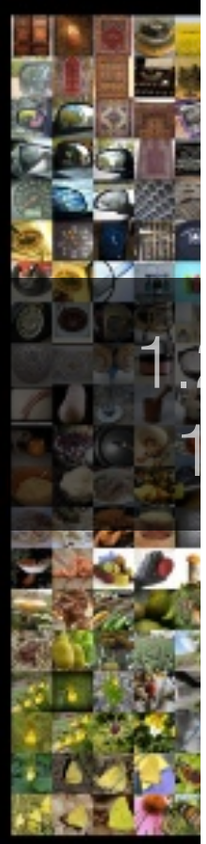
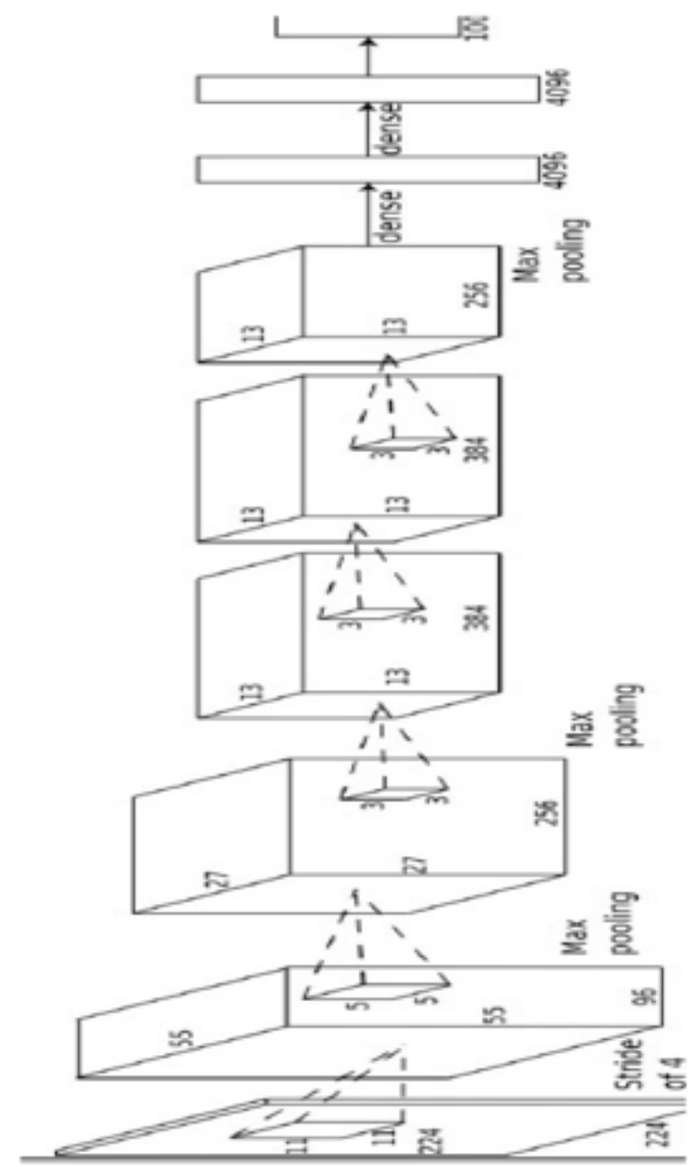


Image source

Won the 2012 ImageNet LSVRC. 60 Million parameters, 832M MAC ops

4M	FULL CONNECT	4Mflop
16M	FULL 4096/ReLU	16M
37M	FULL 4096/ReLU	37M
	MAX POOLING	
442K	CONV 3x3/ReLU 256fm	74M
1.3M	CONV 3x3ReLU 384fm	224M
884K	CONV 3x3/ReLU 384fm	149M
	MAX POOLING 2x2sub	
	LOCAL CONTRAST NORM	
307K	CONV 11x11/ReLU 256fm	223M
	MAX POOL 2x2sub	
	LOCAL CONTRAST NORM	
35K	CONV 11x11/ReLU 96fm	105M



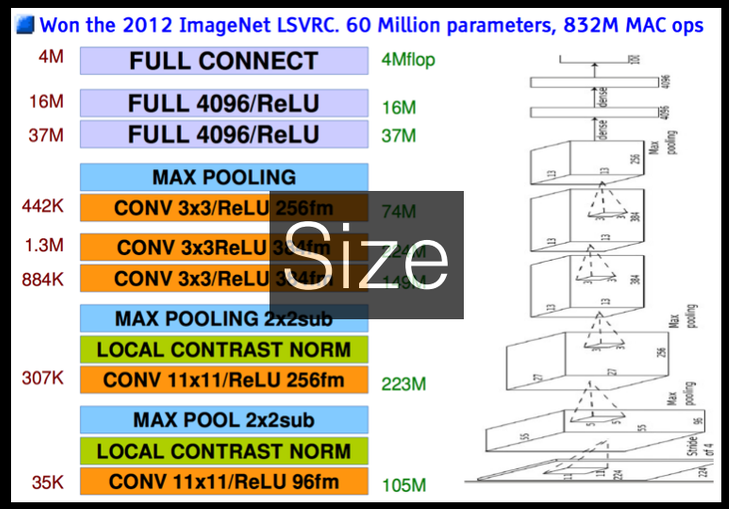
76%

of Amsterdam

Deep learning recipe

Data

Algorithms



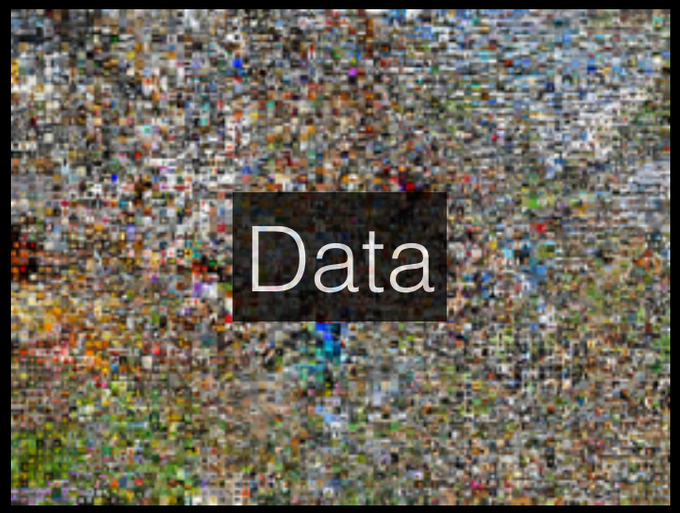
Size

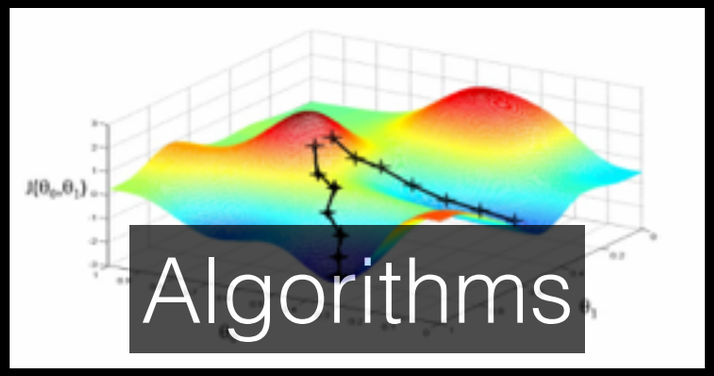
Tricks

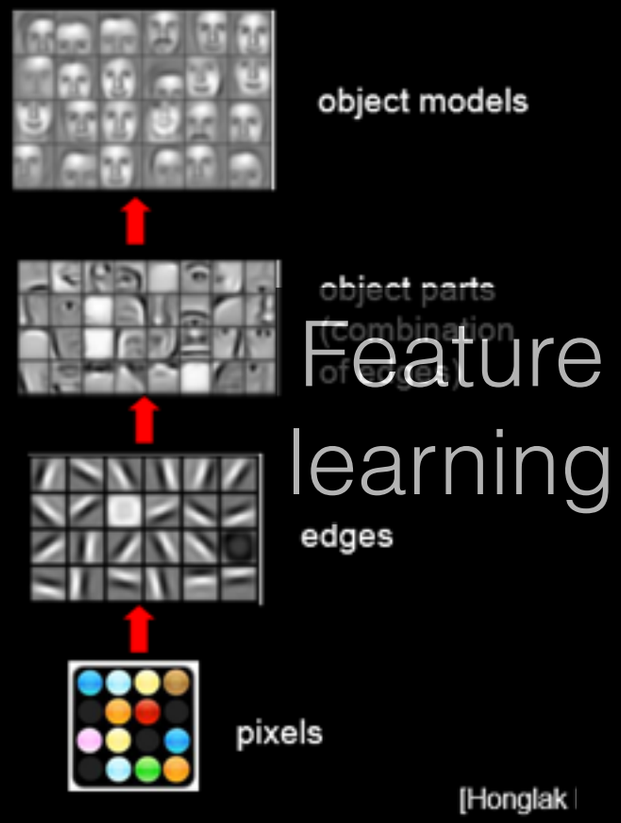
Feature learning

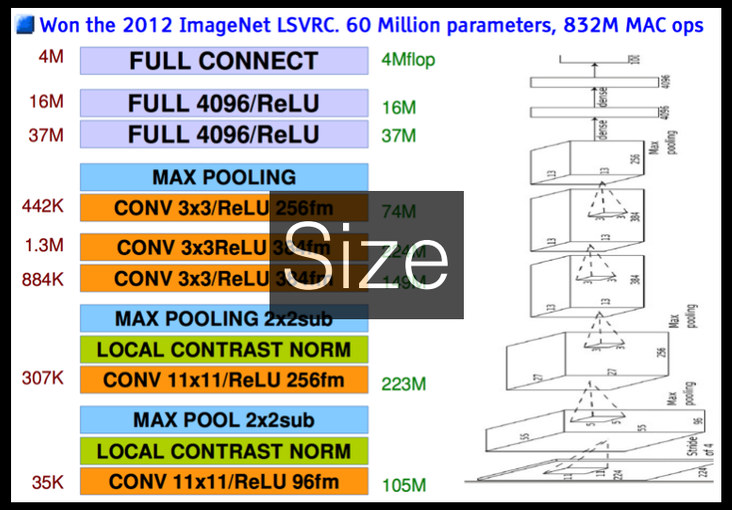
HPC

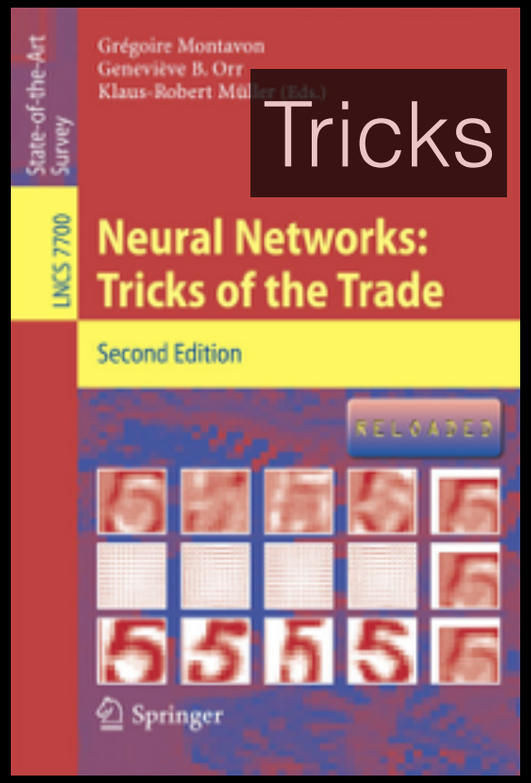
Deep learning recipe



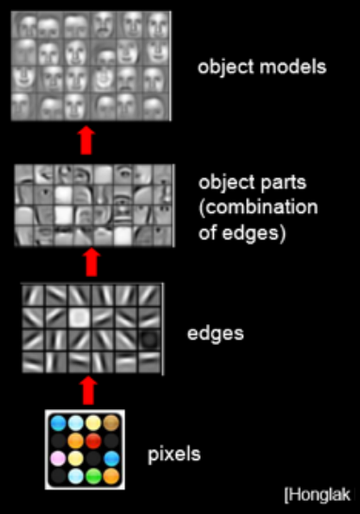




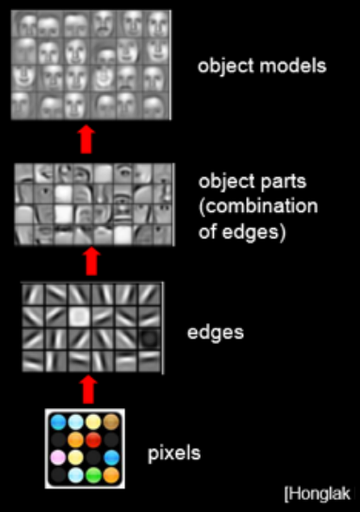


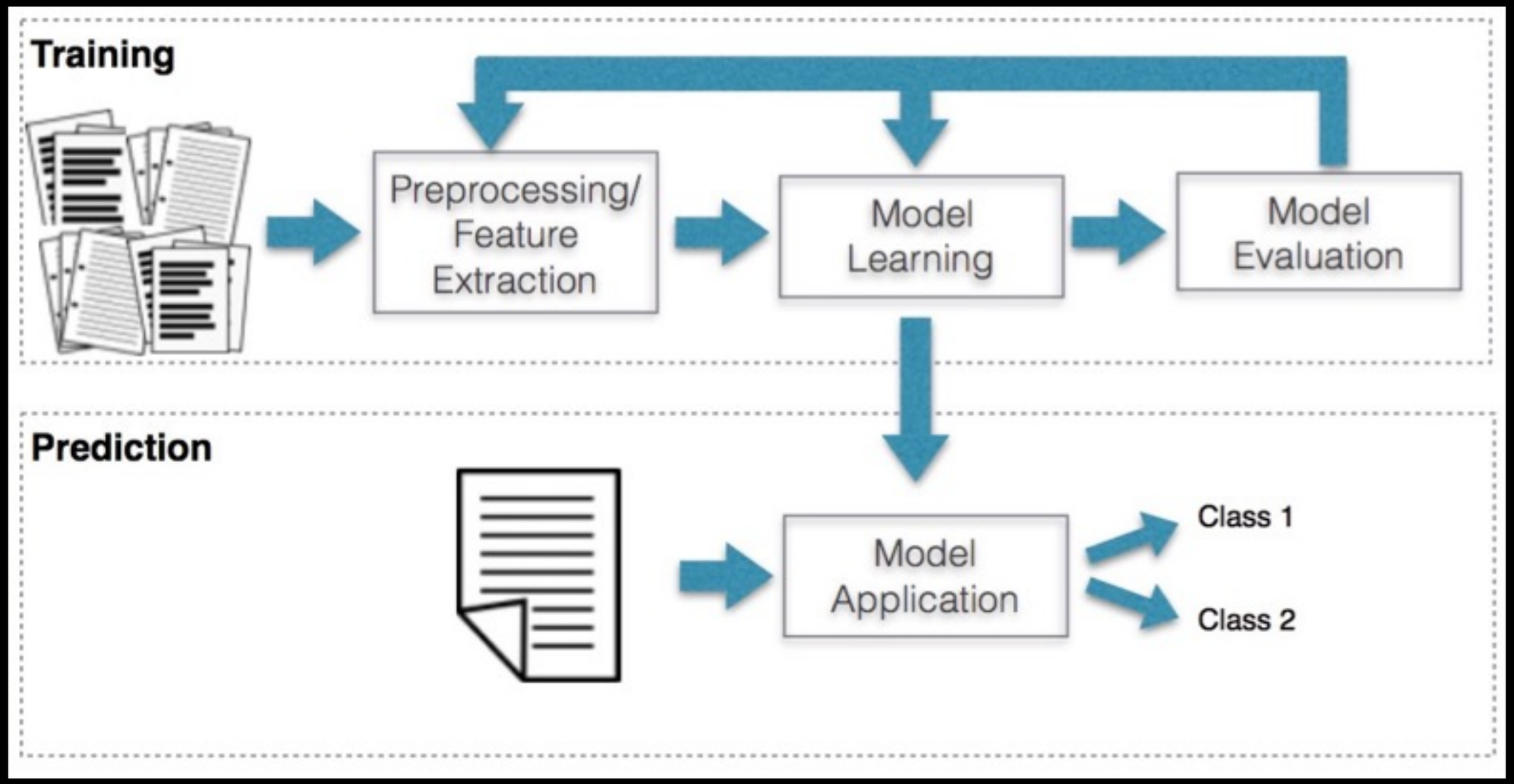


Feature learning

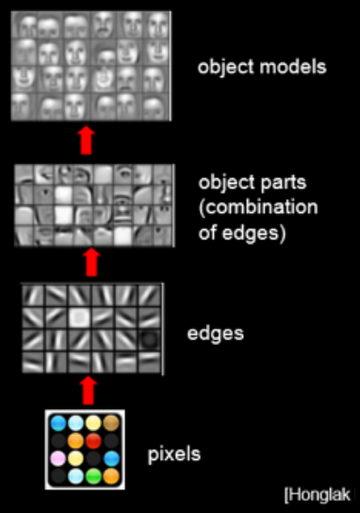


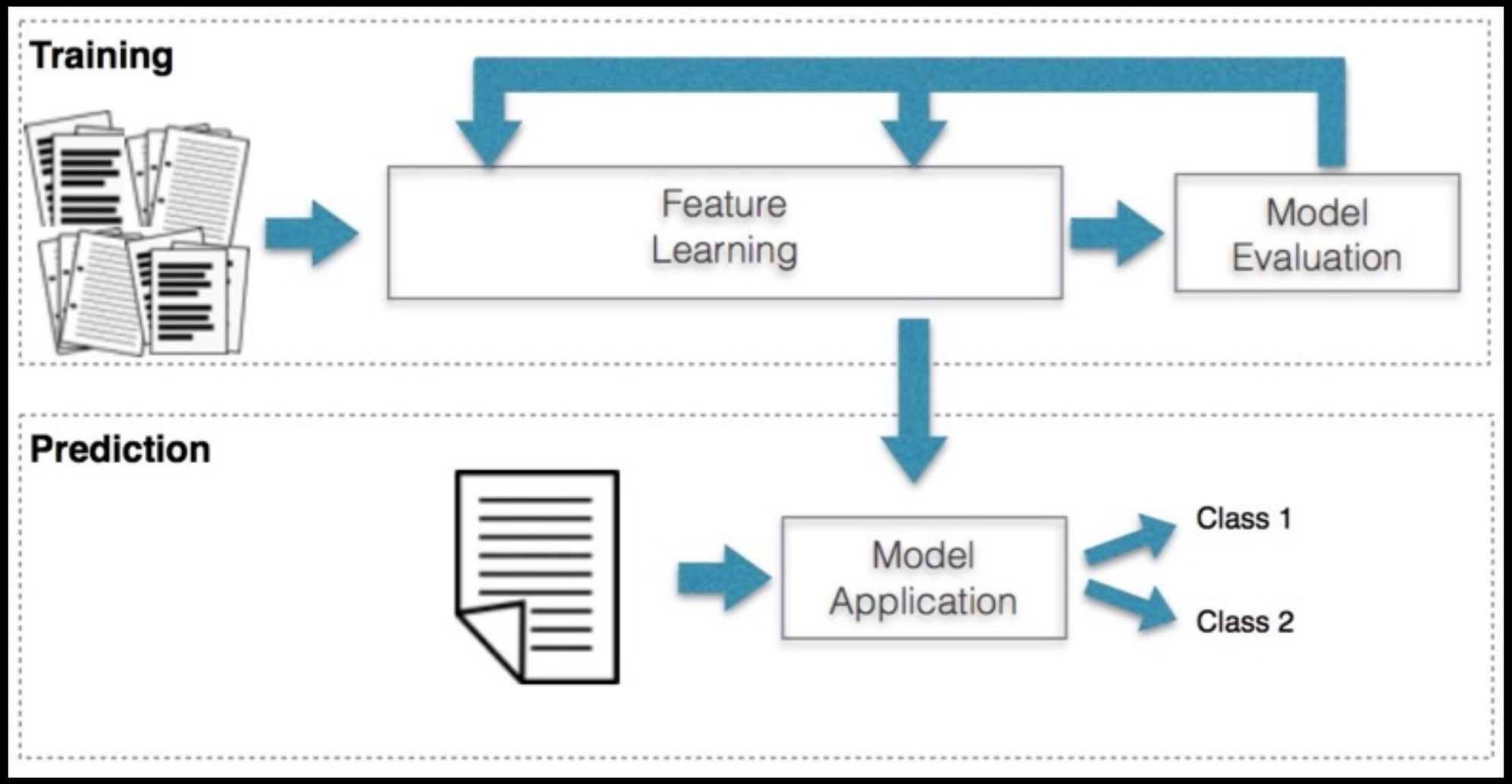
Feature learning



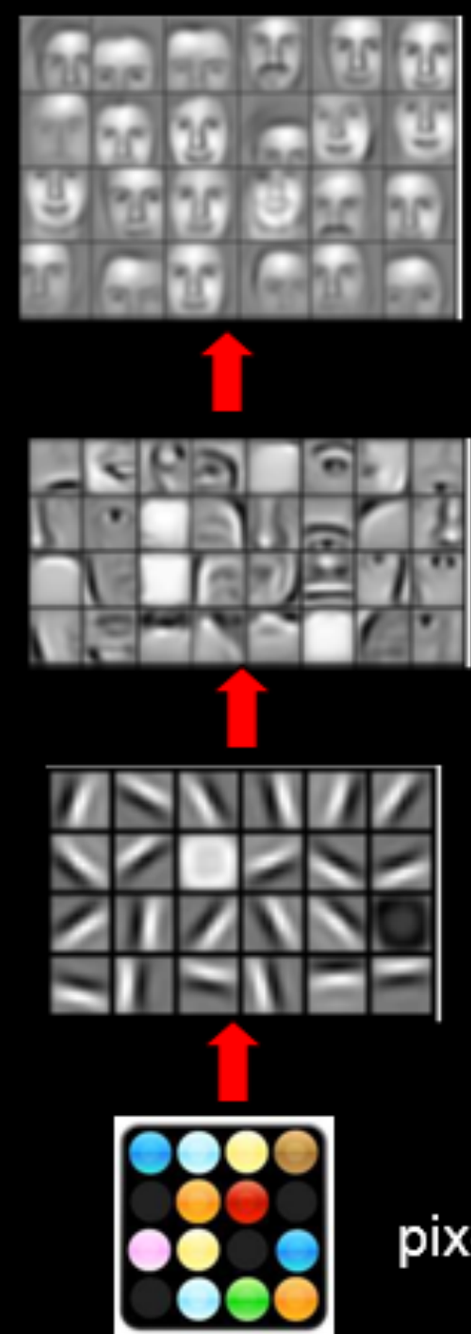


Feature learning



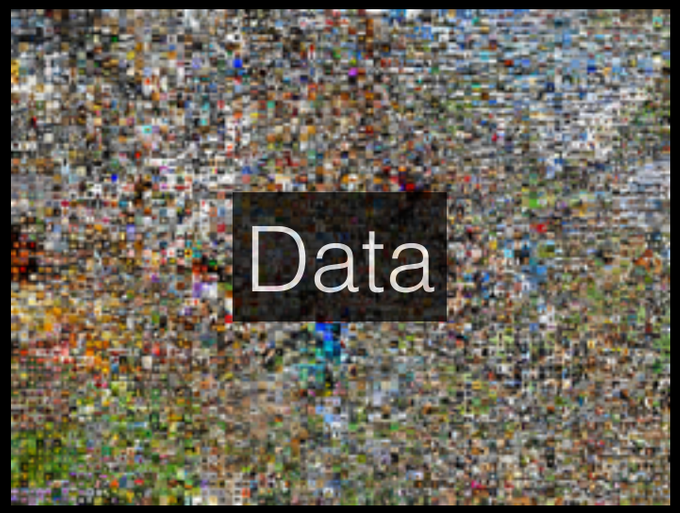


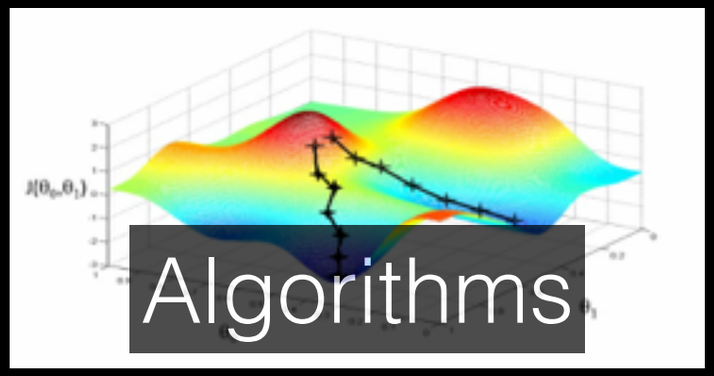
Feature learning

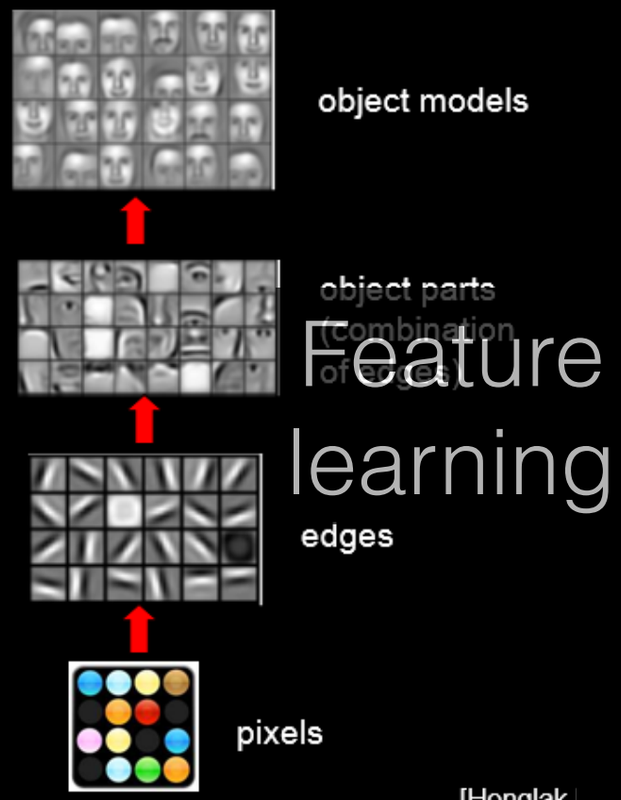


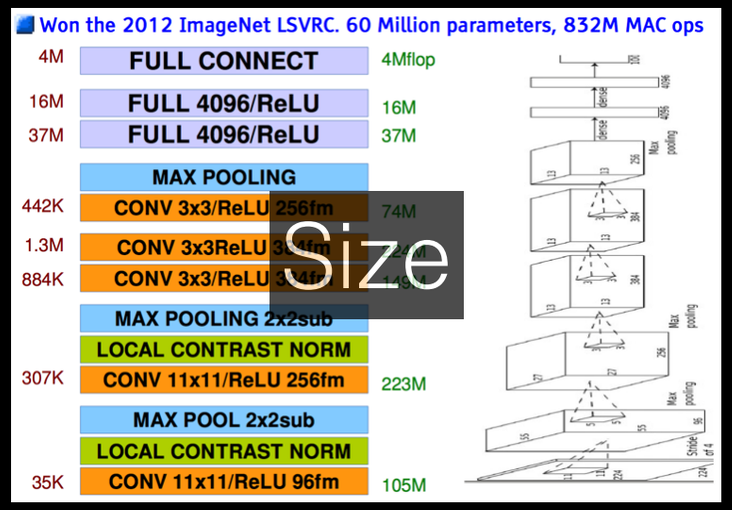
[Honglak

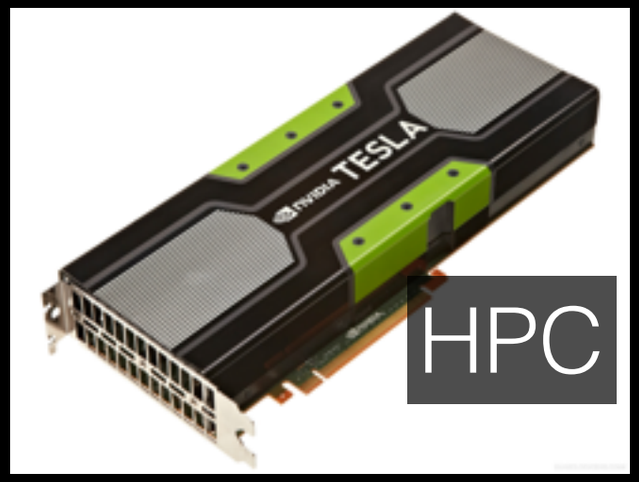
Deep learning recipe

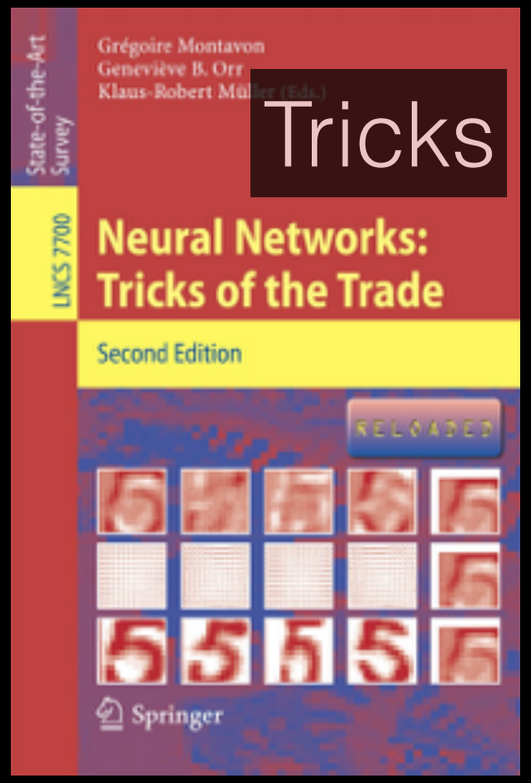




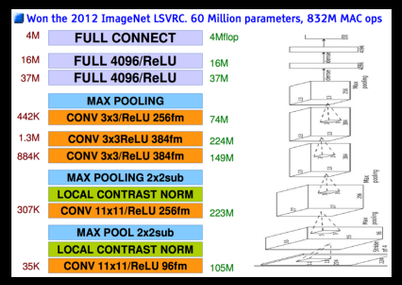


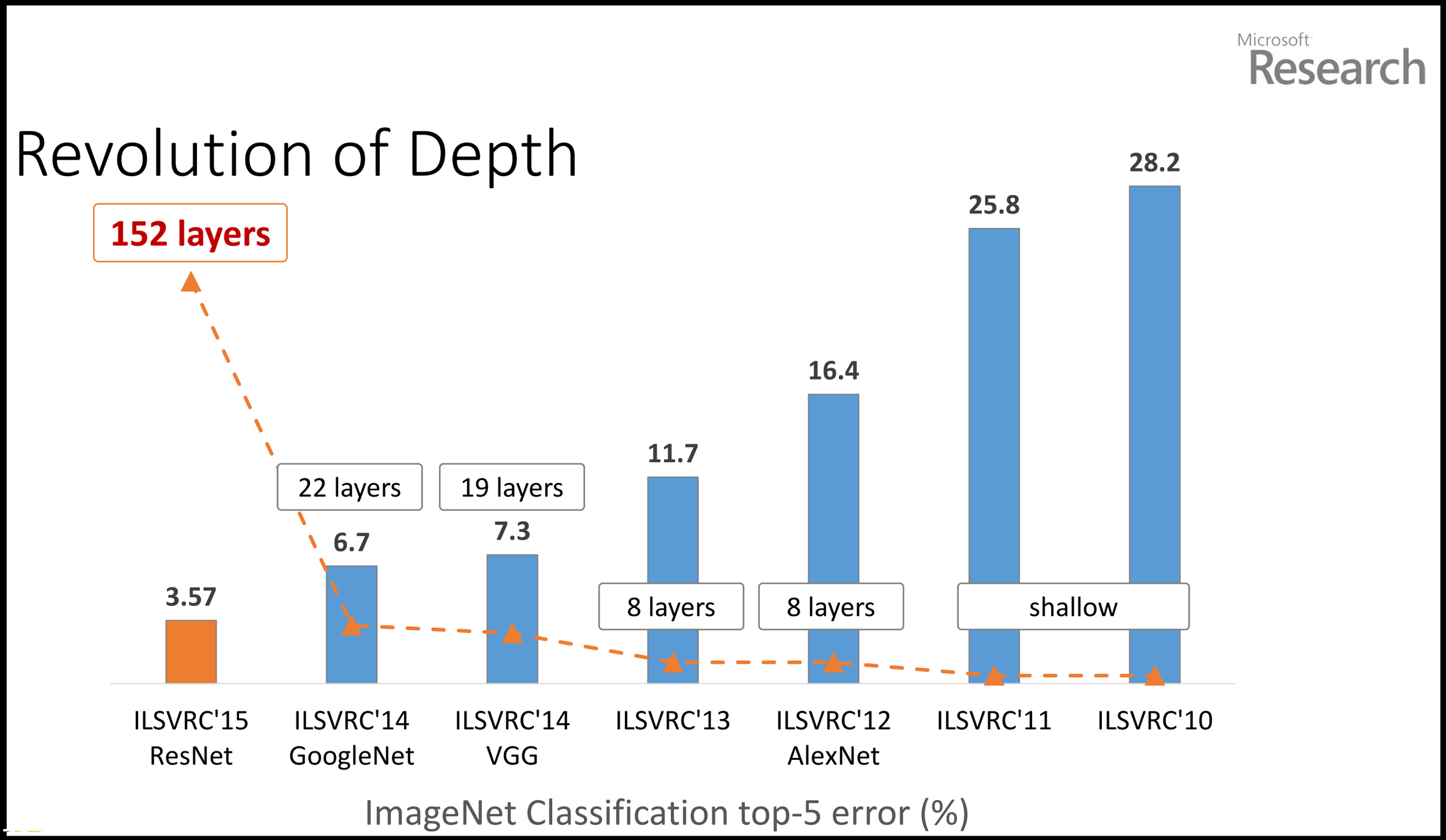




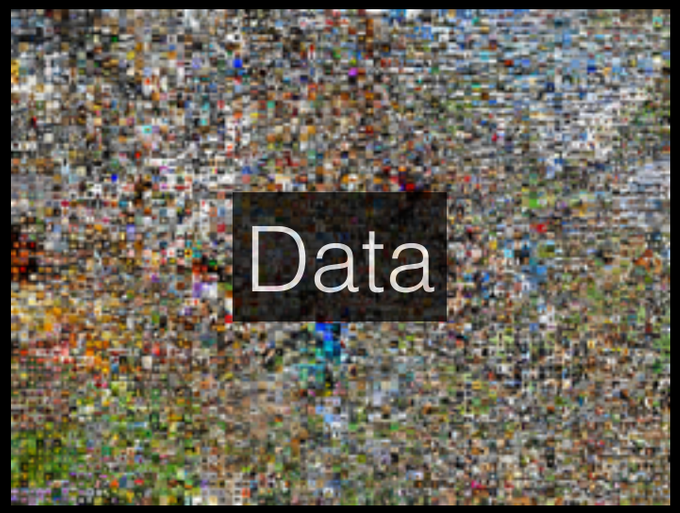


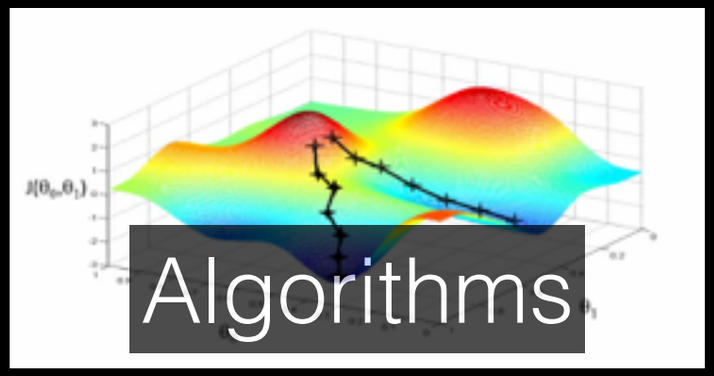
Deep —> Bigger

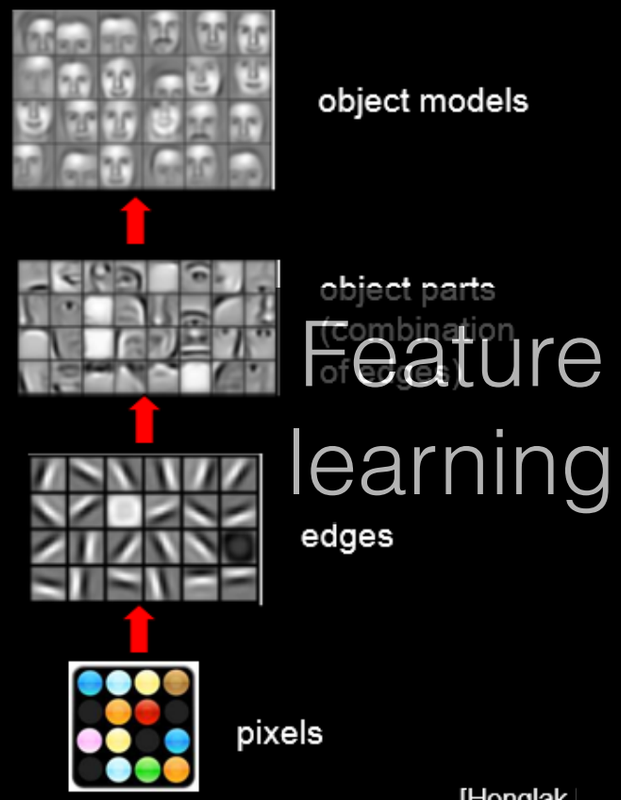

Won the 2012 ImageNet LSVRC. 60 Million parameters, 832M MAC ops
4M FULL CONNECT 4Mflop
16M FULL 4096/ReLU 16M
37M FULL 4096/ReLU 37M
MAX POOLING
442K CONV 3x3/ReLU 256fm 74M
1.3M CONV 3x3ReLU 384fm 224M
884K CONV 3x3/ReLU 384fm 149M
MAX POOLING 2x2sub
LOCAL CONTRAST NORM
307K CONV 11x11/ReLU 256fm 223M
MAX POOL 2x2sub
LOCAL CONTRAST NORM
35K CONV 11x11/ReLU 96fm 105M

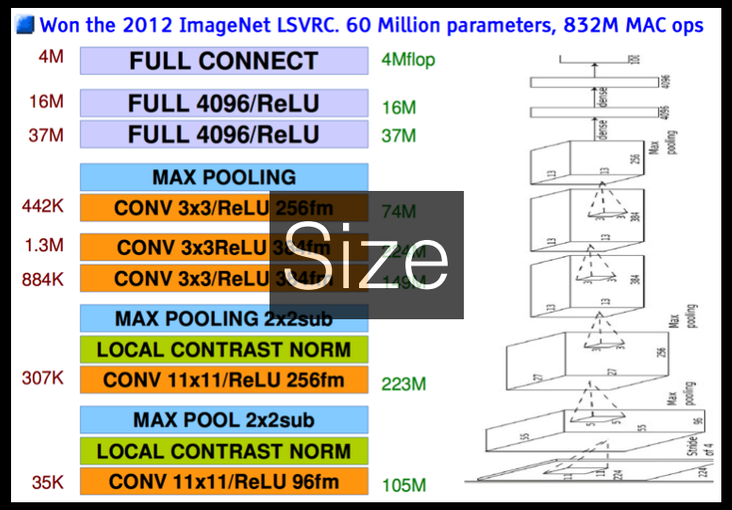

Microsoft Research
Revolution of Depth
152 layers
22 layers
19 layers
8 layers
8 layers
shallow
3.57
6.7
7.3
11.7
16.4
25.8
28.2
ILSVRC'15 ResNet
ILSVRC'14 GoogleNet
ILSVRC'14 VGG
ILSVRC'13
ILSVRC'12 AlexNet
ILSVRC'11
ILSVRC'10
ImageNet Classification top-5 error (%)

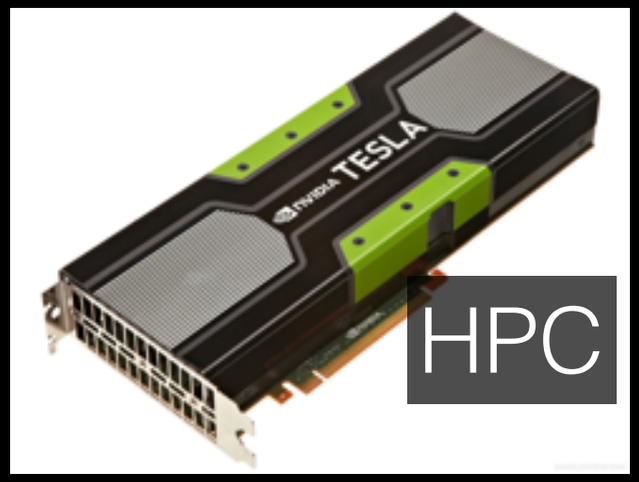
Deep learning recipe

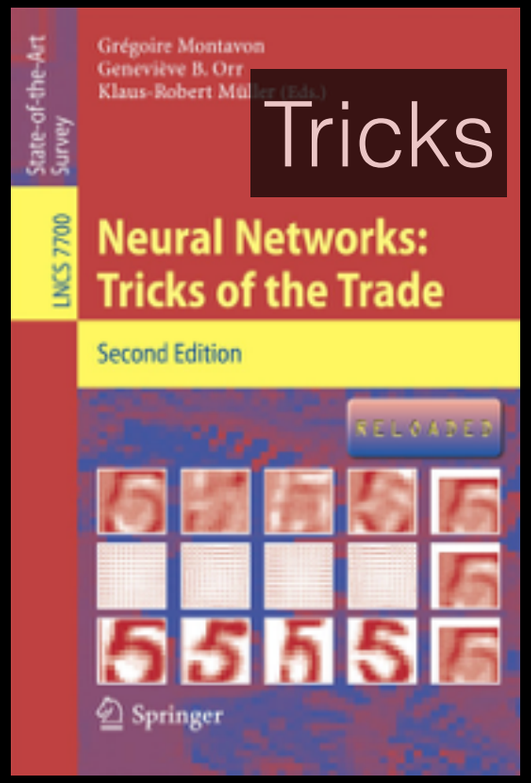






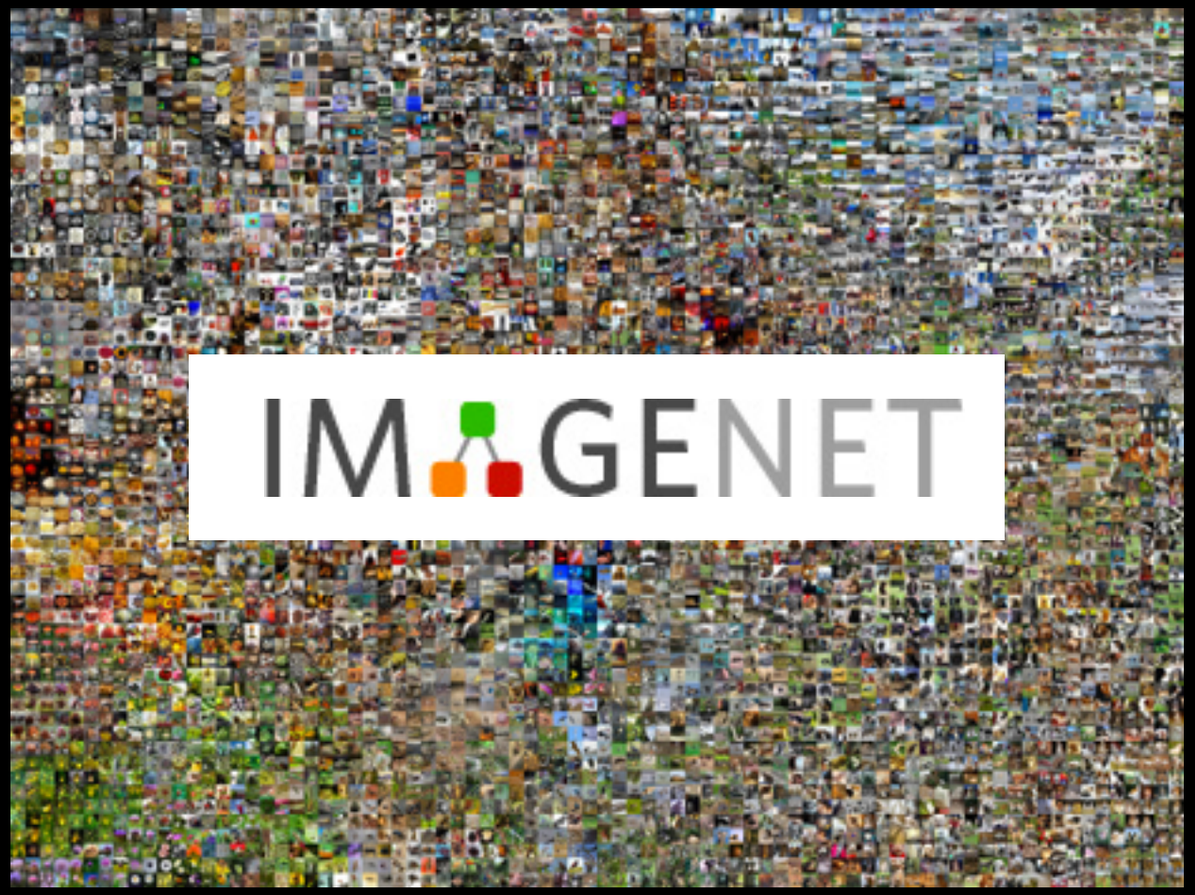




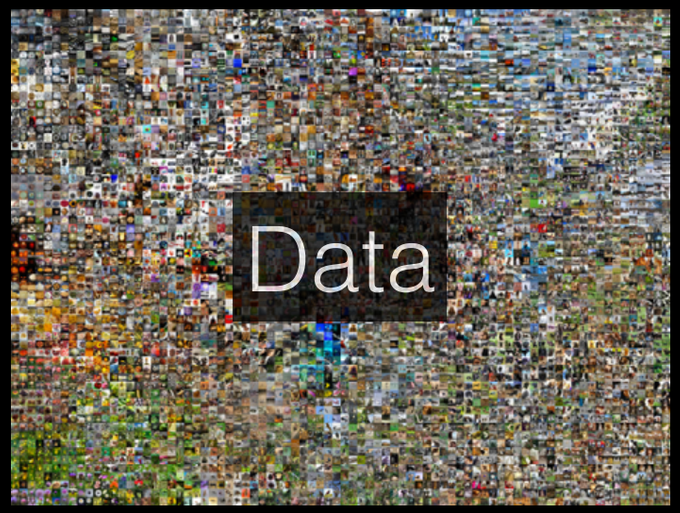


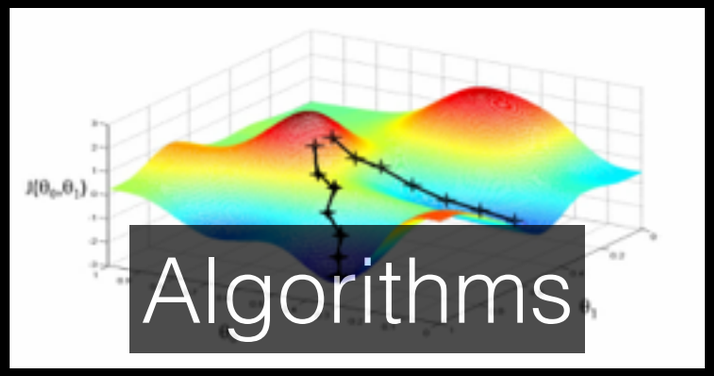
Data...

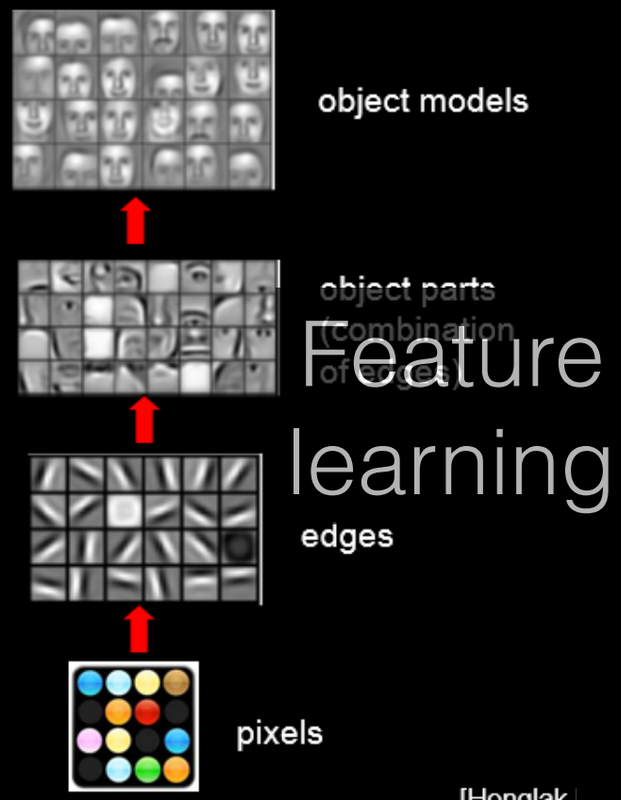
- Images annotated with WordNet concepts
- Concepts: 21,841
- Images: 14,197,122
- Bounding box annotations: 1,034,908
- Crowdsourcing



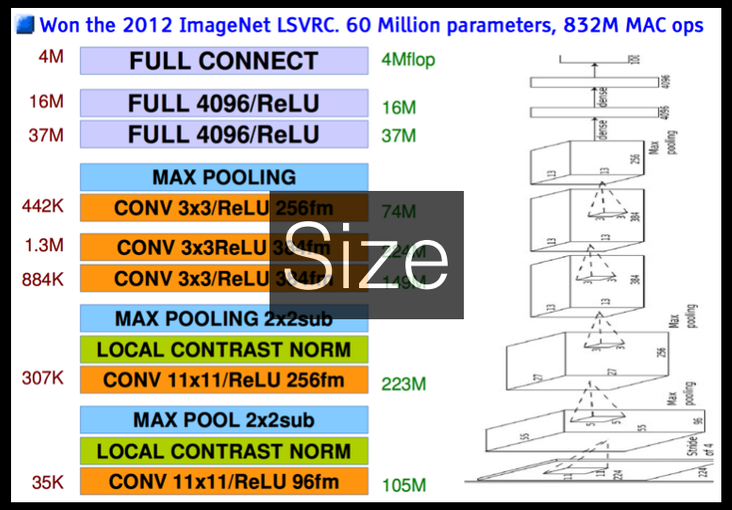
Deep learning recipe



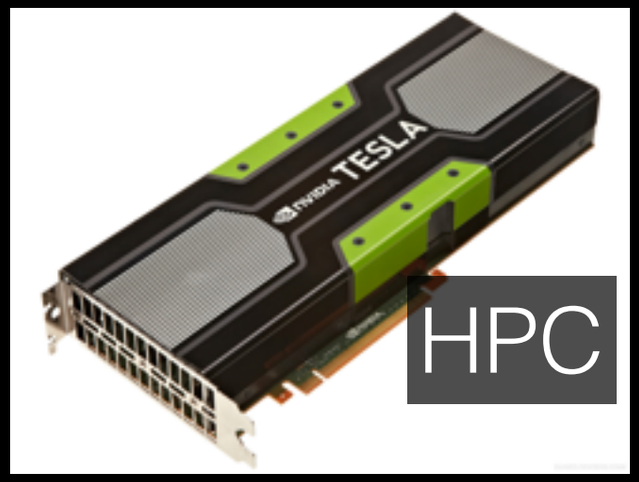




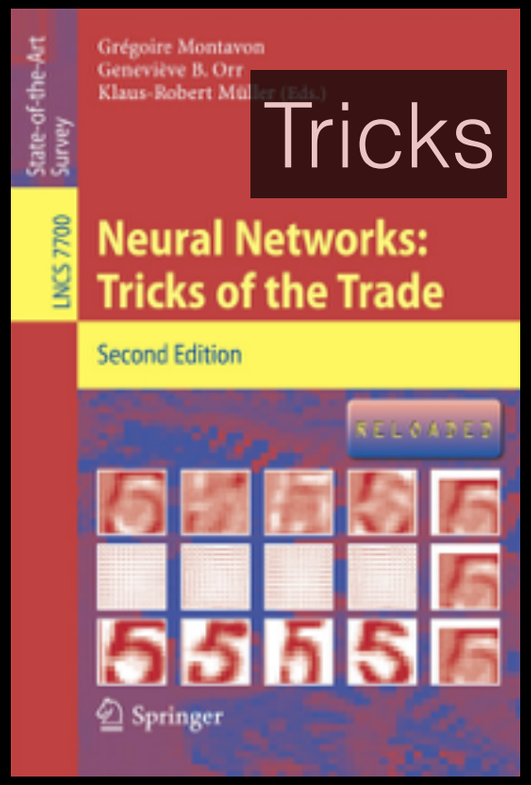
Feature learning



Size



HPC



Tricks

HPC

Tesla K40

Larger datasets
Faster acceleration
Dynamic Power Scaling
Same power

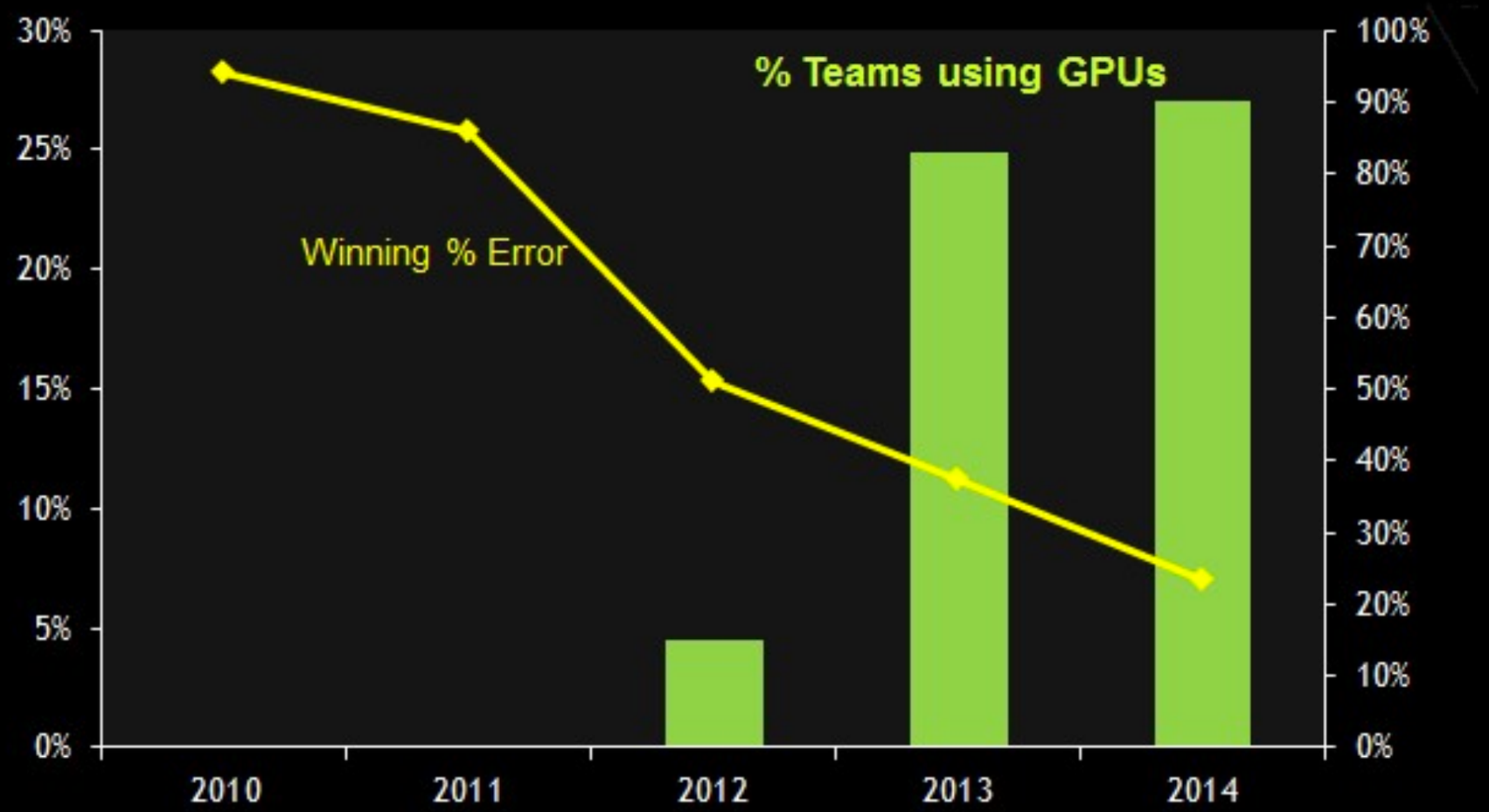
Expands GPU Opportunity & Lead over Competition

K40 specs are preliminary

	K20X	K40 (Atlas)
GPU(s)	1x GK110	1x GK180
Peak Single Precision Peak SGEMM	3.95 TF 2.90 TF	>4.0 TF
Peak Double Precision Peak DGEMM	1.32 TF 1.22 TF	>1.4 TF
Memory size	6 GB	12 GB
Memory BW (ECC off)	250 GB/s	288 GB/s
Workload for Boost Clocks	-	AMBER, ANSYS
PCIe Gen	Gen 2	Gen 3
# CUDA Cores	2688	2880
Total Board Power	235W	235W (245W SXM)
Form factor	PCIe Passive, SXM	PCIe Passive, Active & TTP, SXM

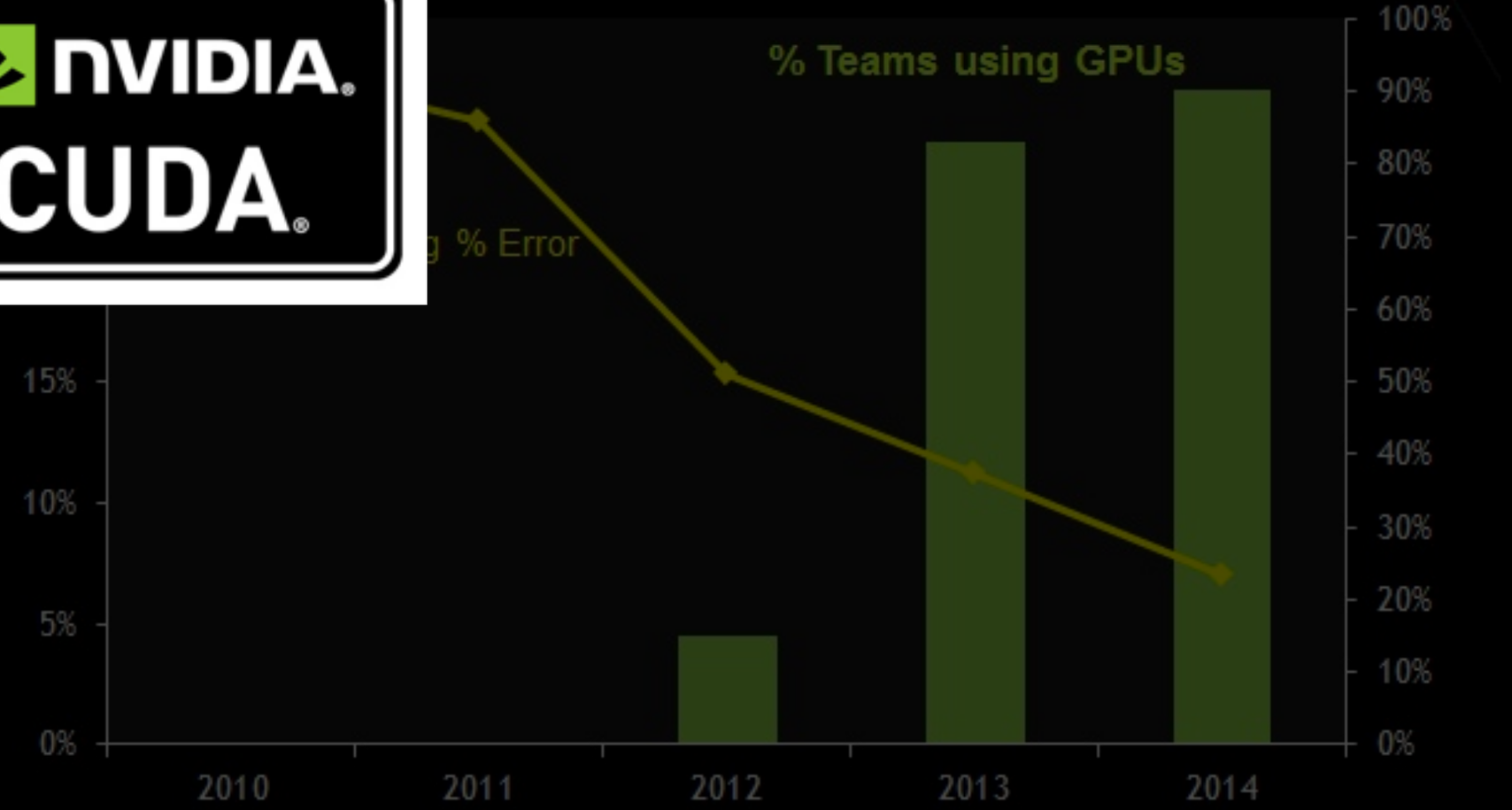
HPC


TESLA


% Teams using GPUs
Winning % Error
30%
25%
20%
15%
10%
5%
0%
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
2010
2011
2012
2013
2014

HPC


TESLA


% Teams using GPUs
% Error
15%
10%
5%
0%
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
2010
2011
2012
2013
2014

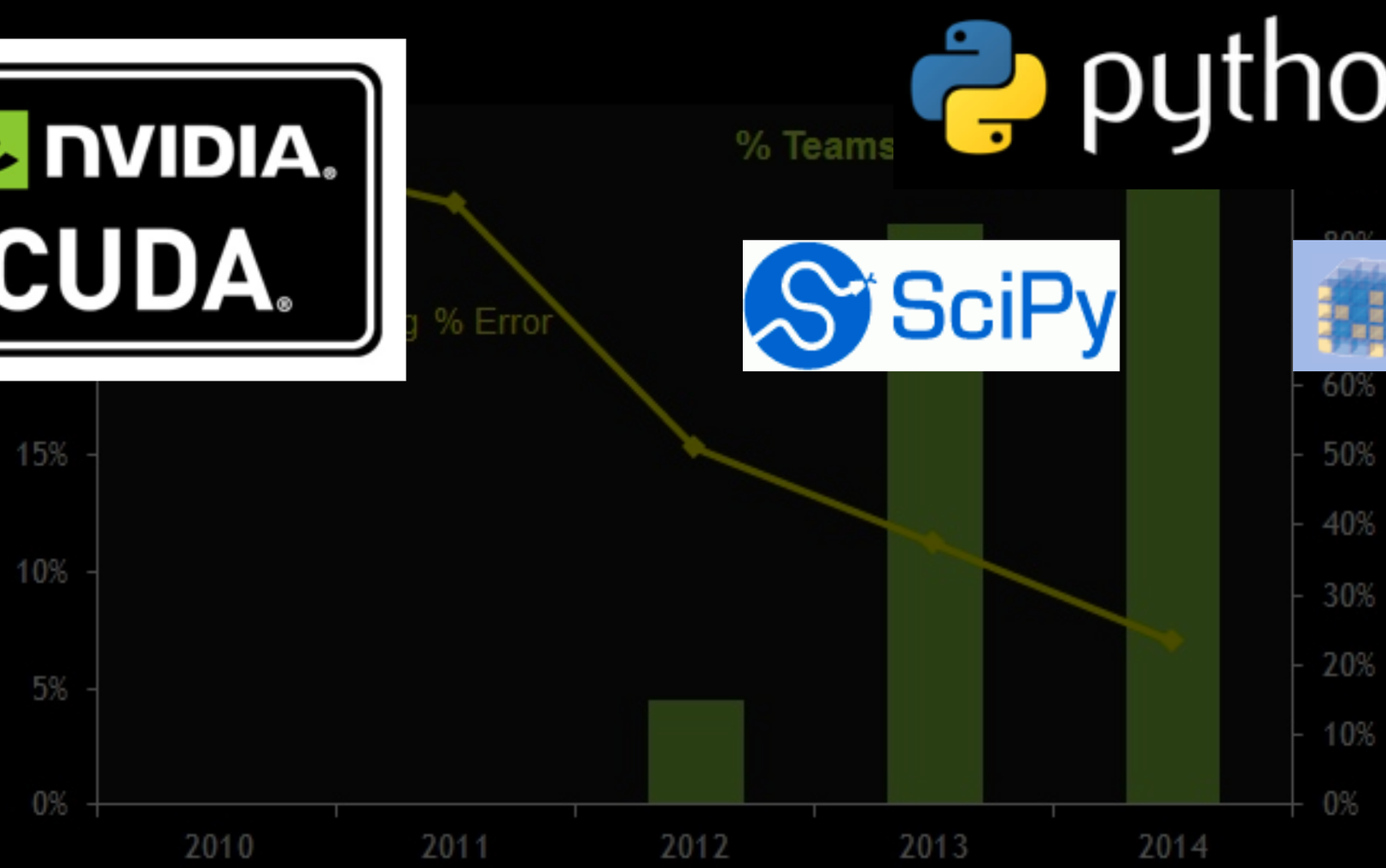
HPC

NVIDIA CUDA

python™

SciPy

NumPy



HPC

NVIDIA CUDA

python™

SciPy

NumPy

CONTINUUM ANALYTICS

theano

TensorFlow

HPC

NVIDIA CUDA

python™

SciPy

NumPy

CONTINUUM ANALYTICS

theano

torch
A SCIENTIFIC COMPUTING FRAMEWORK FOR LUAJIT

TensorFlow

Decaf / Caffe
a Berkeley Vision Project

HPC

NVIDIA CUDA

python™

SciPy

NumPy

CONTINUUM ANALYTICS

Keano

torch
A SCIENTIFIC COMPUTING FRAMEWORK FOR LUAJIT

TensorFlow

Decaf / Caffe
a Berkeley Vision Project

HPC

HPC


```
int main( void ) {
    CPUBitmap bitmap( DIM, DIM );
    unsigned char    *dev_bitmap;

    HANDLE_ERROR( cudaMalloc( (void**)&dev_bitmap,
                              bitmap.image_size() ) );

    dim3    grid(DIM,DIM);
    kernel<<<grid,1>>>( dev_bitmap );

    HANDLE_ERROR( cudaMemcpy( bitmap.get_ptr(),
                              dev_bitmap,
                              bitmap.image_size(),
                              cudaMemcpyDeviceToHost ) );
    bitmap.display_and_exit();

    cudaFree( dev_bitmap );
}
```

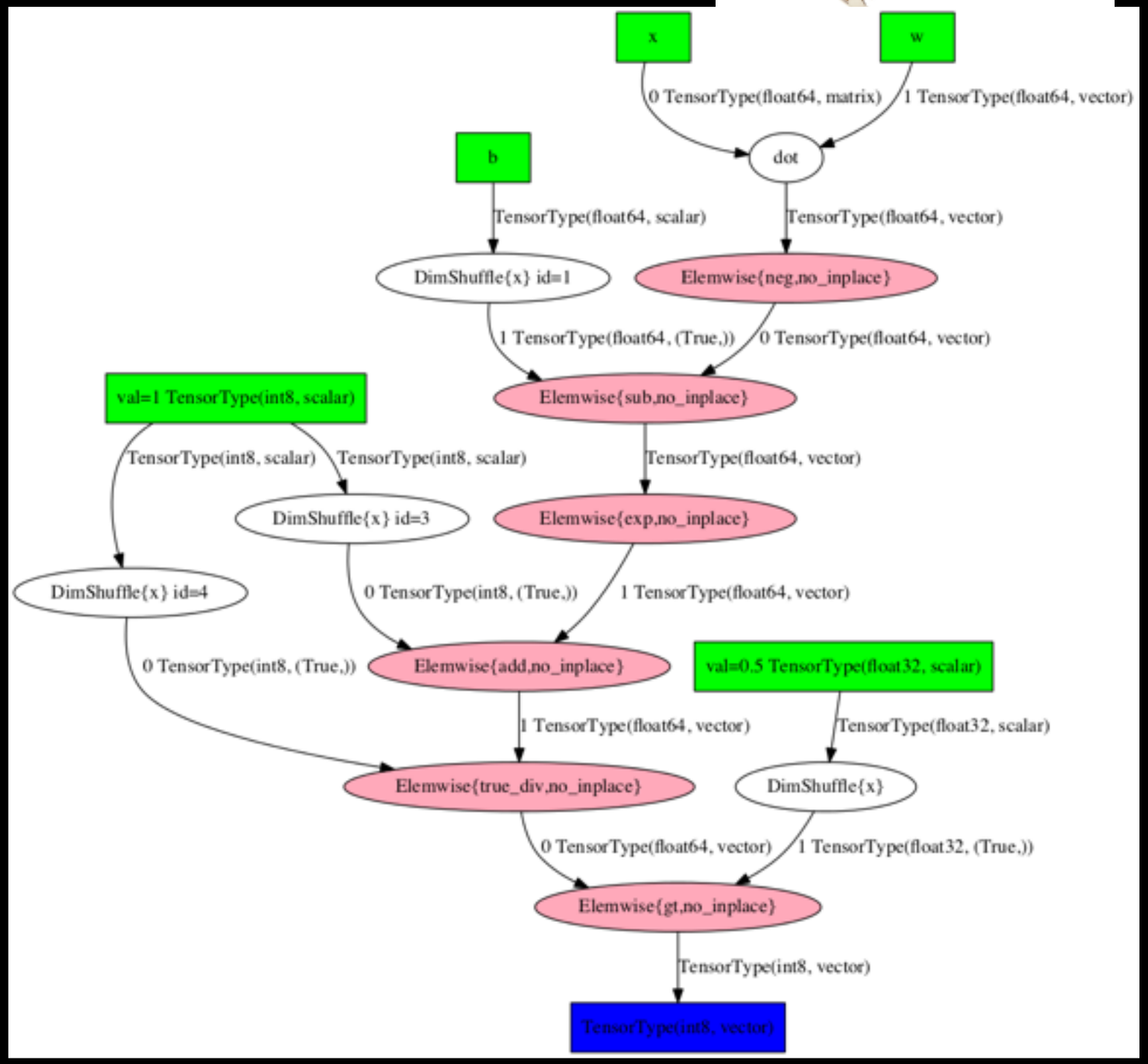
HPC


```python
# Declare Theano symbolic variables
x = T.matrix("x")
y = T.vector("y")
w = theano.shared(rng.randn(feats).astype(theano.config.floatX), name="w")
b = theano.shared(numpy.asarray(0., dtype=theano.config.floatX), name="b")
x.tag.test_value = D[0]
y.tag.test_value = D[1]

# Construct Theano expression graph
p_1 = 1 / (1 + T.exp(-T.dot(x, w)-b)) # Probability of having a one
prediction = p_1 > 0.5 # The prediction that is done: 0 or 1
xent = -y*T.log(p_1) - (1-y)*T.log(1-p_1) # Cross-entropy
cost = xent.mean() + 0.01*(w**2).sum() # The cost to optimize
gw,gb = T.grad(cost, [w,b])

# Compile expressions to functions
train = theano.function(
            inputs=[x,y],
            outputs=[prediction, xent],
            updates=[(w, w-0.01*gw), (b, b-0.01*gb)],
            name = "train")
predict = theano.function(inputs=[x], outputs=prediction,
            name = "predict")
```

theano

NVIDIA CUDA

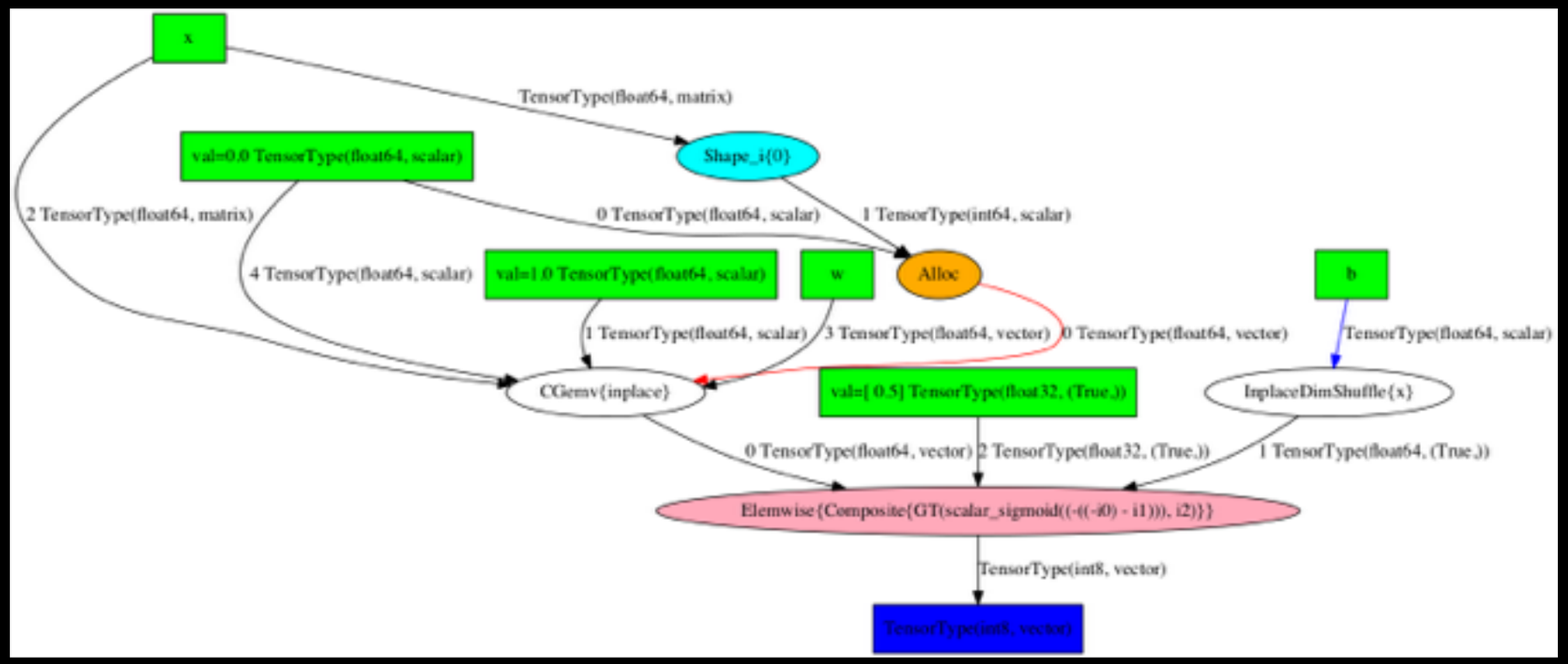
HPC



theano

NVIDIA CUDA

HPC



theano

NVIDIA CUDA

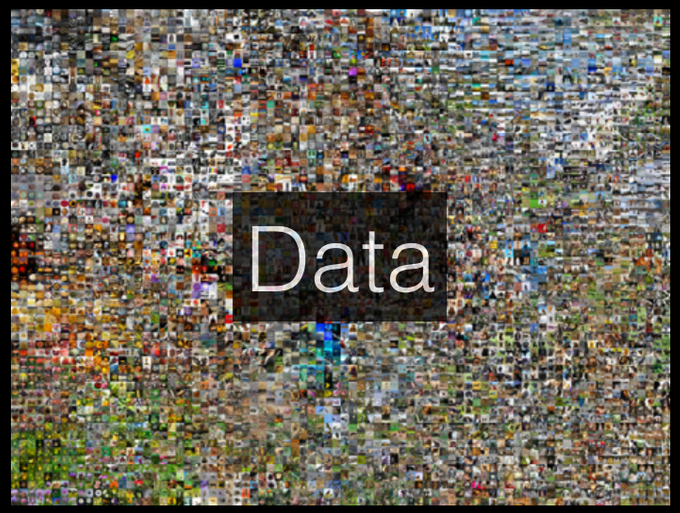
HPC

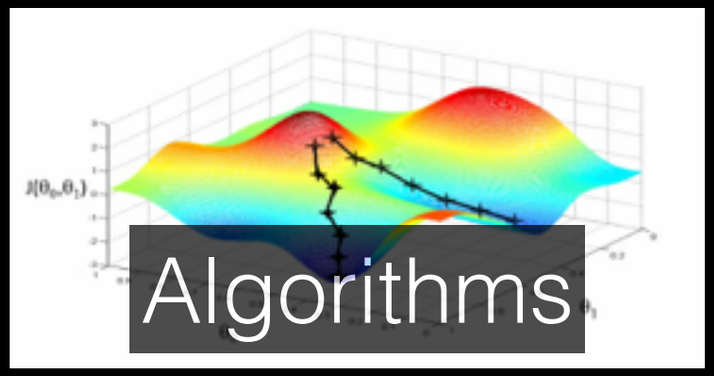

```
# Parameters
learning_rate = 0.01
training_epochs = 25
batch_size = 100
display_step = 1

# tf Graph Input
x = tf.placeholder(tf.float32, [None, 784]) # mnist data image of shape 28*28=784
y = tf.placeholder(tf.float32, [None, 10]) # 0-9 digits recognition => 10 classes

# Set model weights
W = tf.Variable(tf.zeros([784, 10]))
b = tf.Variable(tf.zeros([10]))

# Construct model
pred = tf.nn.softmax(tf.matmul(x, W) + b) # Softmax

# Minimize error using cross entropy
cost = tf.reduce_mean(-tf.reduce_sum(y*tf.log(pred), reduction_indices=1))
# Gradient Descent
optimizer = tf.train.GradientDescentOptimizer(learning_rate).minimize(cost)

# Initializing the variables
init = tf.initialize_all_variables()
```

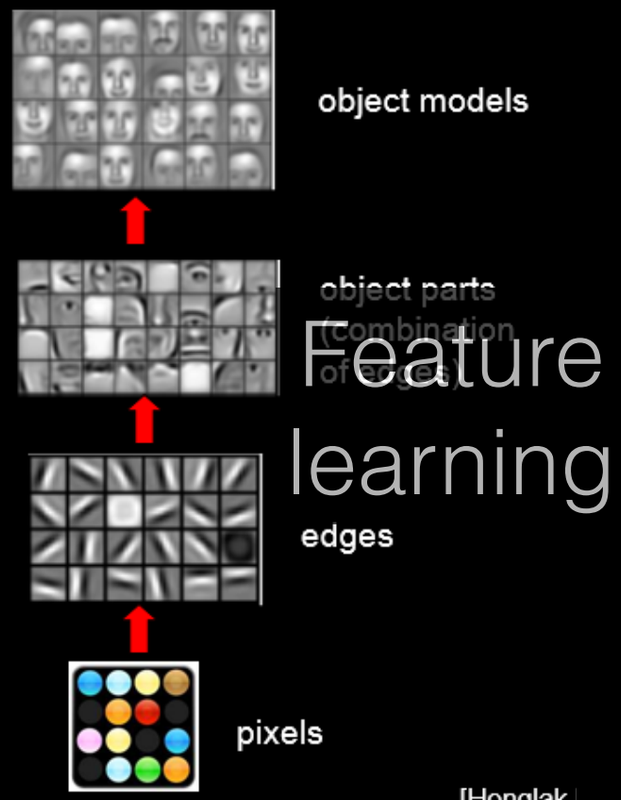
HPC


```
train_y_ohe = one_hot_encode_object_array(train_y)
test_y_ohe = one_hot_encode_object_array(test_y)

model = Sequential()
model.add(Dense(16, input_shape=(4,)))
model.add(Activation('sigmoid'))
model.add(Dense(3))
model.add(Activation('softmax'))
model.compile(loss='categorical_crossentropy', metrics=['accuracy'], optimizer='

# Actual modelling
model.fit(train_X, train_y_ohe, verbose=0, batch_size=1)
score, accuracy = model.evaluate(test_X, test_y_ohe, batch_size=16, verbose=0)
```

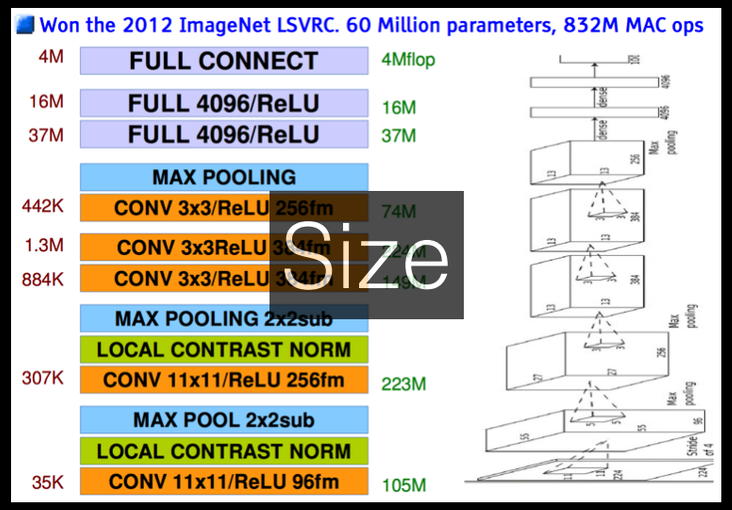
Deep learning recipe



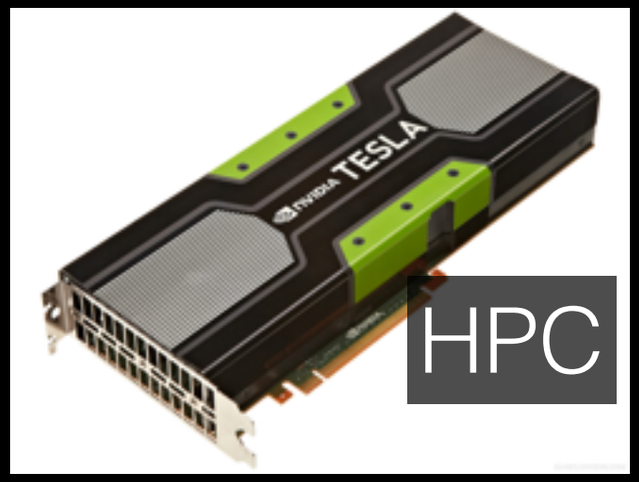




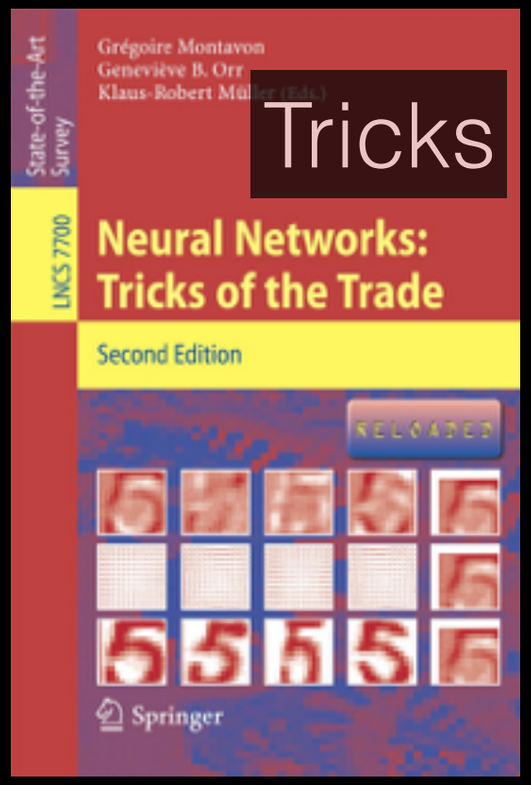
Feature learning



Size



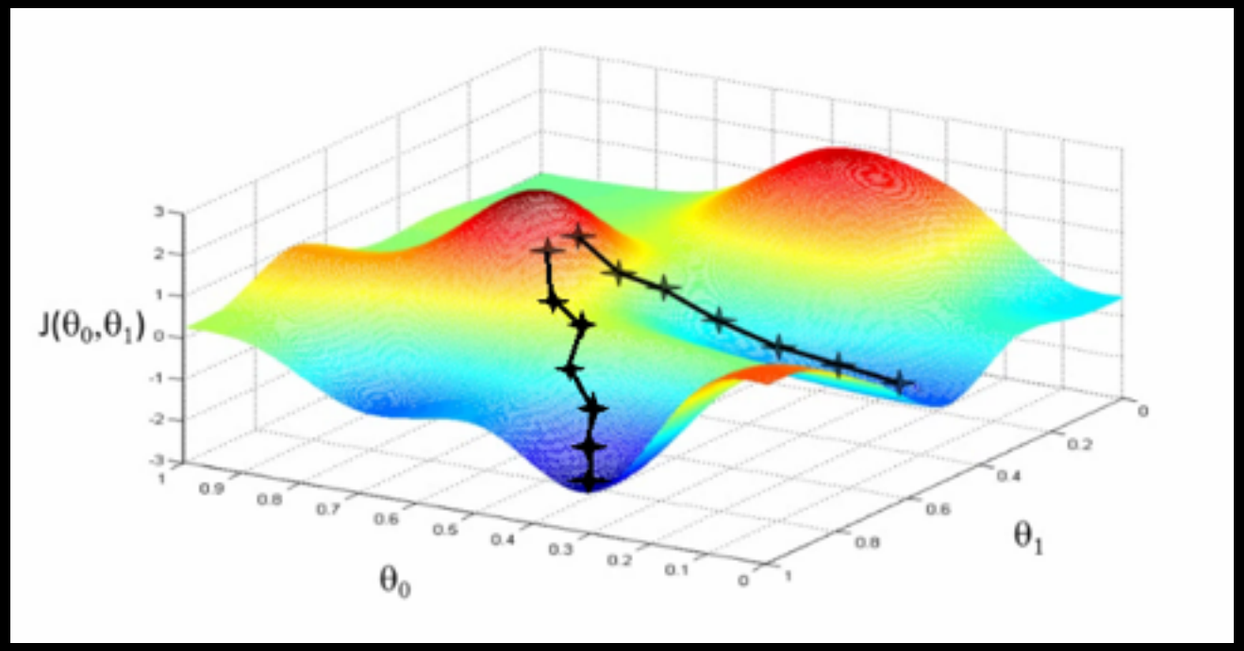
HPC



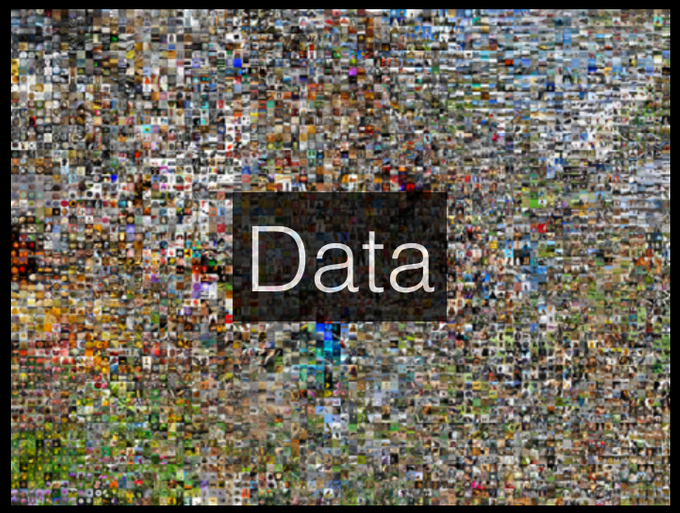
Tricks

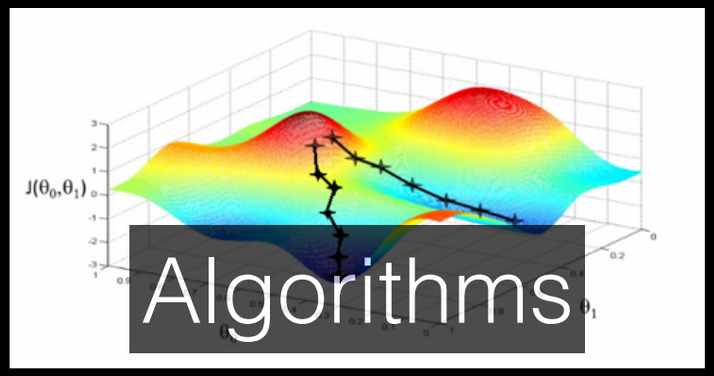
Algorithms

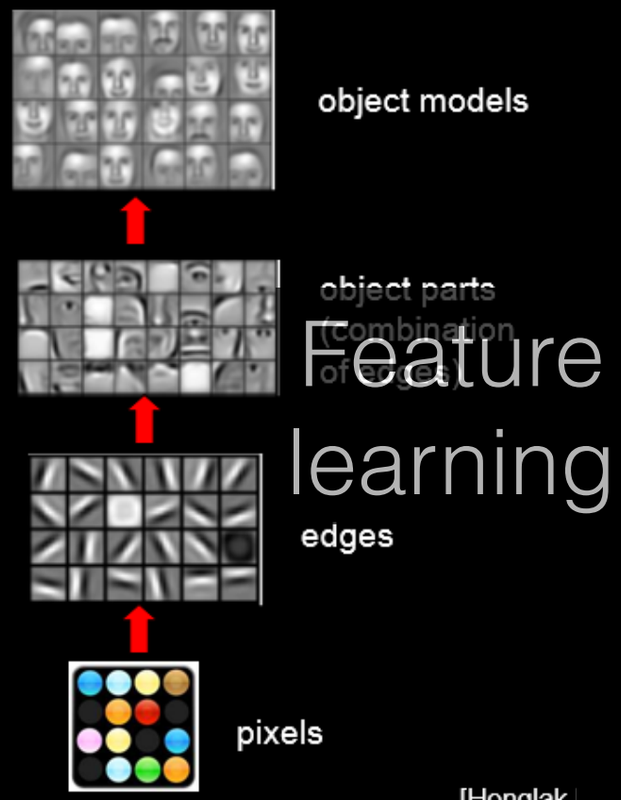
- Backpropagation
- Backpropagation through time
- Online learning (stochastic gradient descent)
- Softmax (hierarchical)

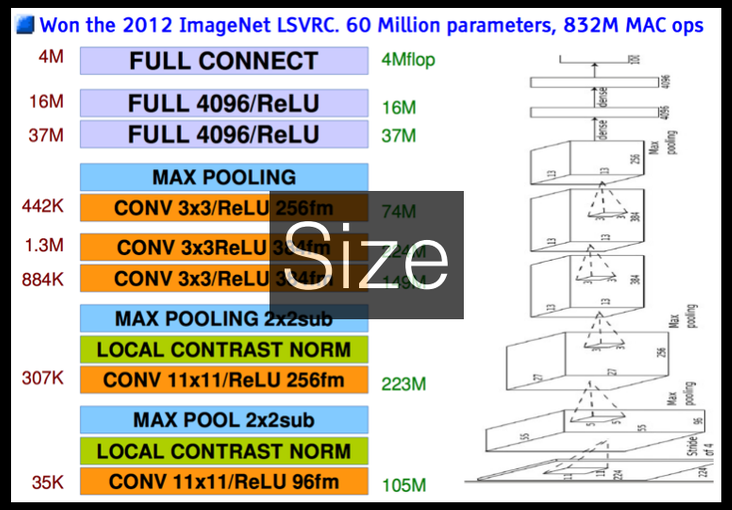


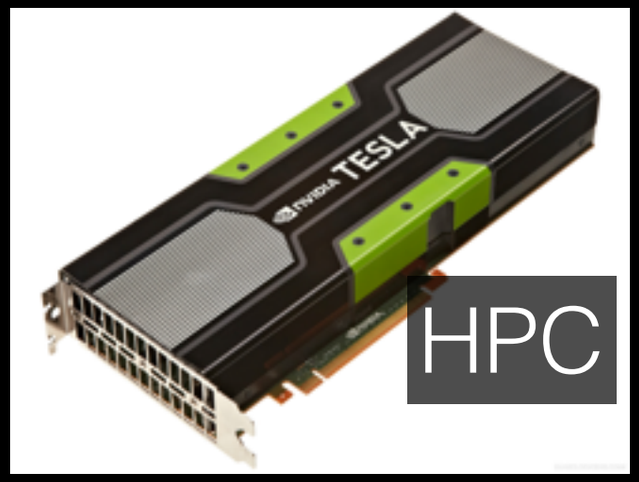
Deep learning recipe

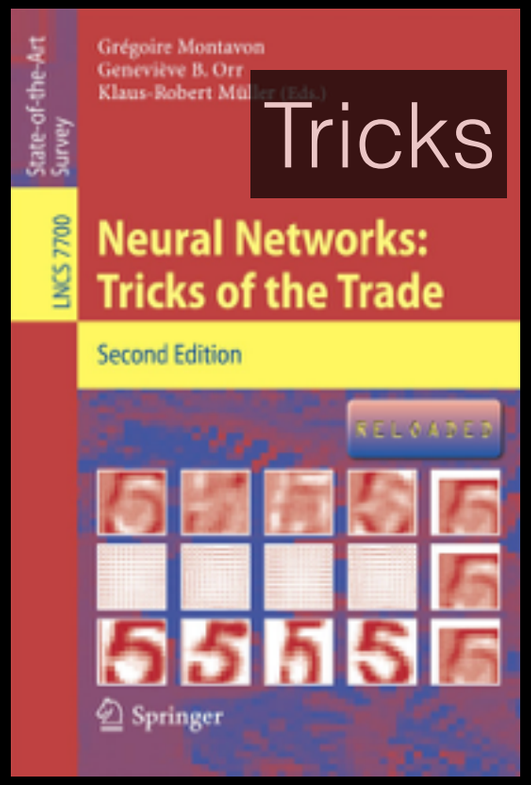






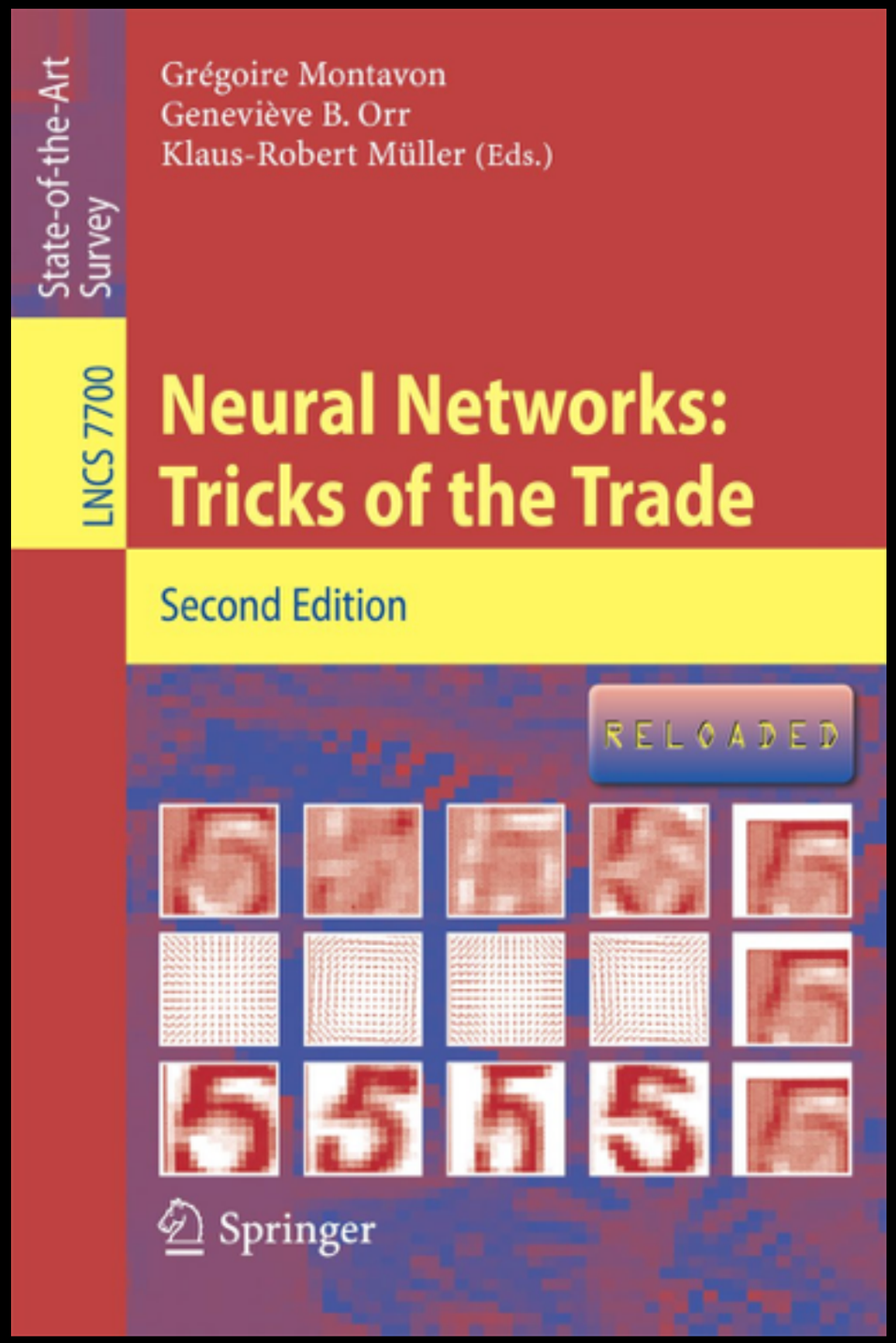




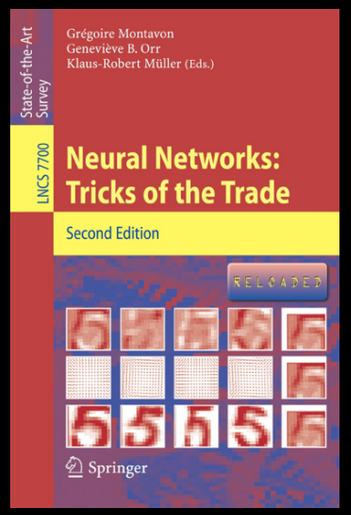


Tricks

- DL is mainly an engineering problem
- DL networks are hard to train
- Several tricks product of years of experience



Tricks

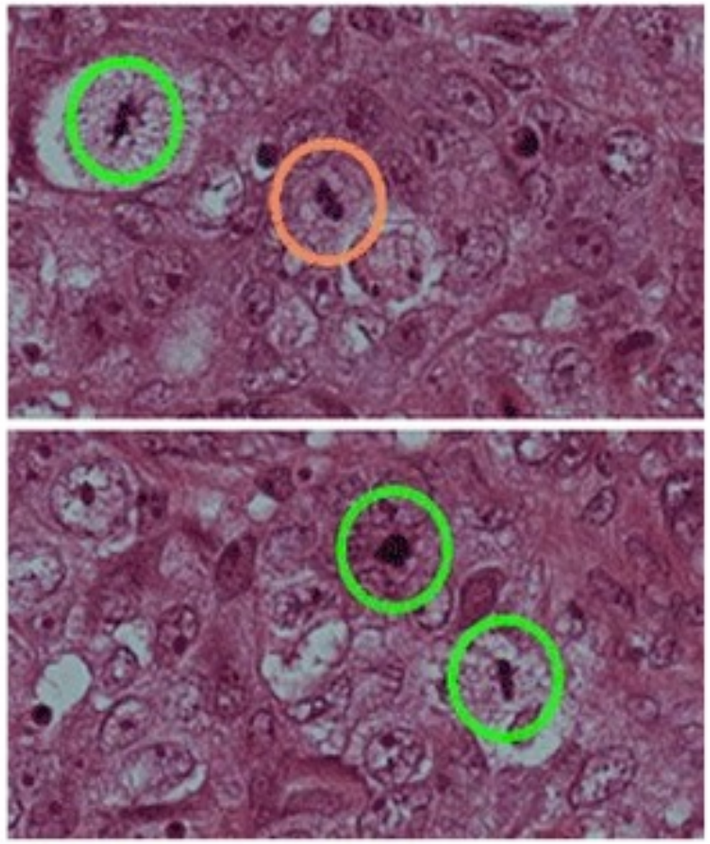


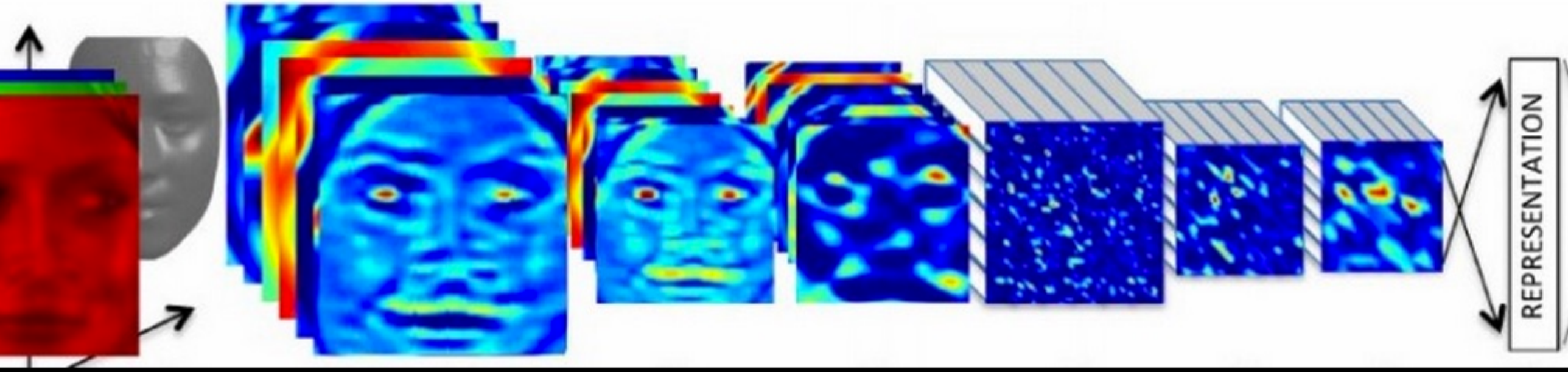
- DL is mainly an engineering problem
- DL networks are hard to train
- Several tricks product of years of experience

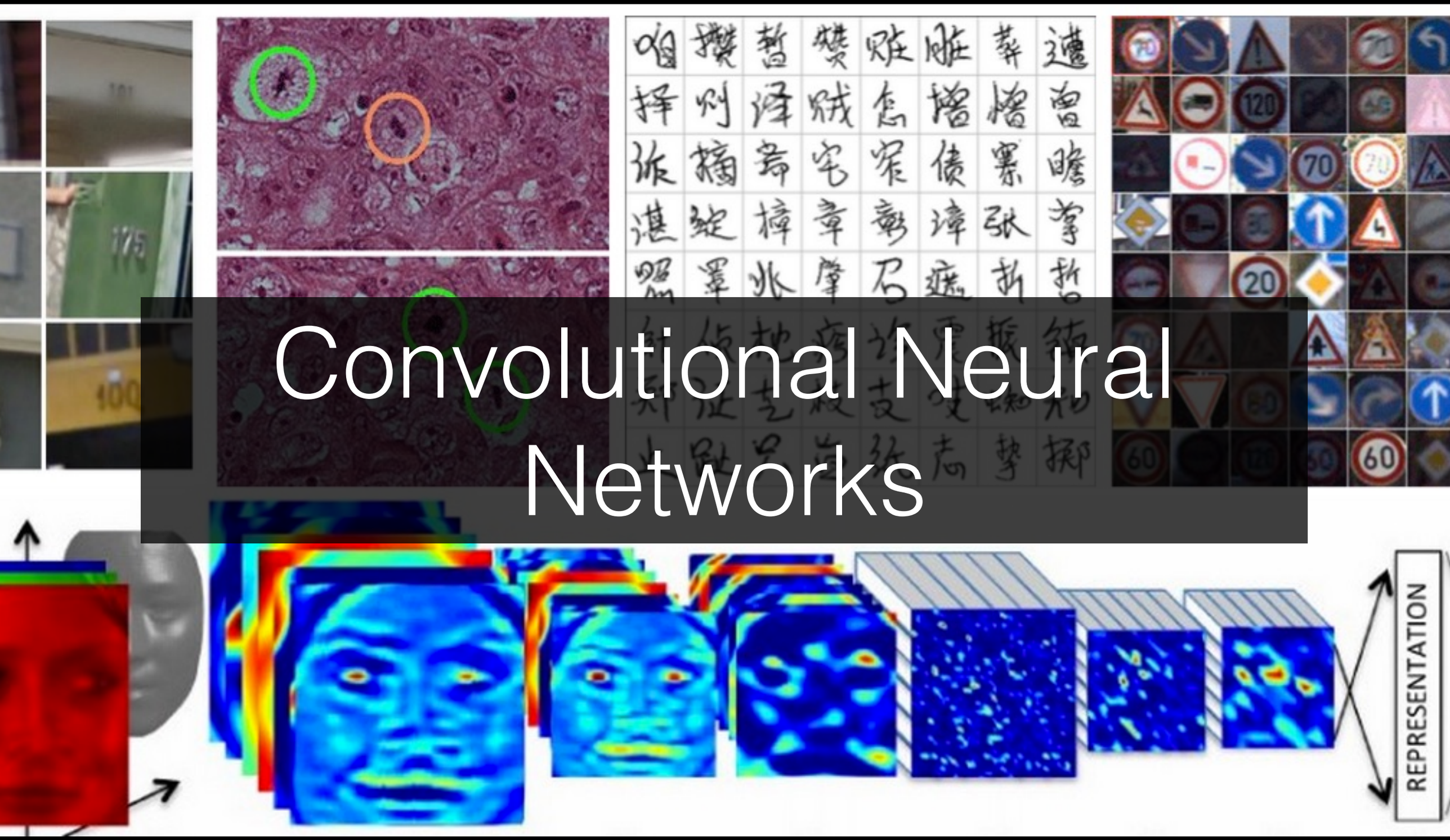
- Layer-wise training
- RELU units
- Dropout
- Adaptive learning rates
- Initialization
- Preprocessing
- Gradient norm clipping

Applications

- Computer vision:
  - Image: annotation, detection, segmentation, captioning
  - Video: object tracking, action recognition, segmentation
- Speech recognition and synthesis
- Text: language modeling, word/text representation, text classification, translation
- Biomedical image analysis

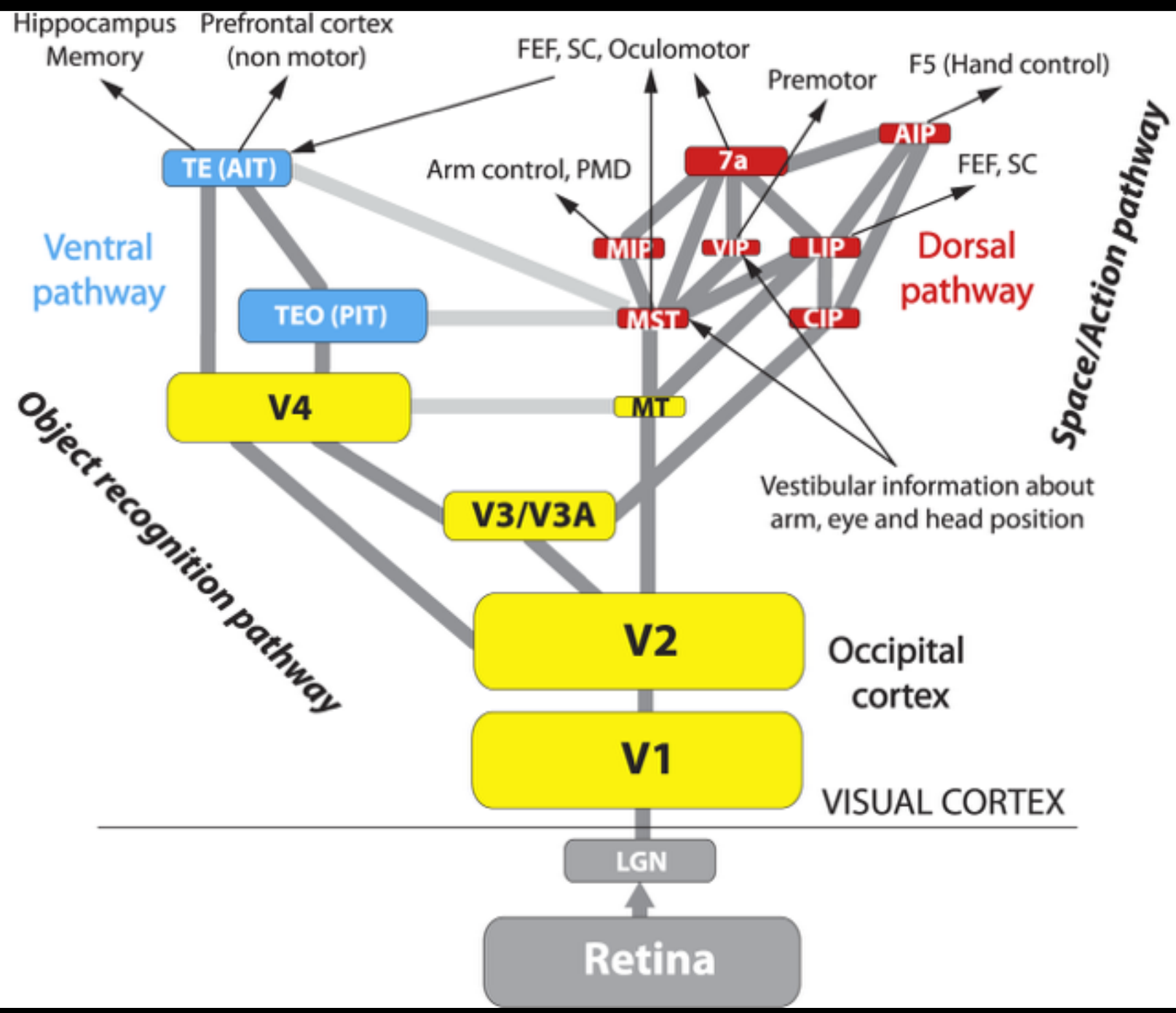


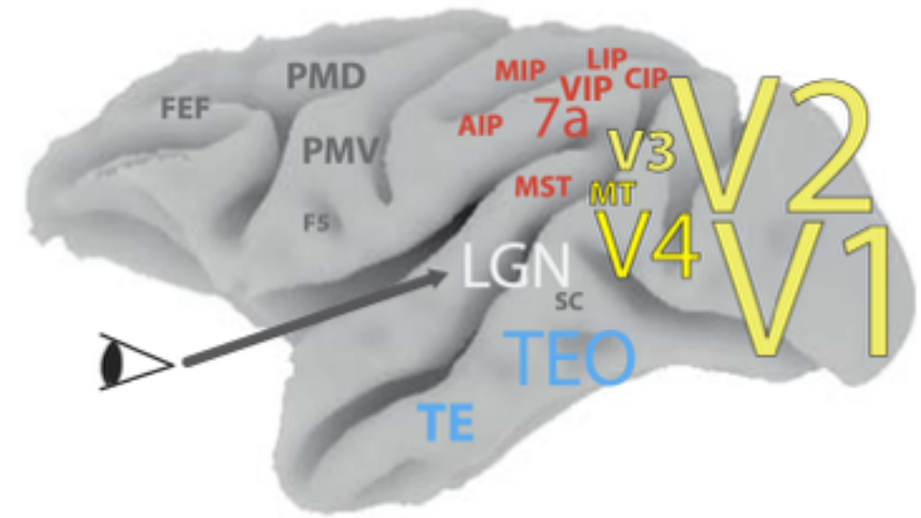

REPRESENTATION



Convolutional Neural Networks

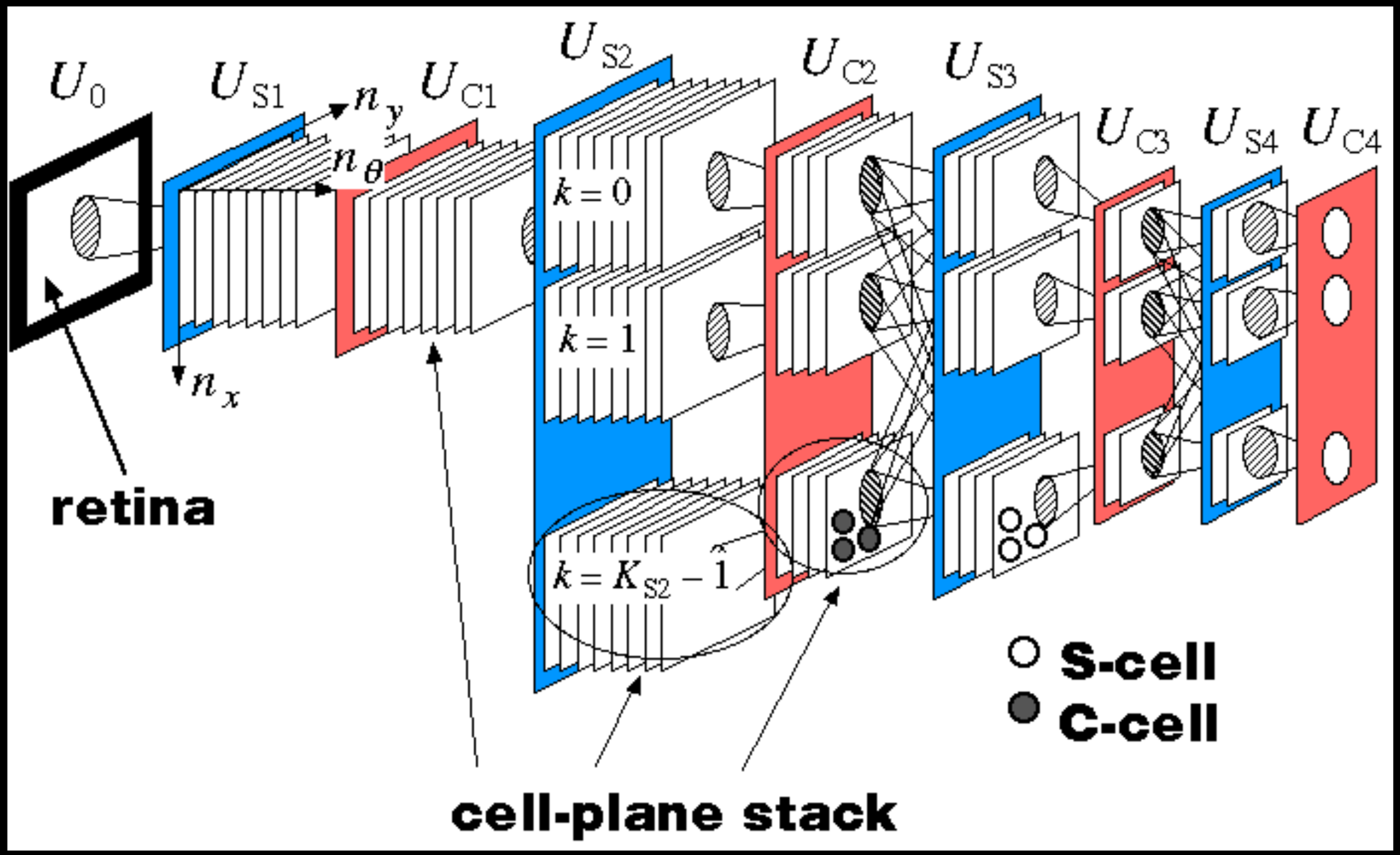
Visual Cortex





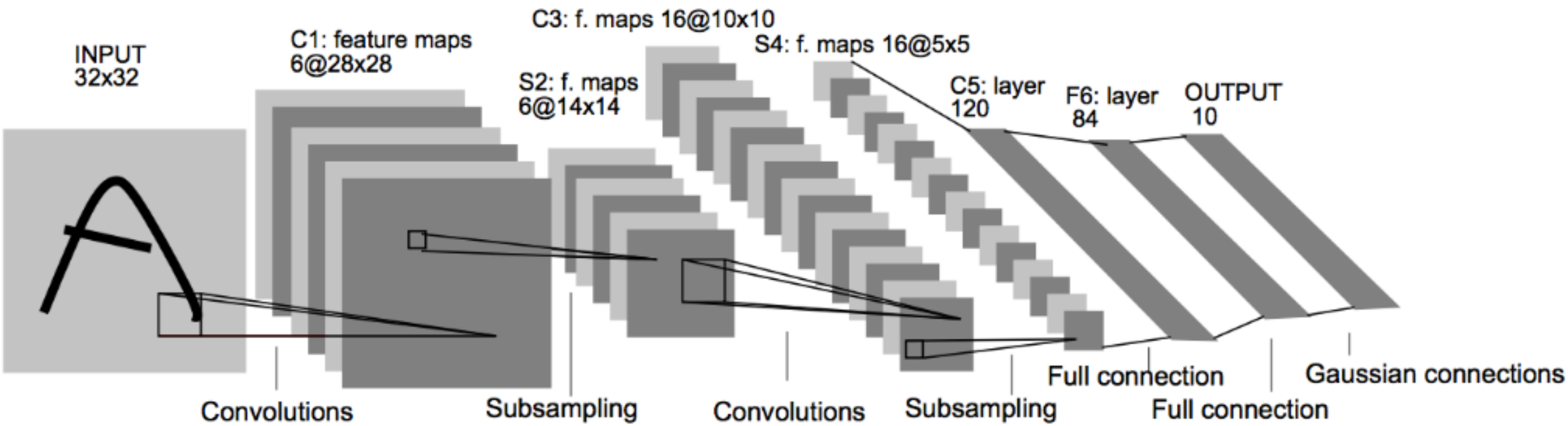
Neocognitron

(Fukushima, 1980)

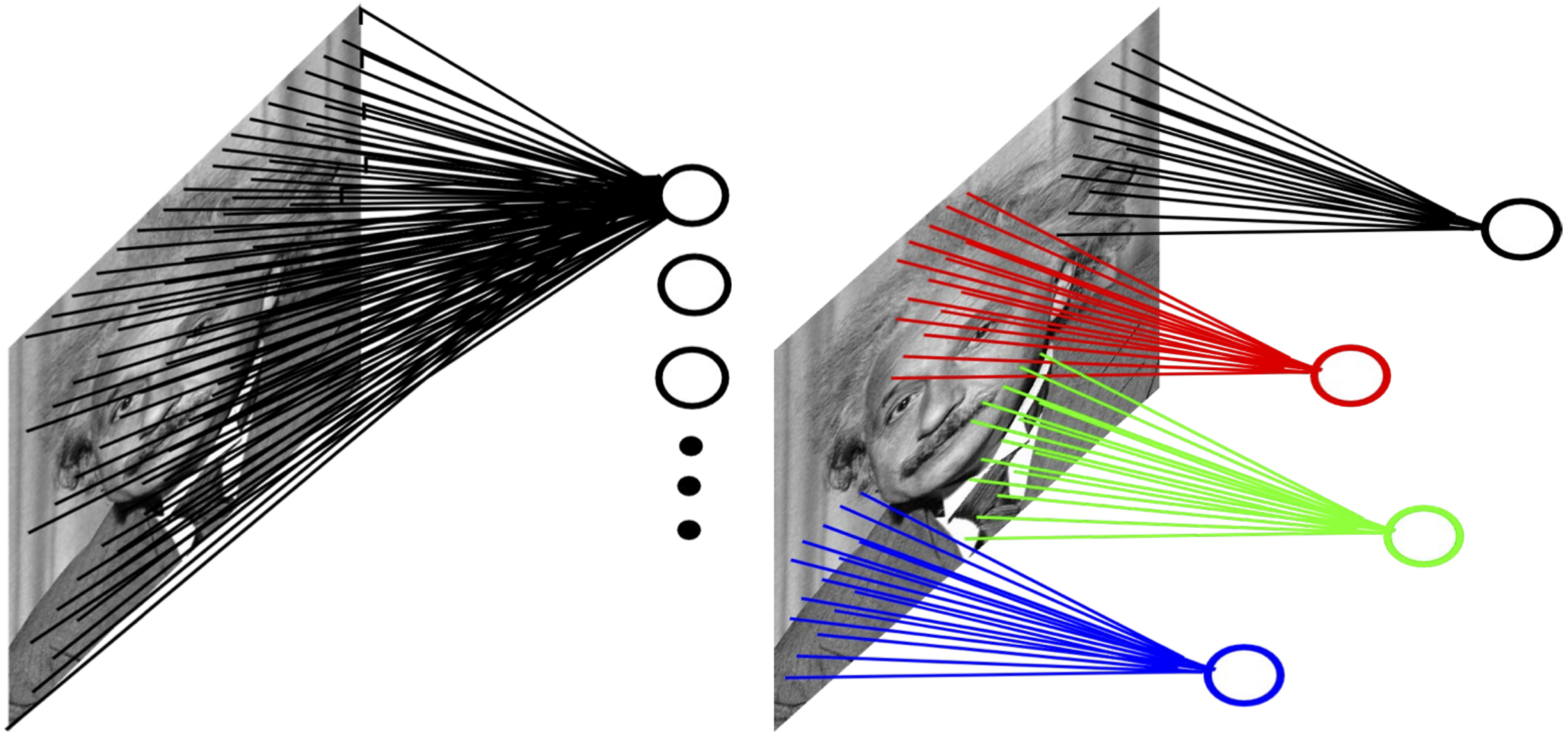


LeNet

(LeCun, 1998)



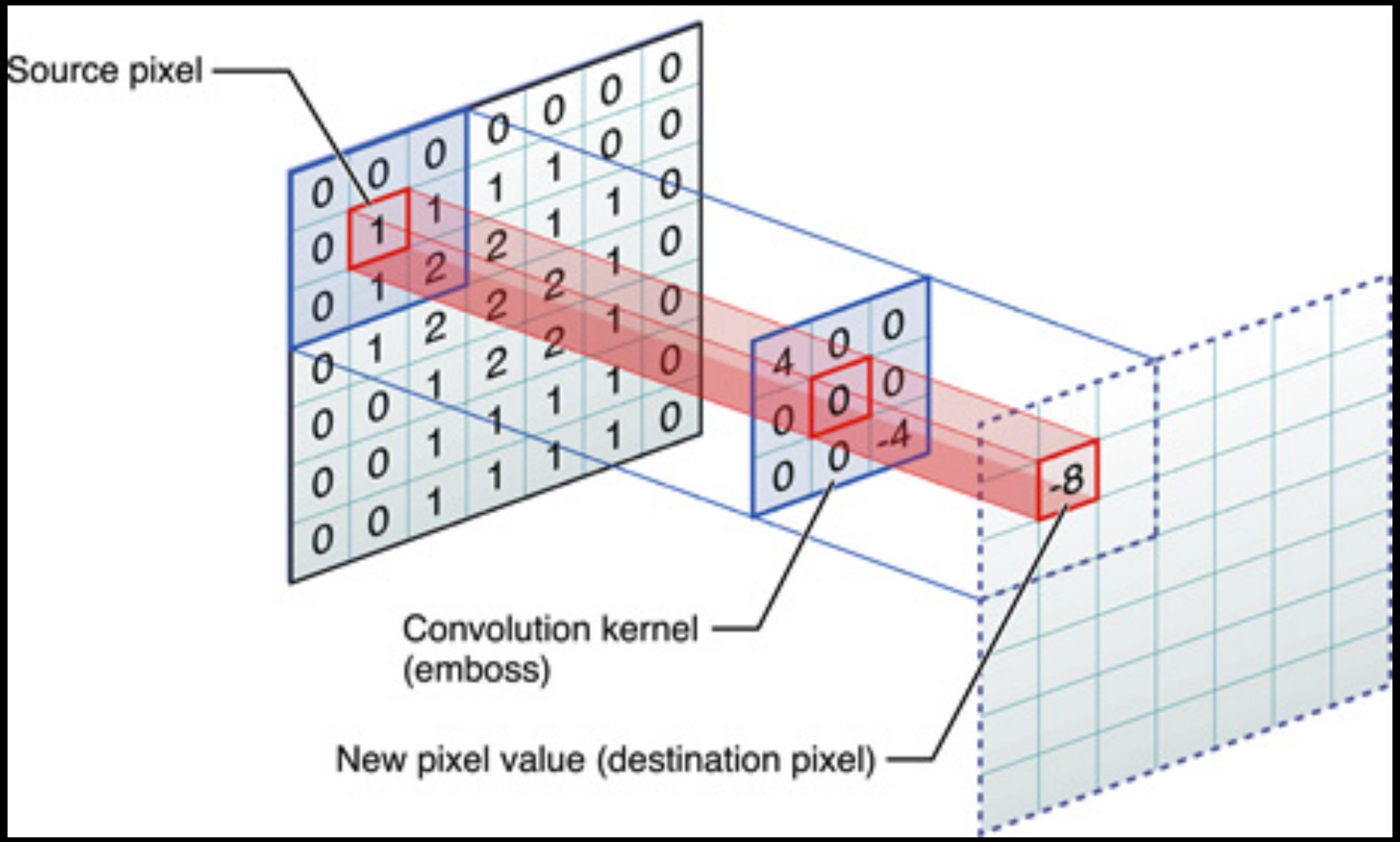
Convolution



(source: ICML2013 Deep Learning Tutorial, Yan LeCun et al.)

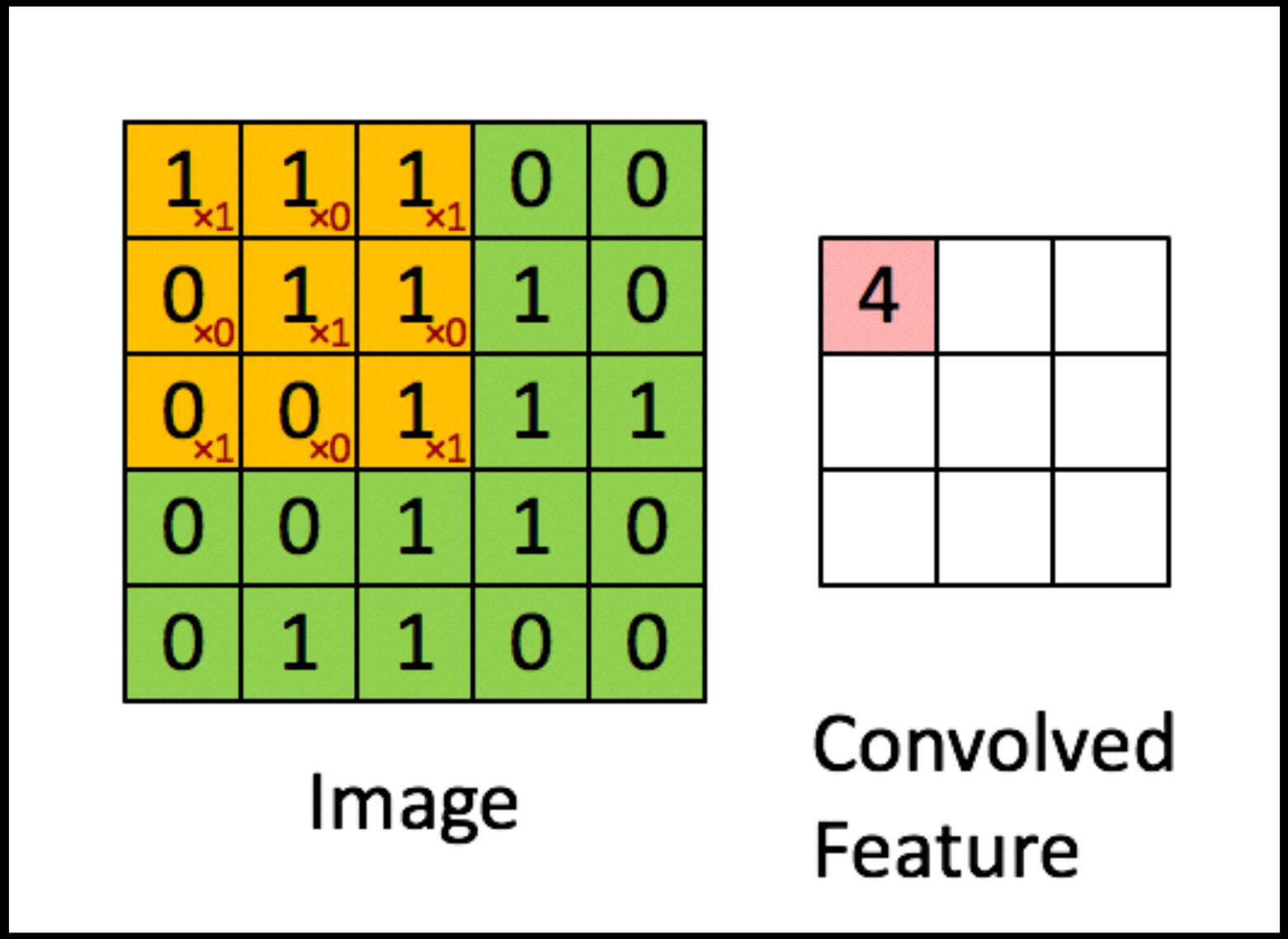
(sources: ICML2013 Deep Learning Tutorial, Yan LeCun et al.
Feature extraction using convolution, Stanford Deep Learning Wiki)

Convolution



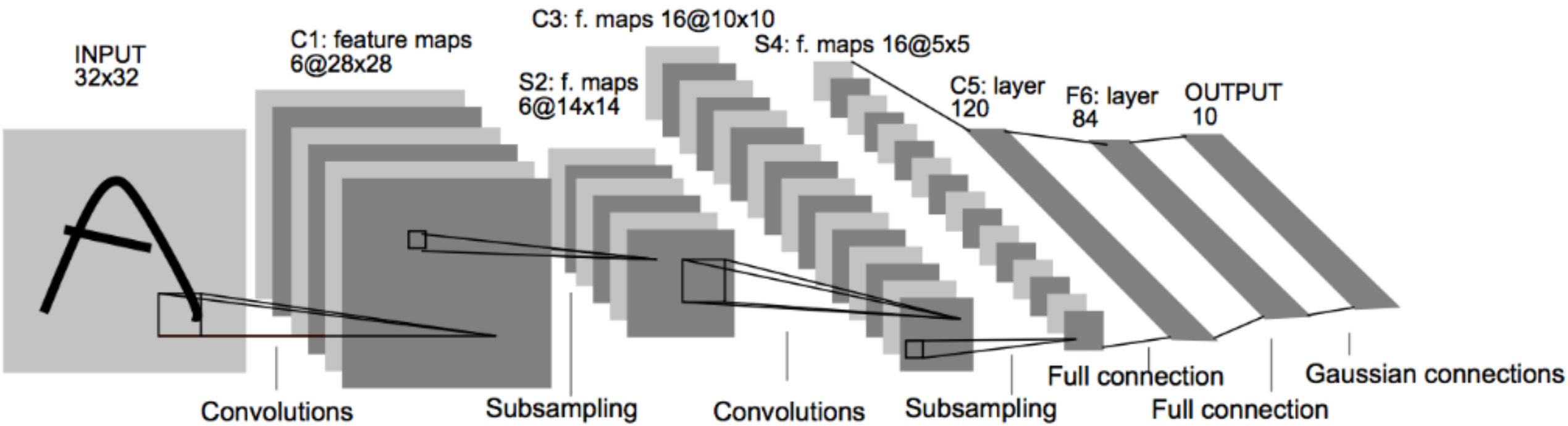
(sources: ICML2013 Deep Learning Tutorial, Yan LeCun et al.
Feature extraction using convolution, Stanford Deep Learning Wiki)

Convolution



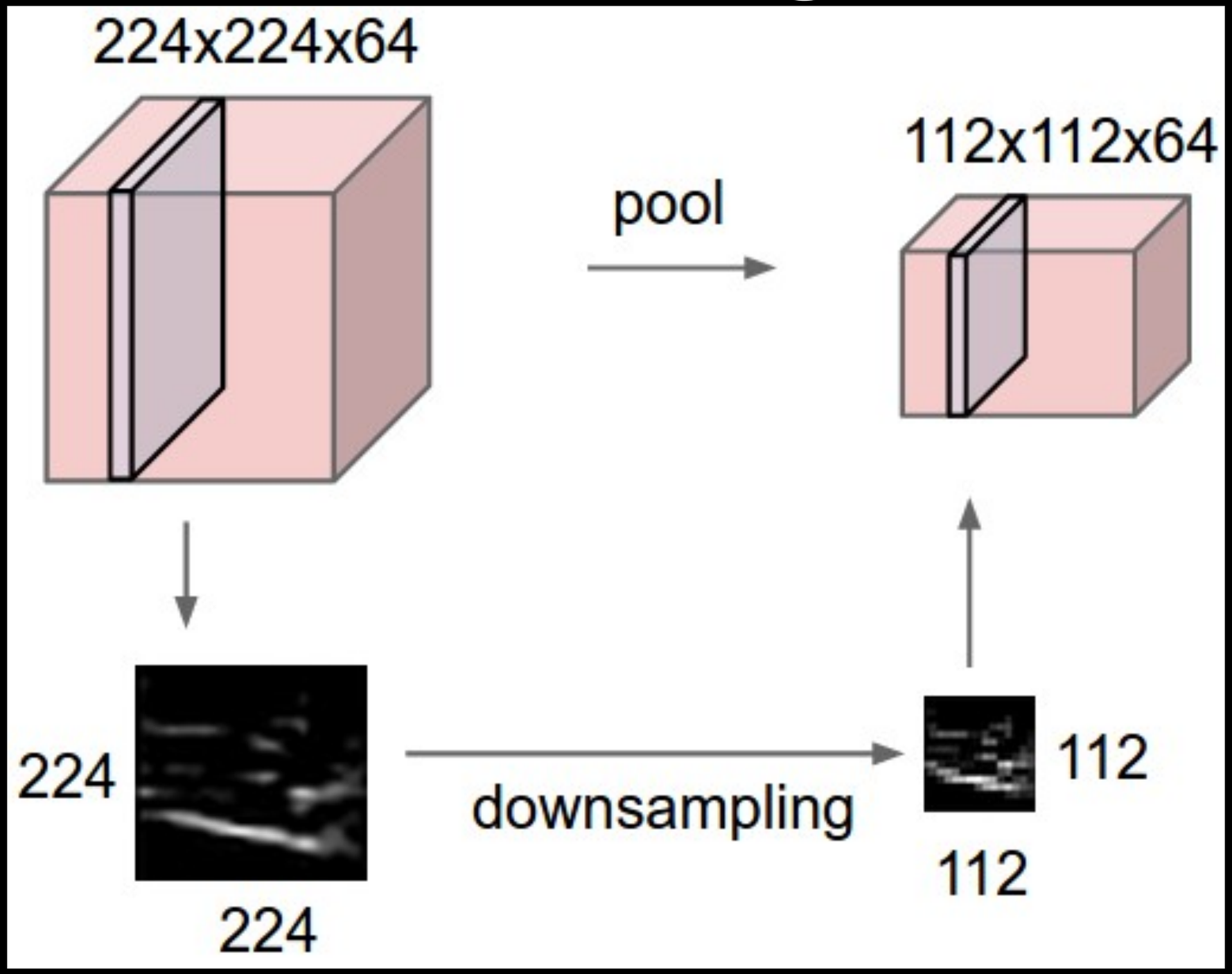
(sources: ICML2013 Deep Learning Tutorial, Yan LeCun et al.
Feature extraction using convolution, Stanford Deep Learning Wiki)

Convolution



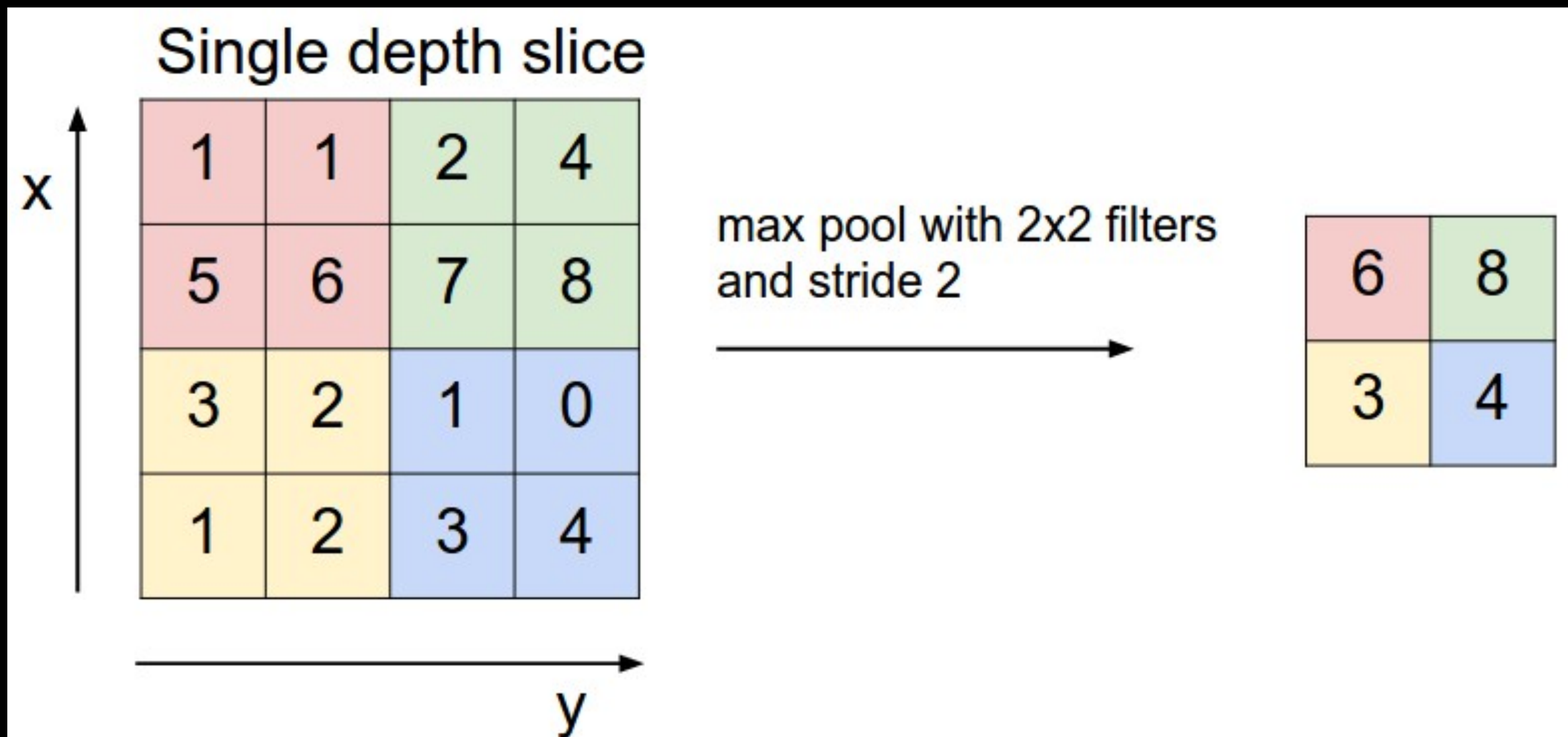
(sources: ICML2013 Deep Learning Tutorial, Yan LeCun et al.
Feature extraction using convolution, Stanford Deep Learning Wiki)

Pooling



(source: Karpathy, CS231n Convolutional Neural Networks for Visual Recognition)

Pooling

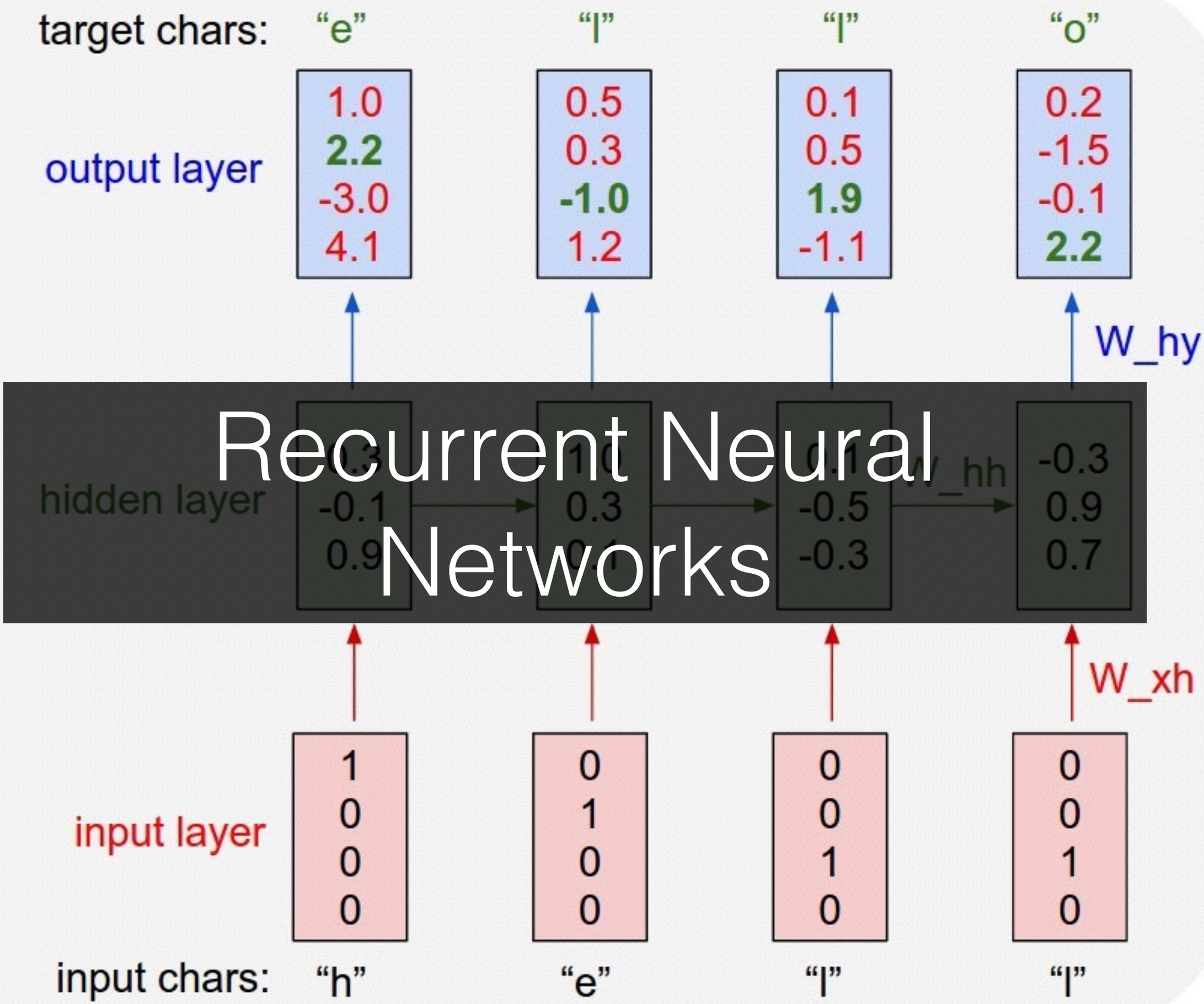


(source: Karpathy, CS231n Convolutional Neural Networks for Visual Recognition)

CNN in Keras

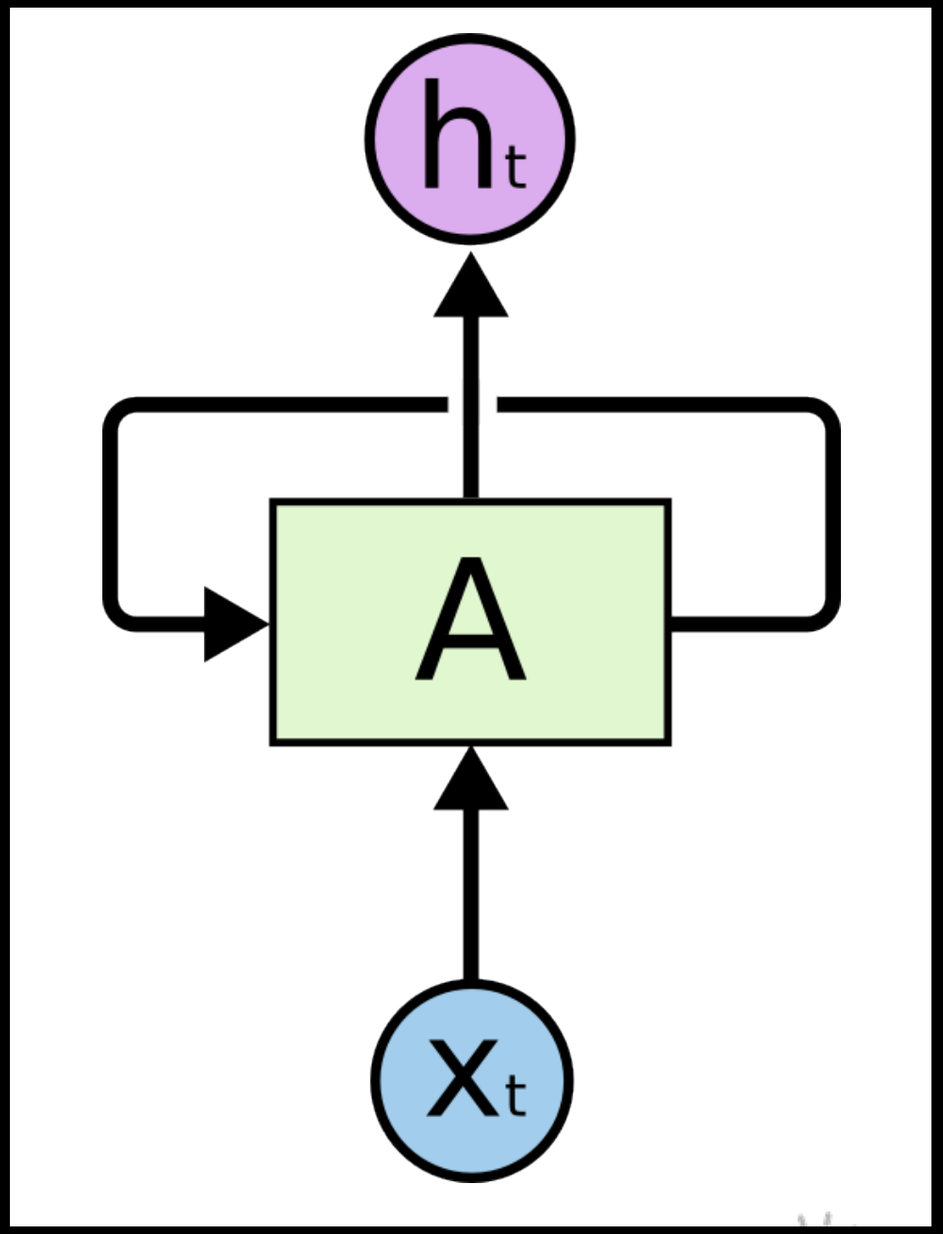
Interactive Demo

Recurrent Neural Networks

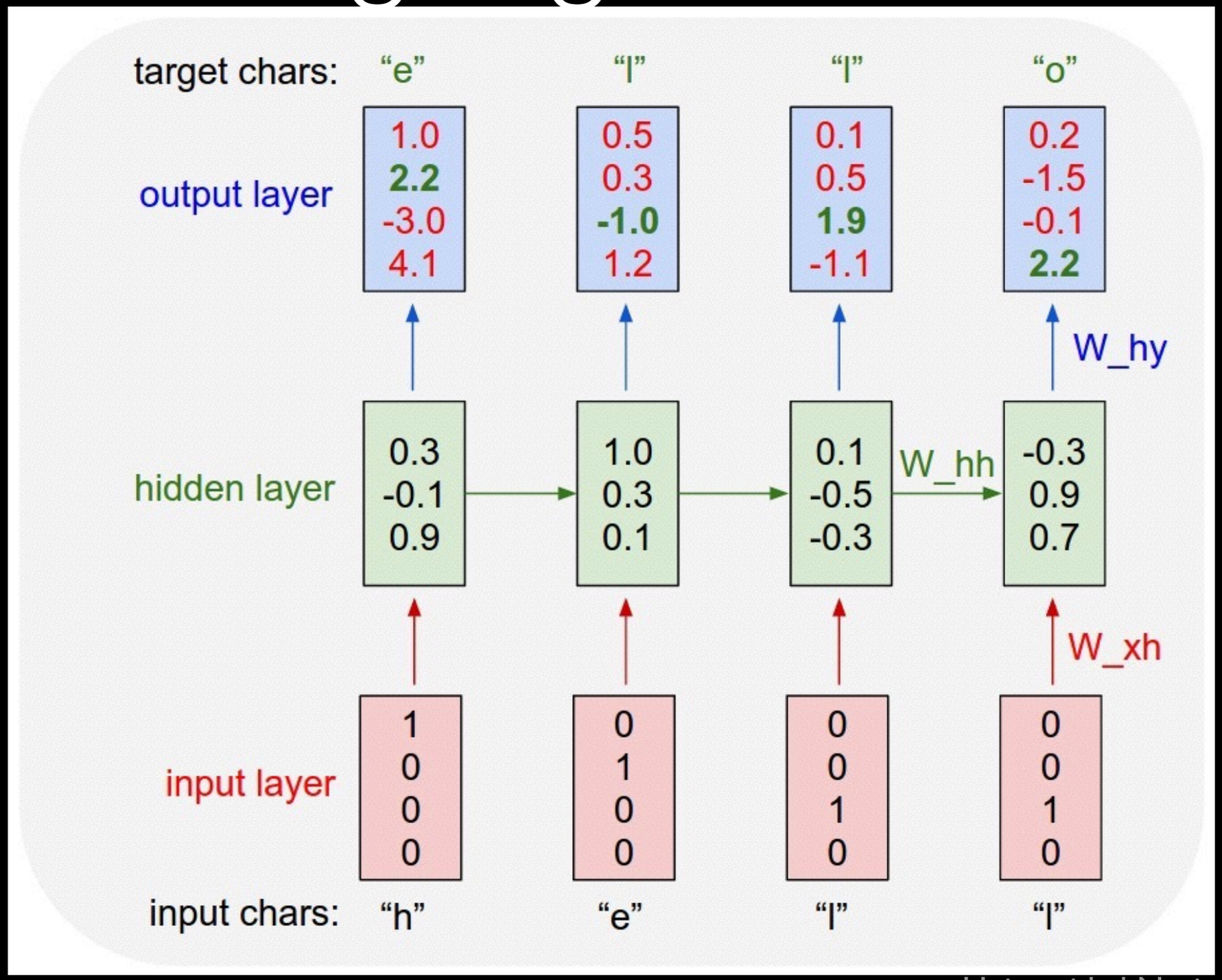


Recurrent NN

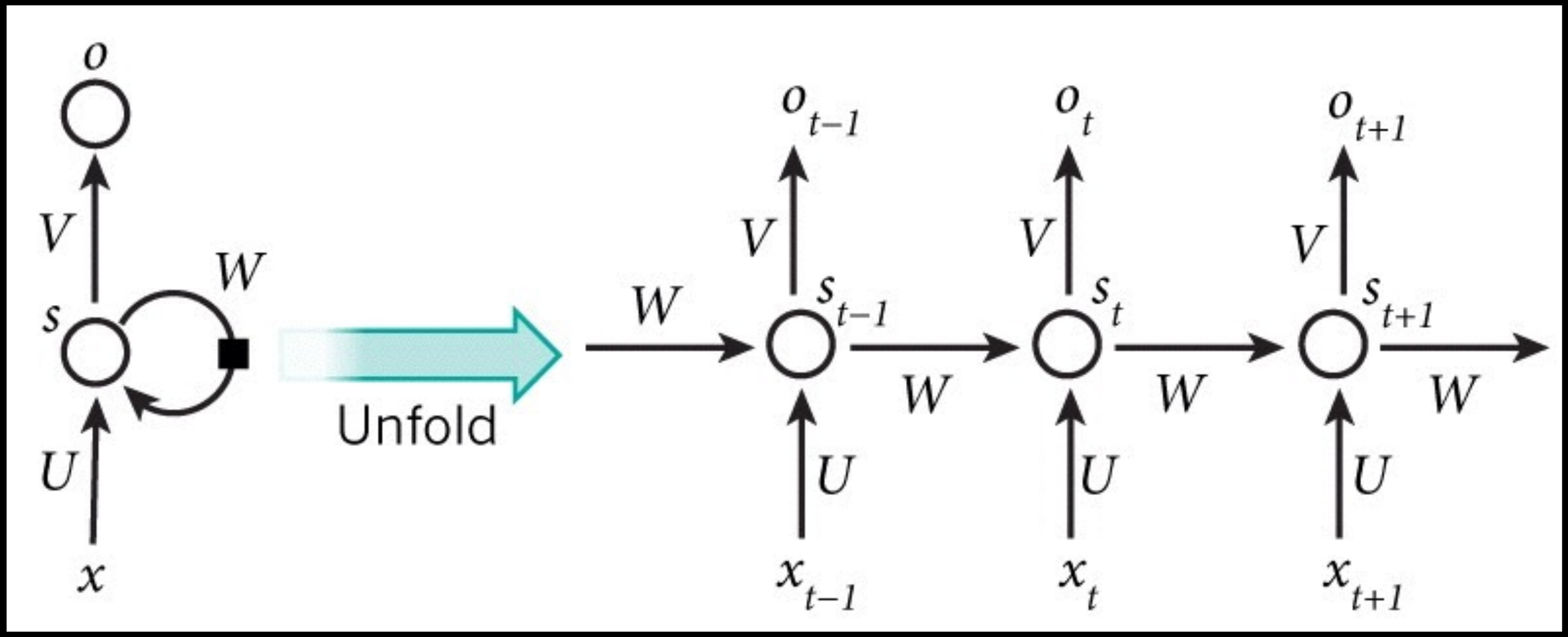
- Neural networks with memory
- Feed-forward NN: output exclusively depends on the current input
- Recurrent NN: output depends on current and previous states
- This is accomplished through lateral/backward connections which carry information while processing a sequence of inputs



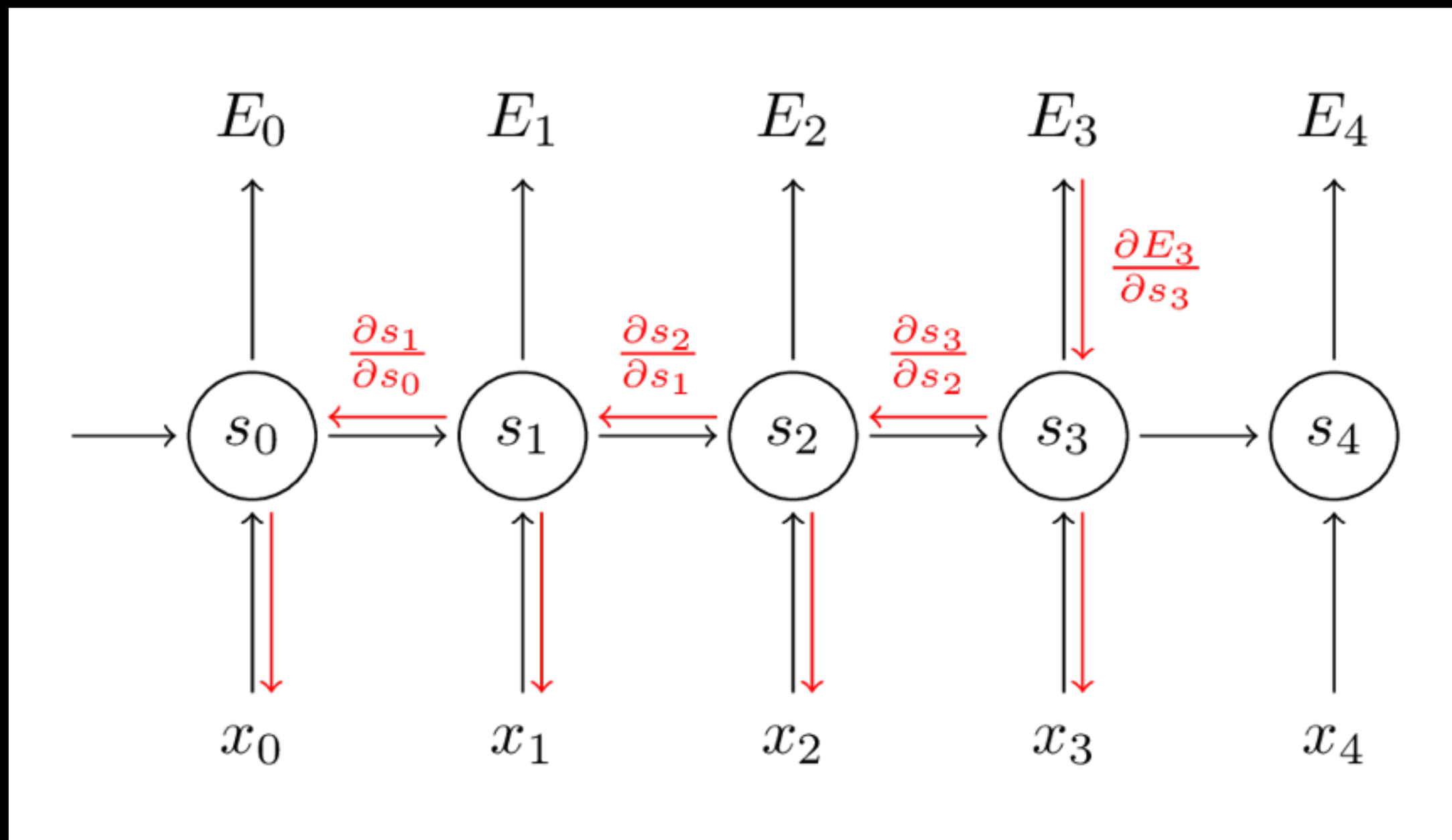
Character-level language model



Network unrolling

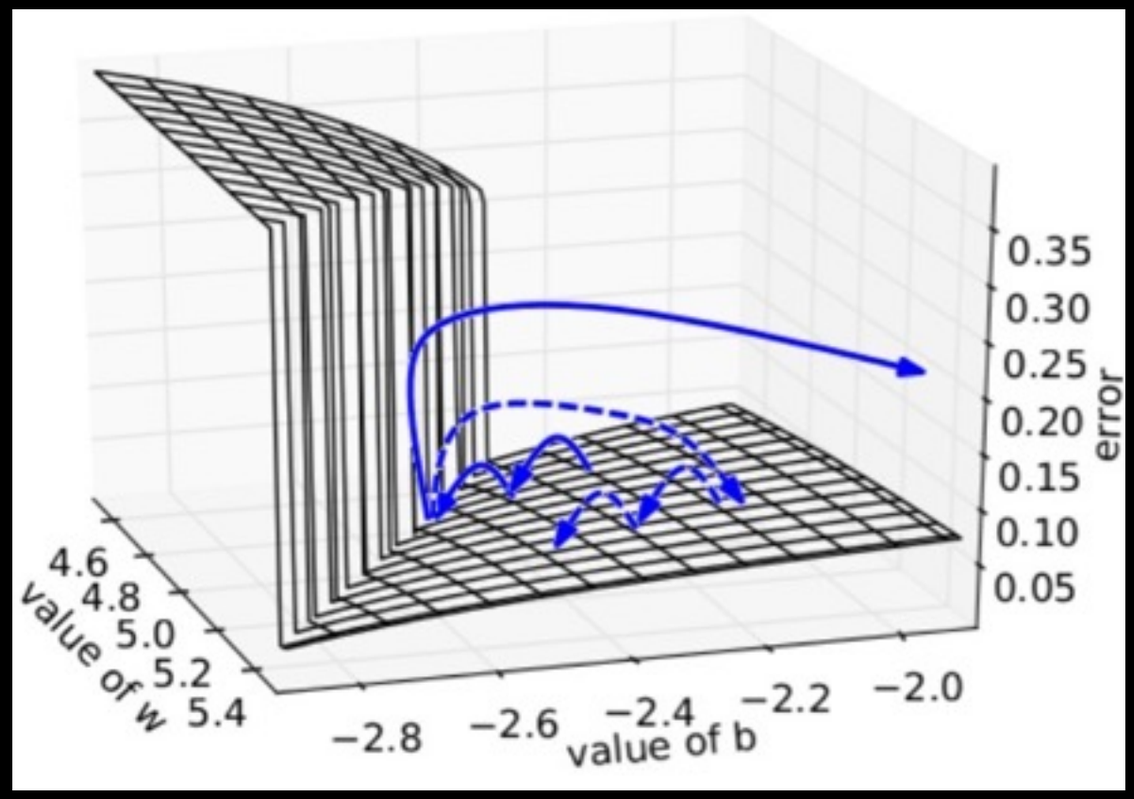


Backpropagation through time (BPTT)

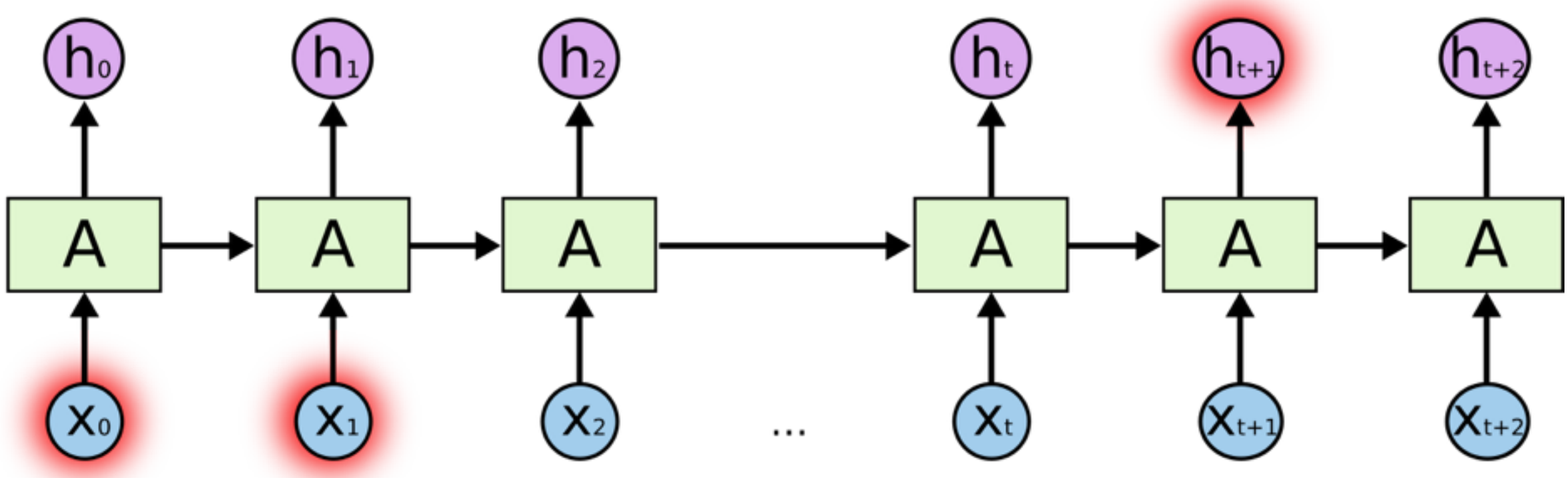


BPTT is hard

- The *vanishing* and the *exploding* gradient problem
- Gradients could vanish (or explode) when propagated several steps back
- This makes difficult to learn long-term dependencies.
- Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. *On the difficulty of training Recurrent Neural Networks*. Proc. of ICML, abs/1211.5063.



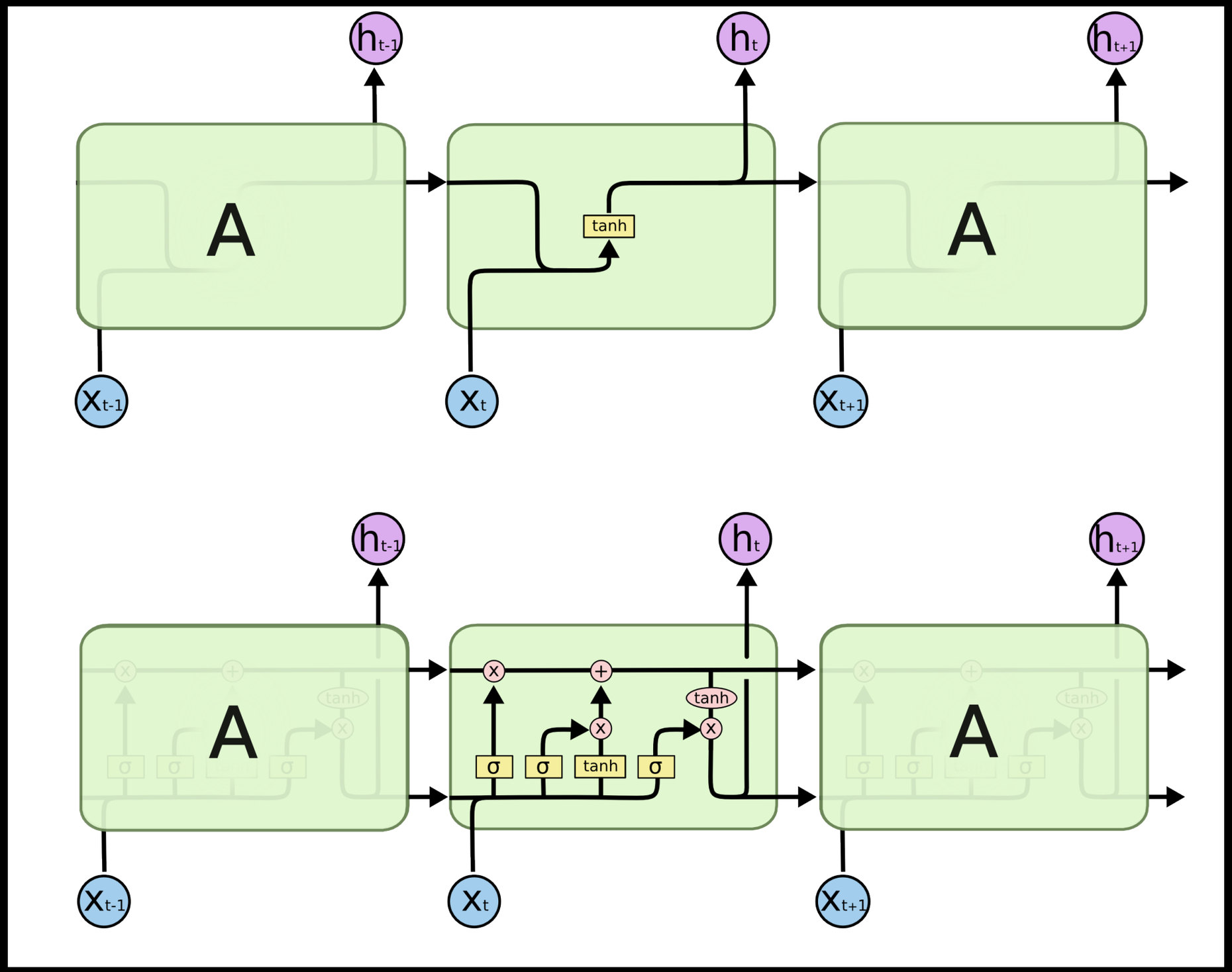
Long term dependencies



Long short-term memory (LSTM)

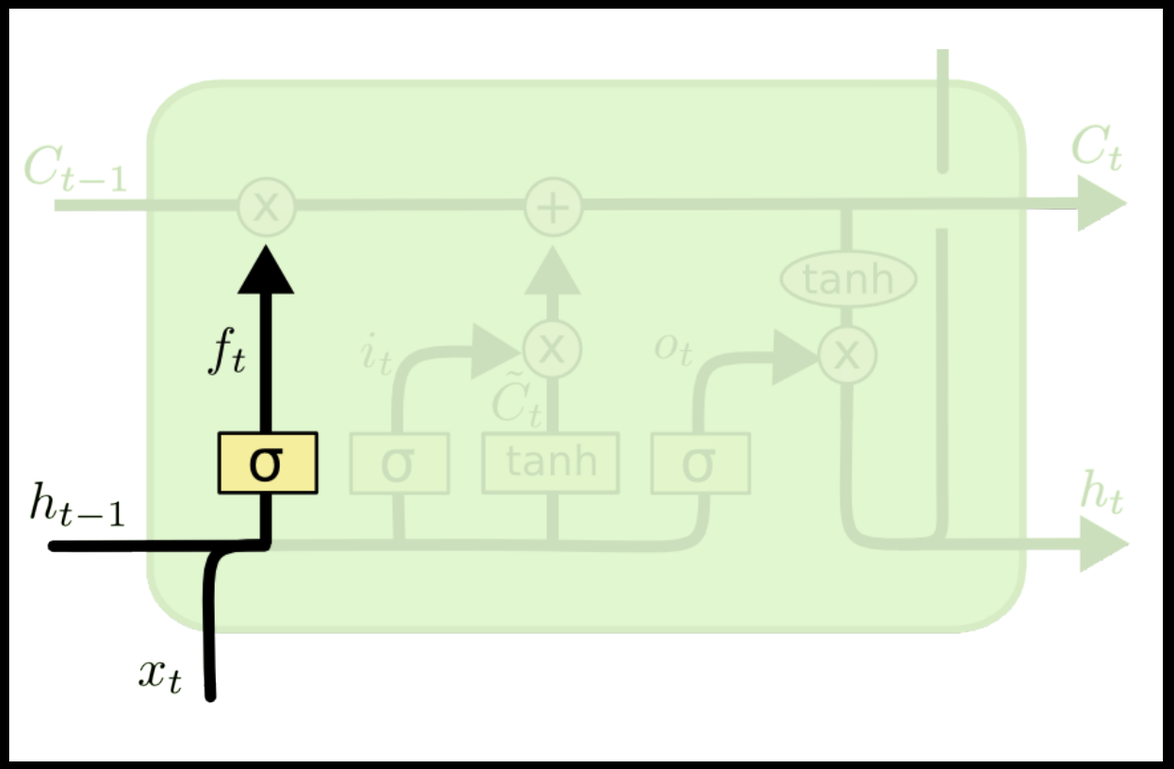
- LSTM networks solve the problem of long-term dependency problem.
- They use gates that allow to keep memory through long sequences and be updated only when required.
- Hochreiter, Sepp, and Jürgen Schmidhuber. "*Long short-term memory*." Neural computation 9, no. 8 (1997): 1735-1780.

Conventional RNN vs LSTM



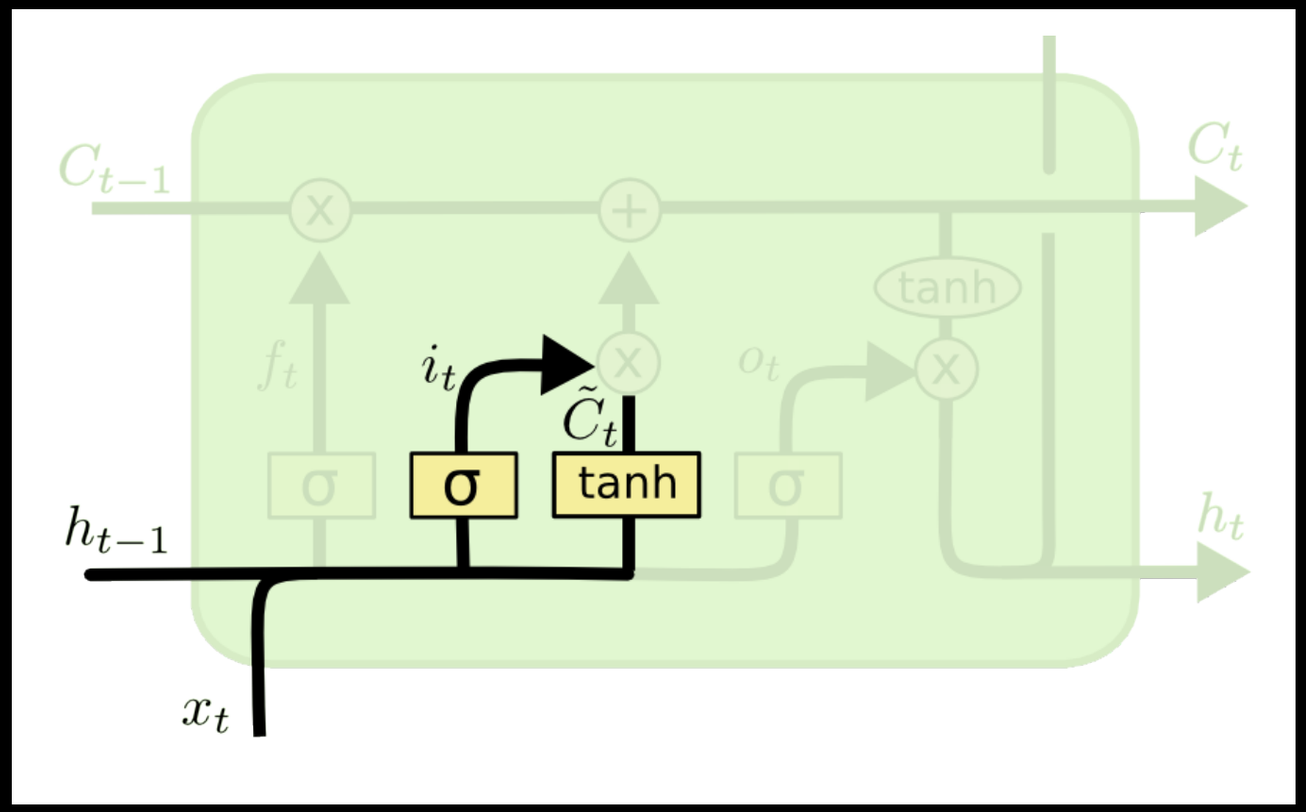
Forget gate

- Controls the flow of the previous internal state C_{t-1}
- $f_t=1 \Rightarrow$ keep previous state
- $f_t=0 \Rightarrow$ forget previous state

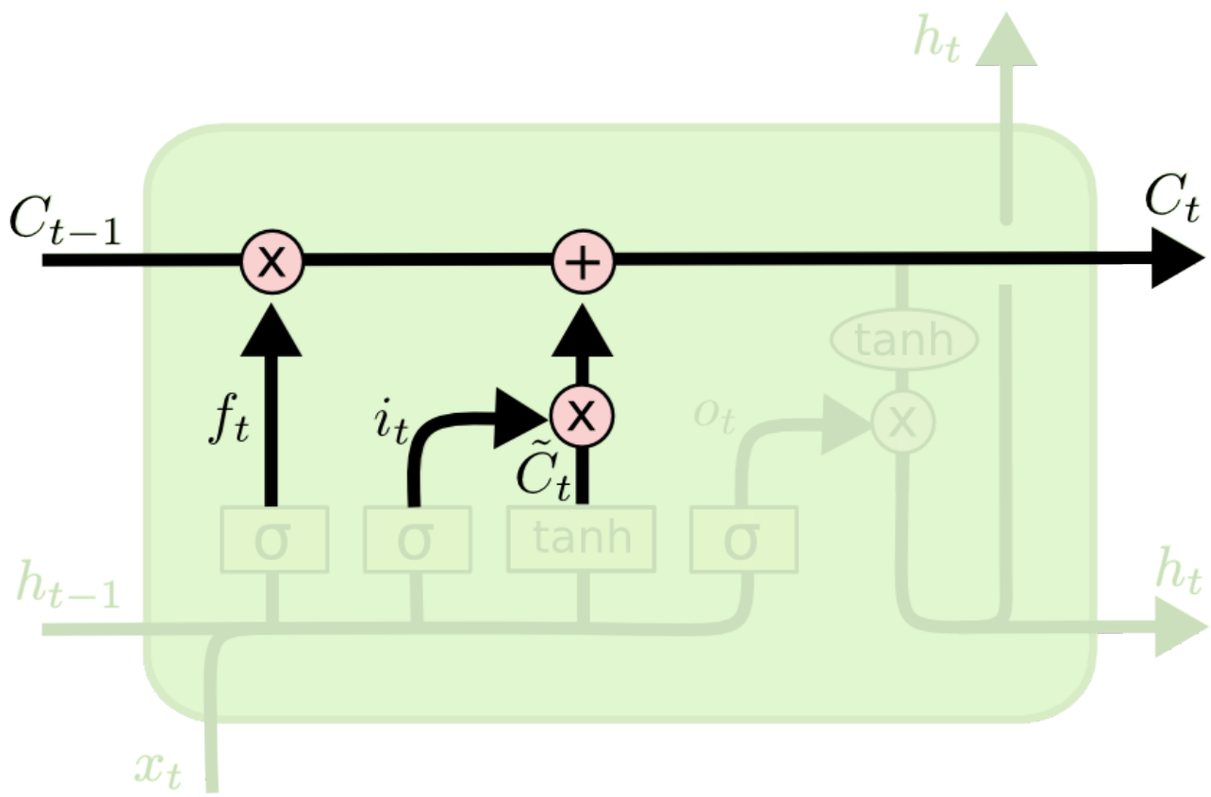


Input gate

- Controls the flow of the input state x_t
- i_t=1 ⇒ take input into account
- i_t=0 ⇒ ignore input

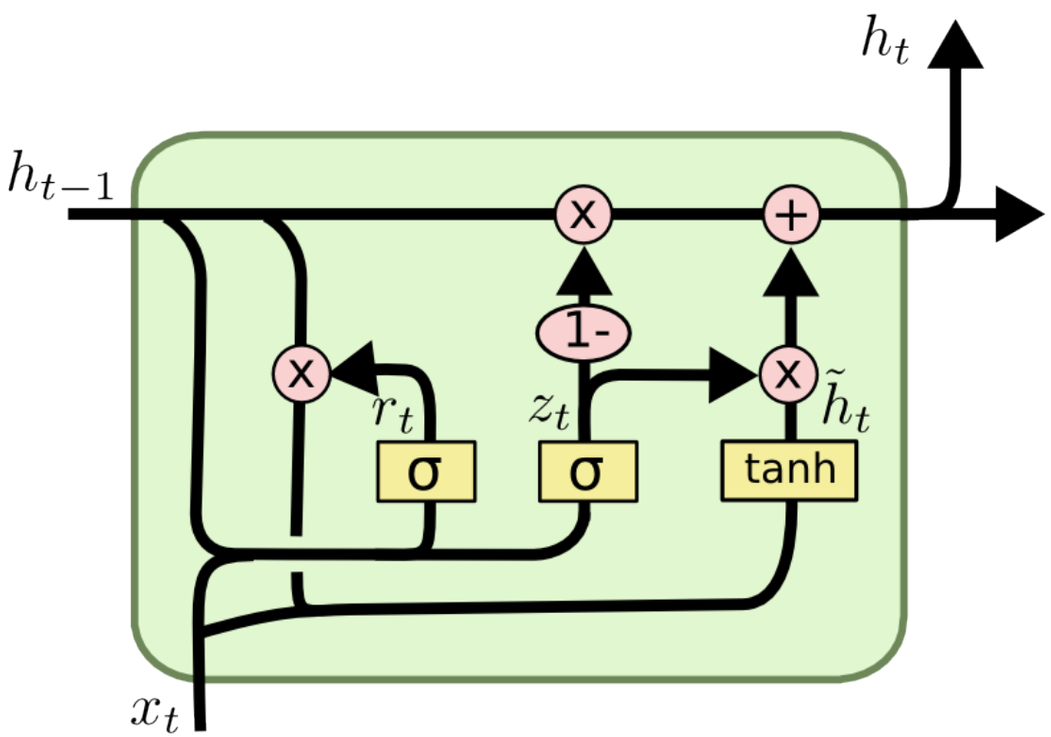


Current state calculation



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Gated recurrent units



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). ***Learning phrase representations using rnn encoder-decoder for statistical machine translation***. arXiv preprint arXiv:1406.1078.

Playing With LSTMs for Language Modeling

Interactive Demo

The Unreasonable Effectiveness of Recurrent Neural Networks

- Famous blog entry from Andrej Karpathy (UofS)
- Character-level language models based on multi-layer LSTMs.
- Data:
  - Shakspare plays
  - Wikipedia
  - LaTeX
  - Linux source code

Algebraic geometry book in LaTeX

Proof. Omitted. □

Lemma 0.1. *Let $\mathcal{C}$ be a set of the construction.*

Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

Lemma 0.2. *This is an integer $\mathcal{Z}$ is injective.*

Proof. See Spaces, Lemma ??. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

be a morphism of algebraic spaces over S and Y.

Proof. Let X be a nonzero scheme of X. Let X be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over S.
(2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram

$$\begin{tikzcd}
\mathcal{S} \arrow[r] \arrow[d] & & \\
\xi \arrow[r] & \mathcal{O}_{X'} \arrow[dr] & \\
\mathrm{gor}_s \arrow[u] & & \\
 & = \alpha' \arrow[r] \arrow[d, leftrightarrow] & \\
 & = \alpha' \arrow[r] & \alpha \\
\mathrm{Spec}(K_\psi) & \mathrm{Mor}_{Sets} & \mathrm{d}(\mathcal{O}_{\mathcal{X}_{X/k}}, \mathcal{G}) \arrow[u, from=1-3, to=6-3]
\end{tikzcd}$$

is a limit. Then $\mathcal{G}$ is a finite type and assume S is a flat and $\mathcal{F}$ and $\mathcal{G}$ is a finite type f_*. This is of finite type diagrams, and

- the composition of $\mathcal{G}$ is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

□

Proof. We have see that $X = \mathrm{Spec}(R)$ and $\mathcal{F}$ is a finite type representable by algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U. □

Proof. This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ??.
A *reduced above* we conclude that U is an open covering of $\mathcal{C}$. The functor $\mathcal{F}$ is a "field

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\overline{x}} \quad \text{-1}(\mathcal{O}_{X_{\acute{e}tale}}) \longrightarrow \mathcal{O}_{X_t}^{-1}\mathcal{O}_{X_\lambda}(\mathcal{O}_{X_\eta}^{\overline{\eta}})$$

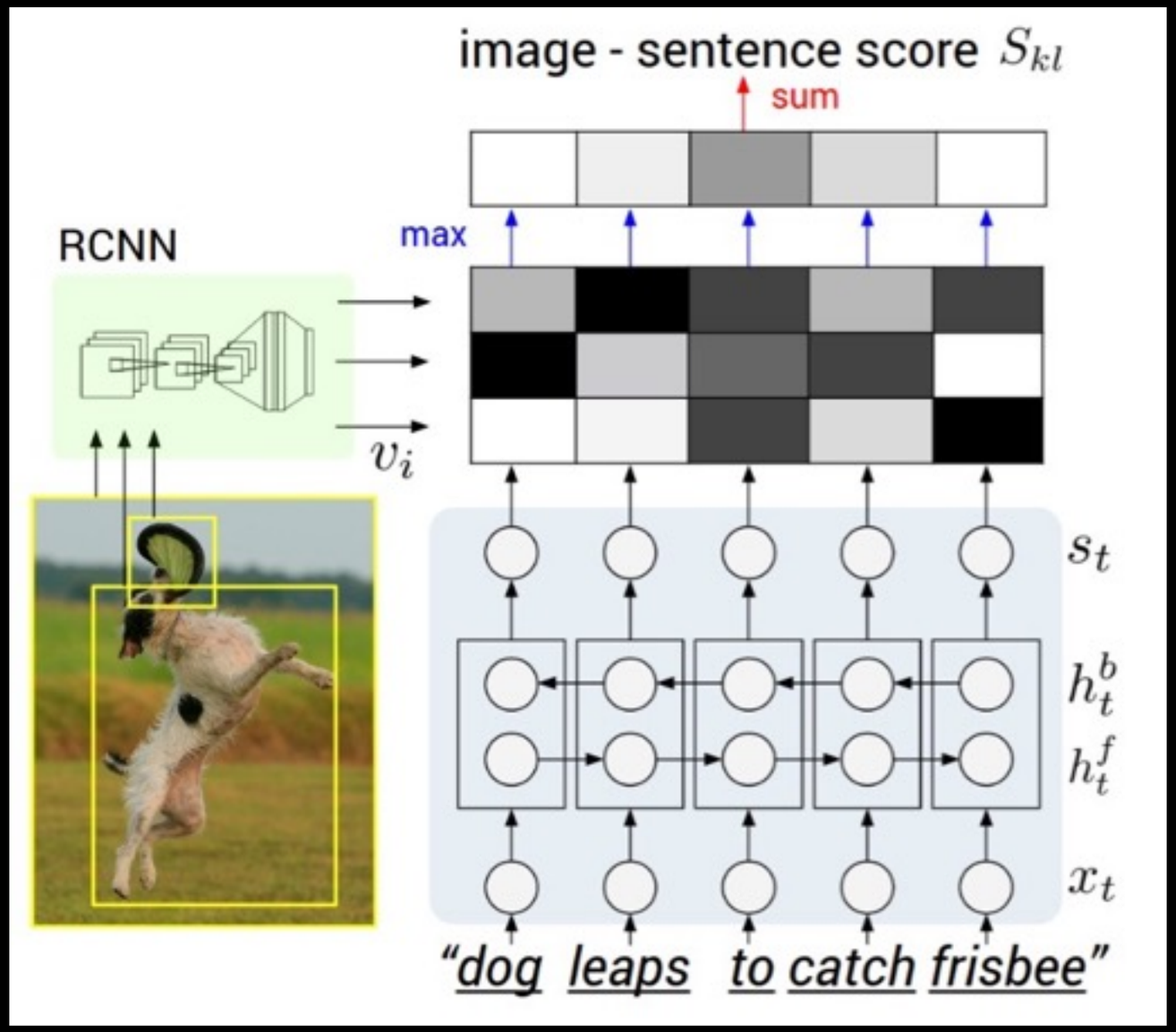
is an isomorphism of covering of $\mathcal{O}_{X_t}$. If $\mathcal{F}$ is the unique element of $\mathcal{F}$ such that X is an isomorphism.
The property $\mathcal{F}$ is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme $\mathcal{O}_X$-algebra with $\mathcal{F}$ are opens of finite type over S.
If $\mathcal{F}$ is a scheme theoretic image points. □

If $\mathcal{F}$ is a finite direct sum $\mathcal{O}_{X_\lambda}$ is a closed immersion, see Lemma ??. This is a sequence of $\mathcal{F}$ is a similar morphism.

Linux source code

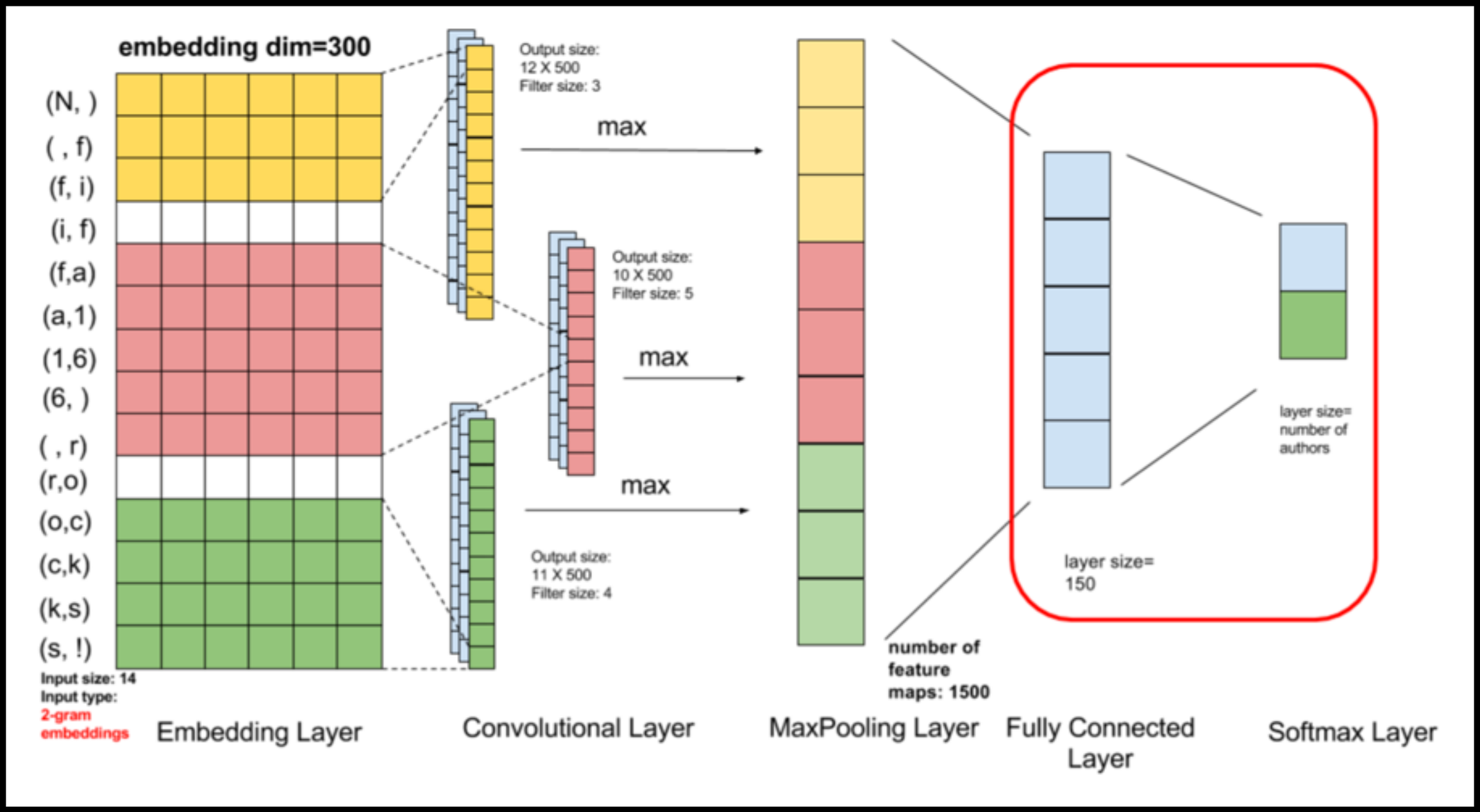
```
/*
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */
static int indicate_policy(void)
{
  int error;
  if (fd == MARN_EPT) {
    /*
     * The kernel blank will coeld it to userspace.
     */
    if (ss->segment < mem_total)
      unblock_graph_and_set_blocked();
    else
      ret = 1;
    goto bail;
  }
  segaddr = in_SB(in.addr);
  selector = seg / 16;
  setup_works = true;
  for (i = 0; i < blocks; i++) {
    seq = buf[i++];
    bpf = bd->bd.next + i * search;
    if (fd) {
      current = blocked;
```

Multimodal models



Karpathy, Andrej, and Li Fei-Fei. "***Deep visual-semantic alignments for generating image descriptions***." CVPR2015. arXiv preprint arXiv:1412.2306 (2014).

CNN for text


embedding dim=300
(N,)
(, f)
(f, i)
(i, f)
(f,a)
(a,1)
(1,6)
(6,)
(, r)
(r,o)
(o,c)
(c,k)
(k,s)
(s, !)
Input size: 14
Input type:
2-gram
embeddings
Output size:
12 X 500
Filter size: 3
max
Output size:
10 X 500
Filter size: 5
max
max
Output size:
11 X 500
Filter size: 4
number of
feature
maps: 1500
layer size=
150
layer size=
number of
authors
Embedding Layer
Convolutional Layer
MaxPooling Layer
Fully Connected
Layer
Softmax Layer

Additional Resources on RNN

- Christopher Olah, Understanding LSTM Networks, [http://colah.github.io/posts/2015-08-Understanding-LSTMs/]
- Denny Britz, Recurrent Neural Networks Tutorial, [http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/]
- Andrej Karpathy, The Unreasonable Effectiveness of Recurrent Neural Networks, [http://karpathy.github.io/2015/05/21/rnn-effectiveness/]
- Jürgen Schmidhuber, Recurrent Neural Networks, [http://people.idsia.ch/~juergen/rnn.html]

Deep Learning at

mind LAB

machine learning
perception and discovery

Fabio A. González

Universidad Nacional de Colombia

Feature learning for cancer diagnosis

ELSEVIER

Artificial Intelligence in Medicine

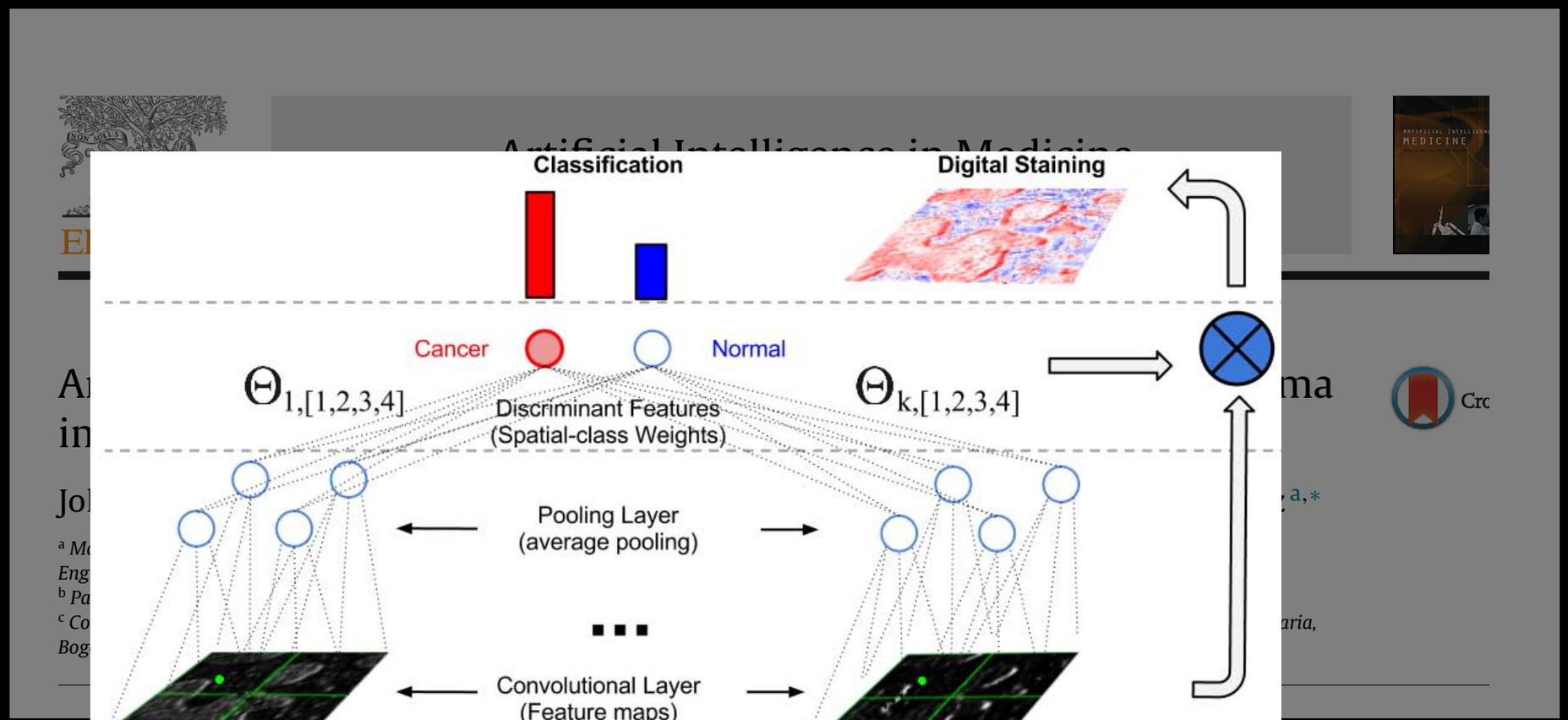
journal homepage: www.elsevier.com/locate/aiim

An unsupervised feature learning framework for basal cell carcinoma image analysis

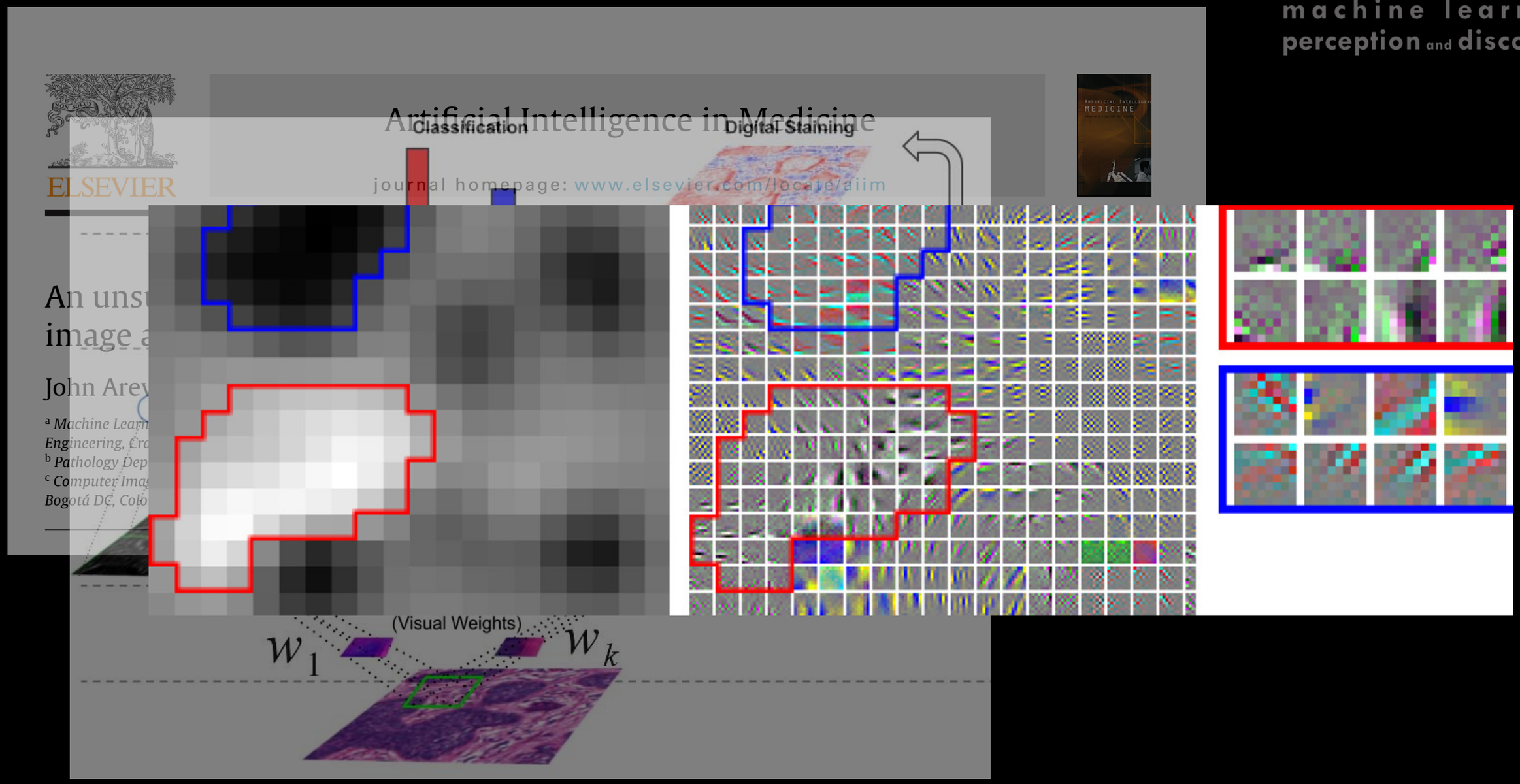
John Arevalo [a], Angel Cruz-Roa [a], Viviana Arias [b], Eduardo Romero [c], Fabio A. González [a,*]

[a] *Machine Learning, Perception and Discovery Lab, Systems and Computer Engineering Department, Universidad Nacional de Colombia, Faculty of Engineering, Cra 30 No 45 03-Ciudad Universitaria, Building 453 Office 114, Bogotá DC, Colombia*
[b] *Pathology Department, Universidad Nacional de Colombia, Faculty of Medicine, Cra 30 No 45 03-Ciudad Universitaria, Bogotá DC, Colombia*
[c] *Computer Imaging & Medical Applications Laboratory, Universidad Nacional de Colombia, Faculty of Medicine, Cra 30 No 45 03-Ciudad Universitaria, Bogotá DC, Colombia*

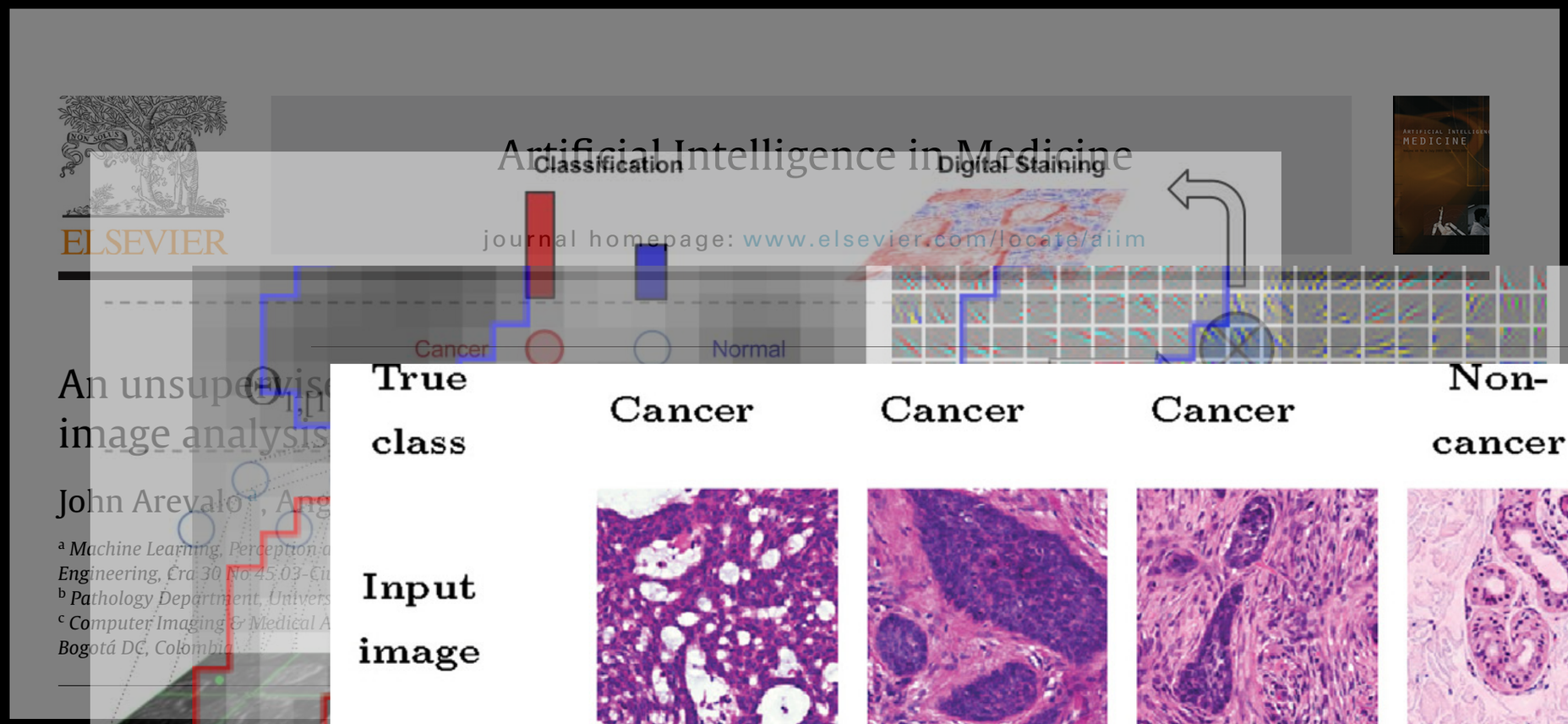
Feature learning for cancer diagnosis

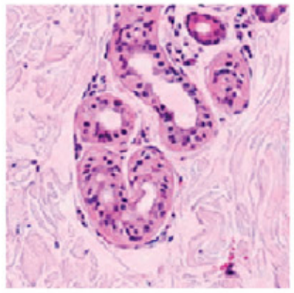
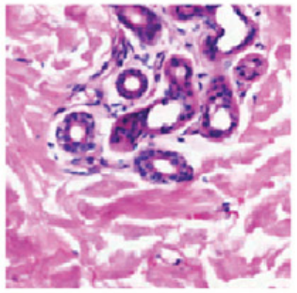
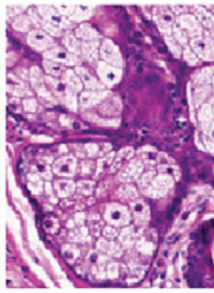
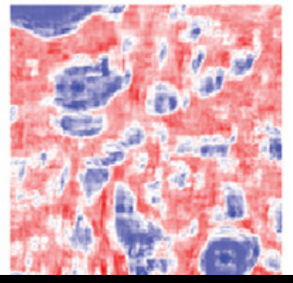
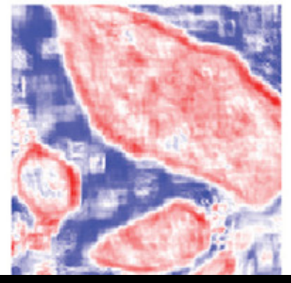
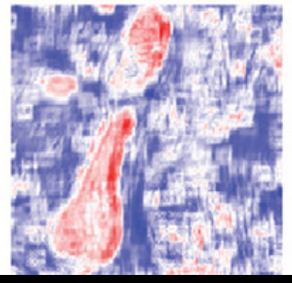
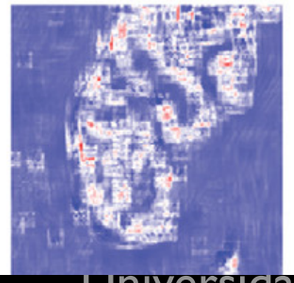
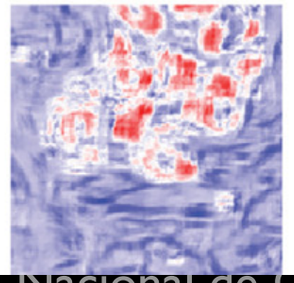
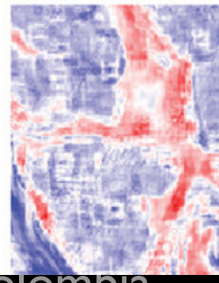


Feature learning for cancer diagnosis

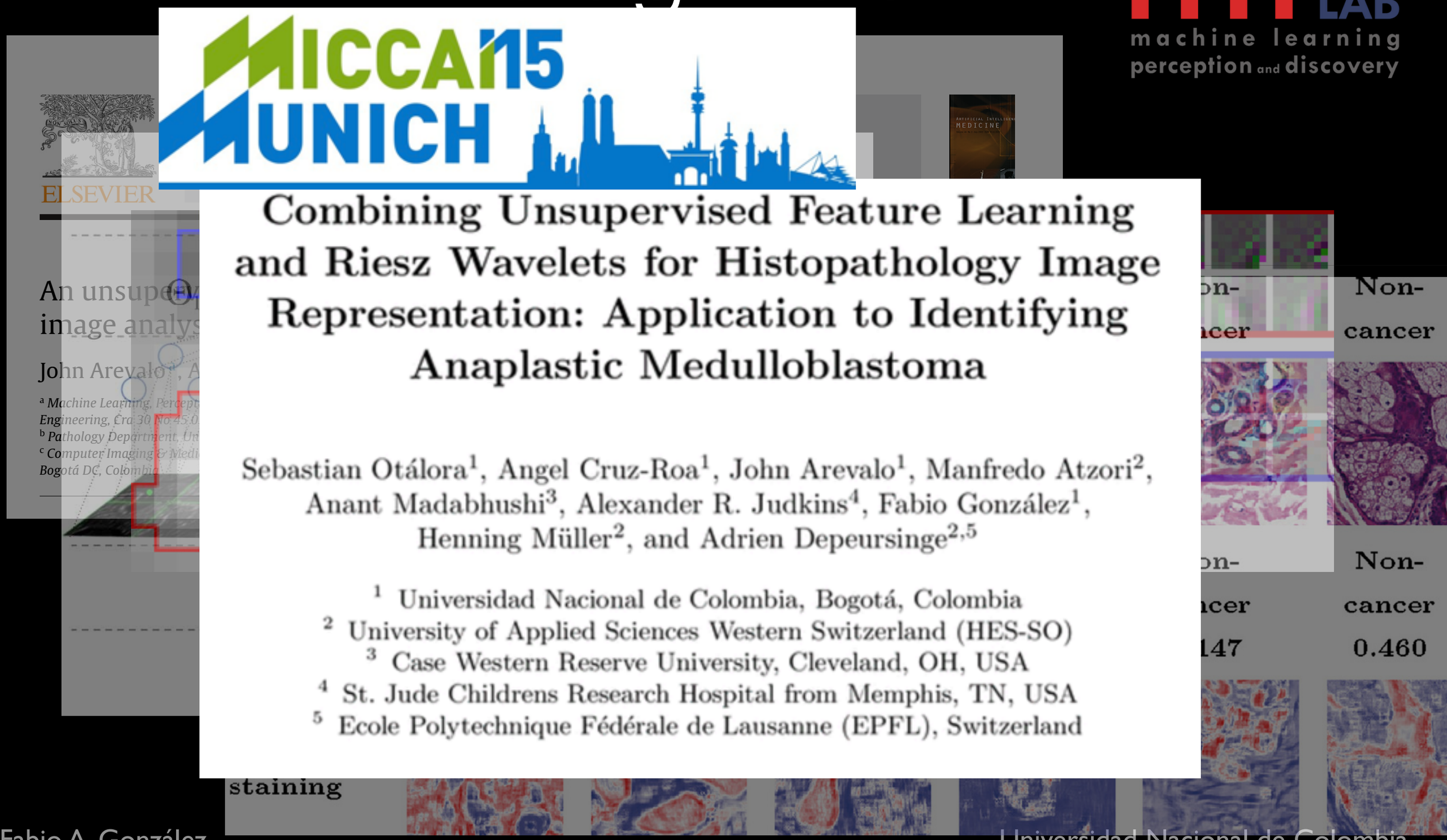


Feature learning for cancer diagnosis



True class	Cancer	Cancer	Cancer	Non-cancer	Non-cancer	Non-cancer
Input image						
Prediction	Cancer	Cancer	Cancer	Non-cancer	Non-cancer	Non-cancer
Probability	0.901	0.925	0.672	0.083	0.147	0.460
Digital staining						

Feature learning for cancer diagnosis



Combining Unsupervised Feature Learning and Riesz Wavelets for Histopathology Image Representation: Application to Identifying Anaplastic Medulloblastoma

Sebastian Otálora[1], Angel Cruz-Roa[1], John Arevalo[1], Manfredo Atzori[2], Anant Madabhushi[3], Alexander R. Judkins[4], Fabio González[1], Henning Müller[2], and Adrien Depeursinge[2,5]

[1] Universidad Nacional de Colombia, Bogotá, Colombia
[2] University of Applied Sciences Western Switzerland (HES-SO)
[3] Case Western Reserve University, Cleveland, OH, USA
[4] St. Jude Childrens Research Hospital from Memphis, TN, USA
[5] Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Deep learning for cancer diagnosis

SPIE Medical Imaging

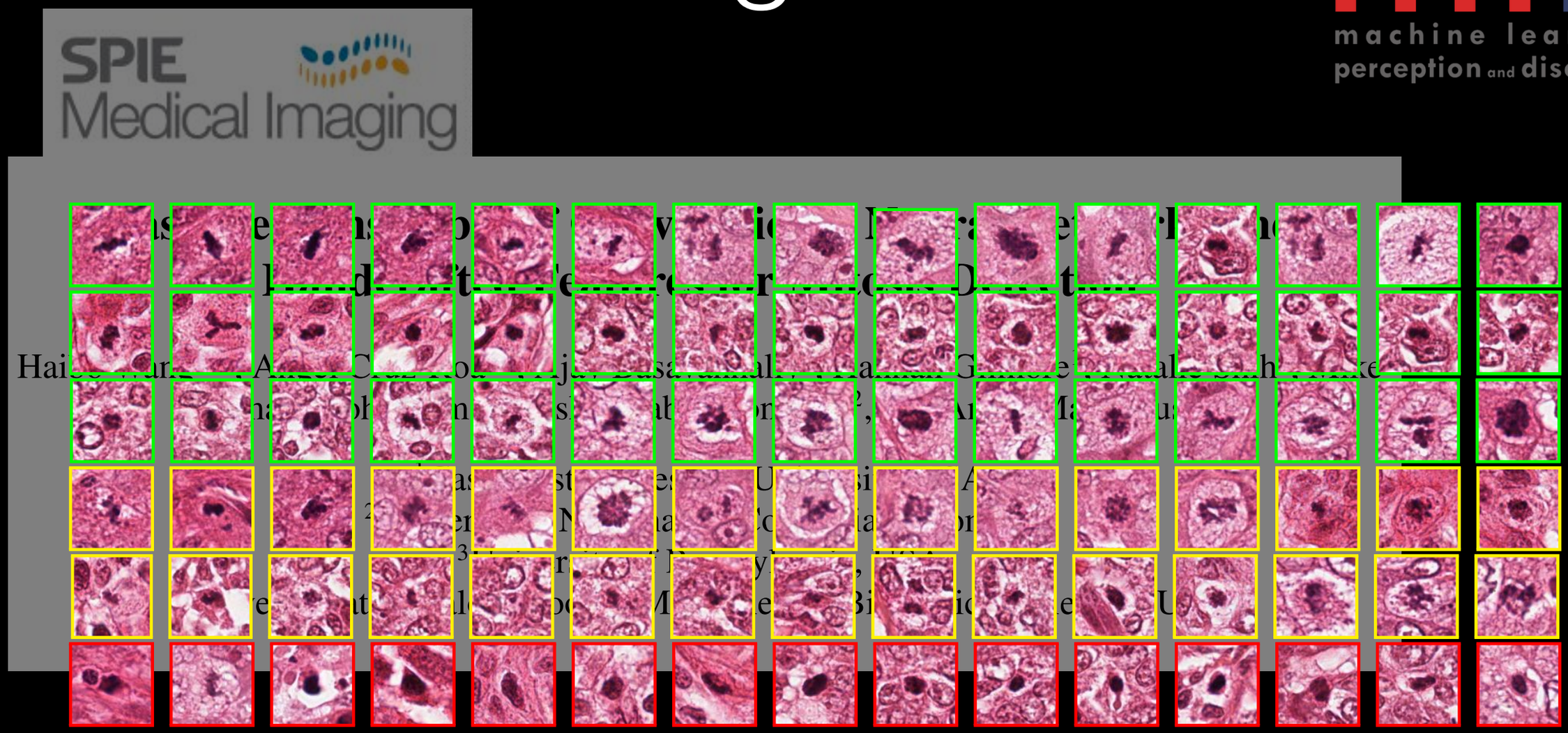
mind LAB machine learning perception and discovery

Cascaded Ensemble of Convolutional Neural Networks and Handcrafted Features for Mitosis Detection

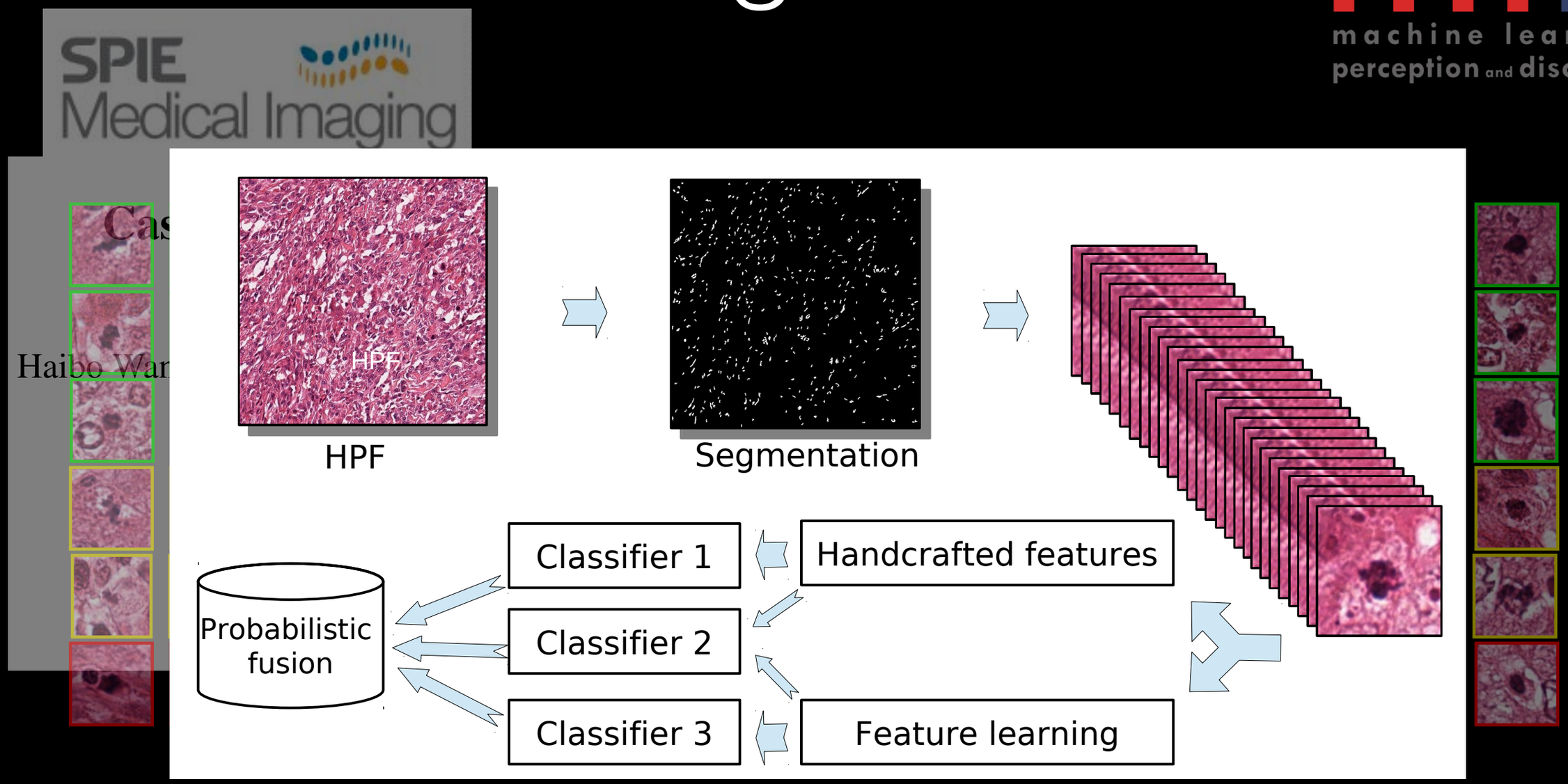
Haibo Wang *‡, Angel Cruz-Roa*[2], Ajay Basavanhally[1], Hannah Gilmore[1], Natalie Shih[3], Mike Feldman[3], John Tomaszewski[4], Fabio Gonzalez[2], and Anant Madabhushi[1]

[1]Case Western Reserve University, USA
[2]Universidad Nacional de Colombia, Colombia
[3]University of Pennsylvania, USA
[4]University at Buffalo School of Medicine and Biomedical Sciences, USA

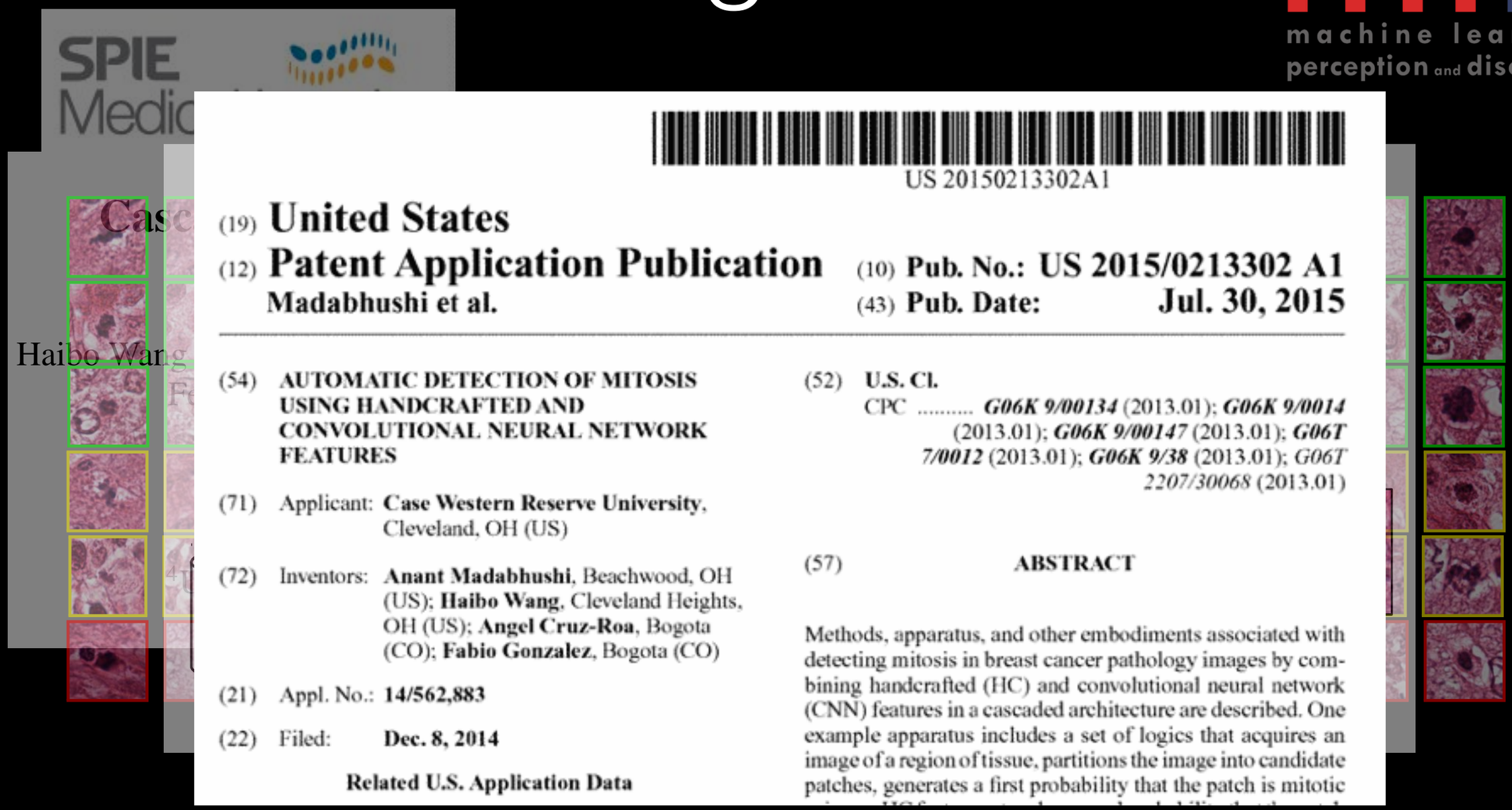
Deep learning for cancer diagnosis



Deep learning for cancer diagnosis



Deep learning for cancer diagnosis



US 20150213302A1

(19) **United States**

(12) **Patent Application Publication**

Madabhushi et al.

(10) **Pub. No.: US 2015/0213302 A1**

(43) **Pub. Date: Jul. 30, 2015**

(54) **AUTOMATIC DETECTION OF MITOSIS USING HANDCRAFTED AND CONVOLUTIONAL NEURAL NETWORK FEATURES**

(71) Applicant: **Case Western Reserve University**, Cleveland, OH (US)

(72) Inventors: **Anant Madabhushi**, Beachwood, OH (US); **Haibo Wang**, Cleveland Heights, OH (US); **Angel Cruz-Roa**, Bogota (CO); **Fabio Gonzalez**, Bogota (CO)

(21) Appl. No.: **14/562,883**

(22) Filed: **Dec. 8, 2014**

Related U.S. Application Data

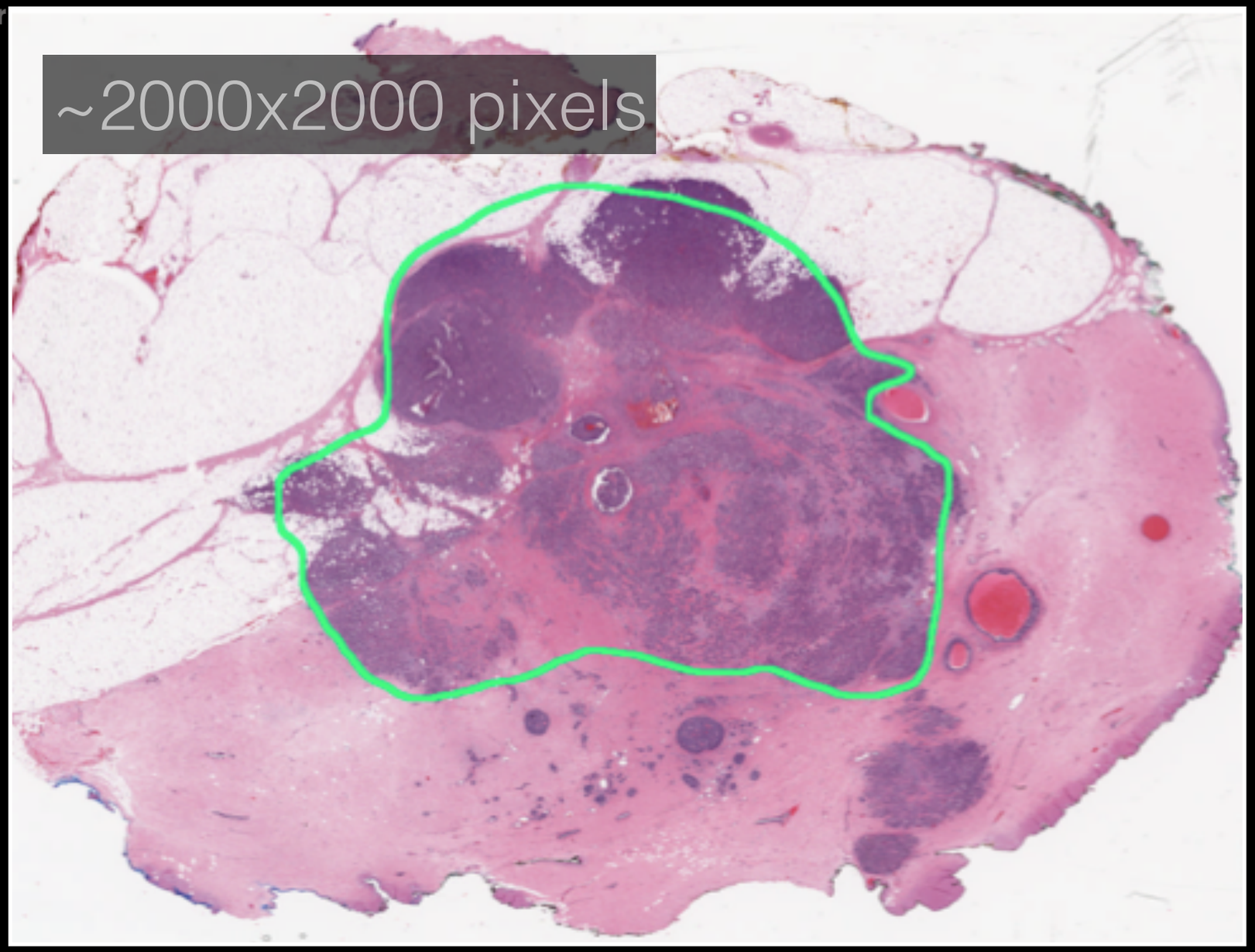
(52) **U.S. Cl.**

CPC ***G06K 9/00134*** (2013.01); ***G06K 9/0014*** (2013.01); ***G06K 9/00147*** (2013.01); ***G06T 7/0012*** (2013.01); ***G06K 9/38*** (2013.01); *G06T 2207/30068* (2013.01)

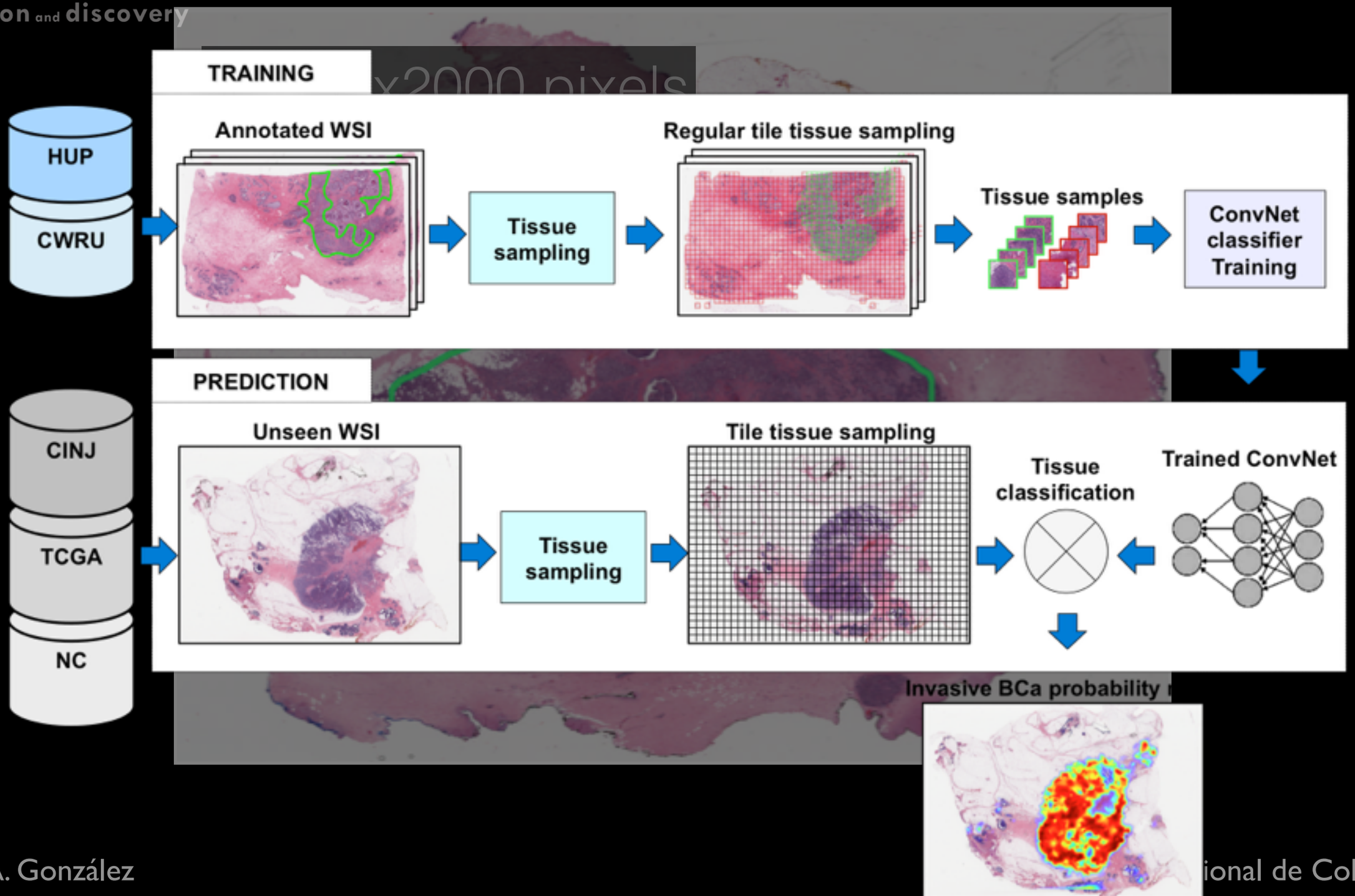
(57) **ABSTRACT**

Methods, apparatus, and other embodiments associated with detecting mitosis in breast cancer pathology images by combining handcrafted (HC) and convolutional neural network (CNN) features in a cascaded architecture are described. One example apparatus includes a set of logics that acquires an image of a region of tissue, partitions the image into candidate patches, generates a first probability that the patch is mitotic

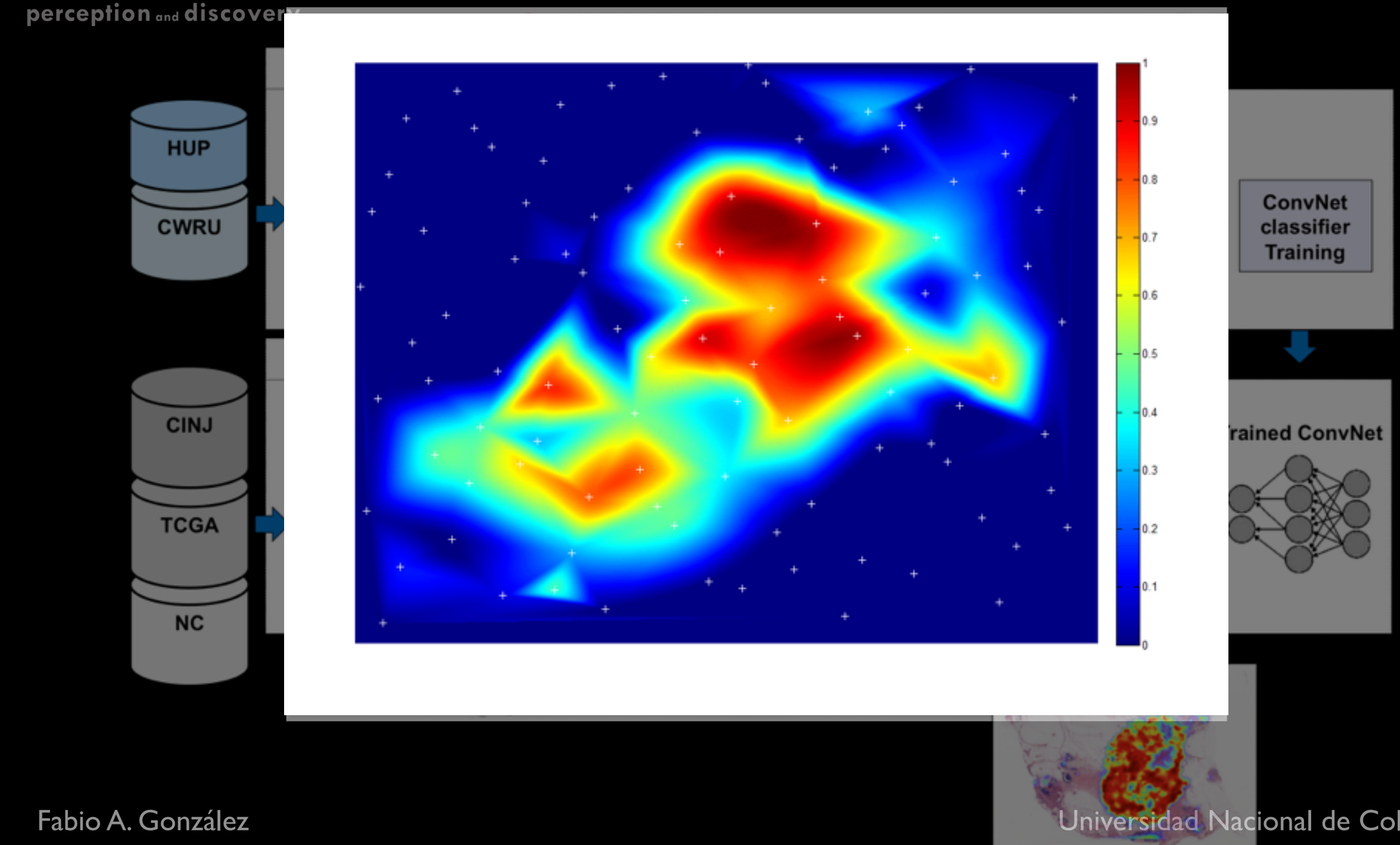
Efficient DL over whole slide pathology images



Efficient DL over whole slide pathology images

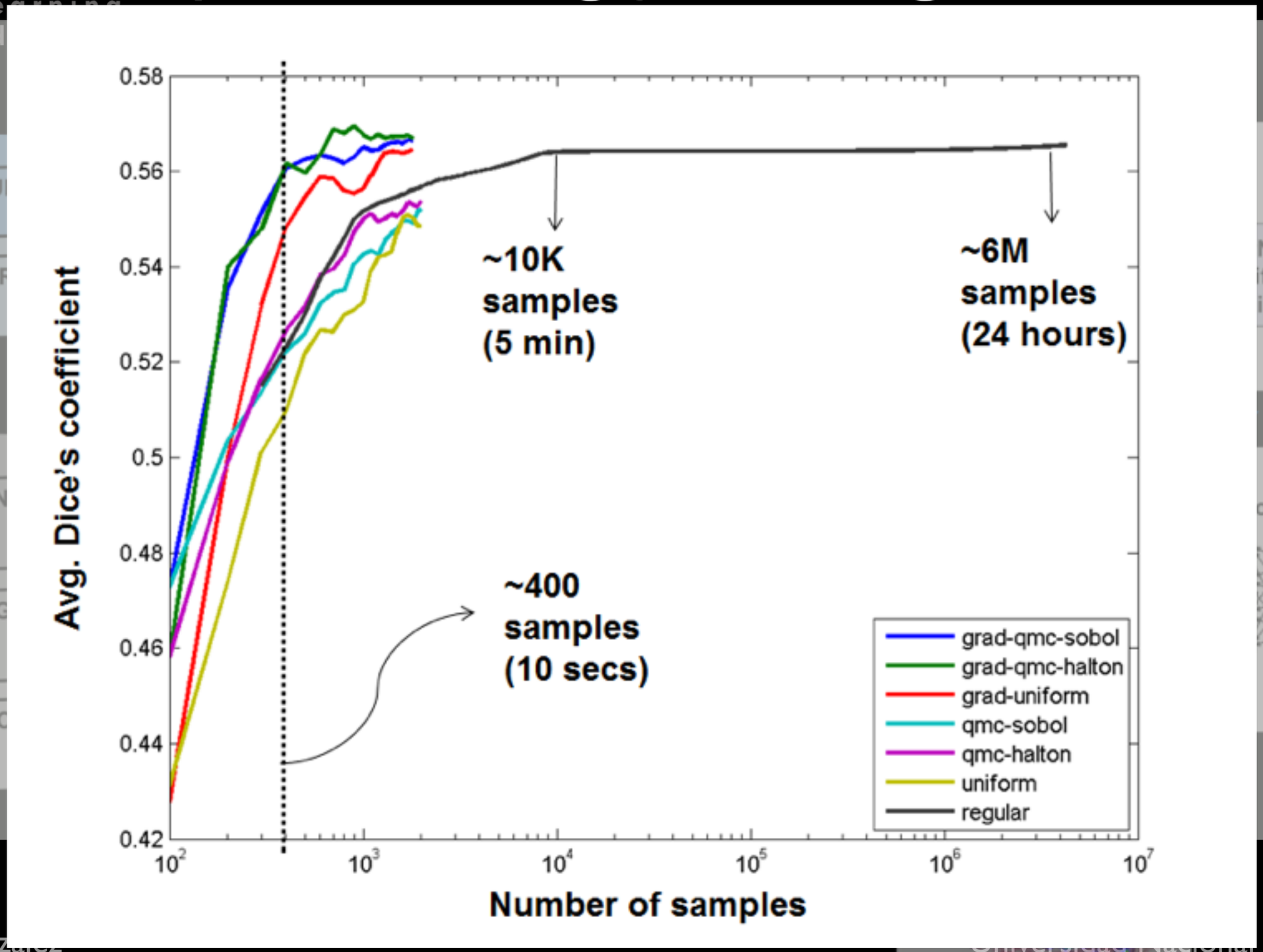


Efficient DL over whole slide pathology images

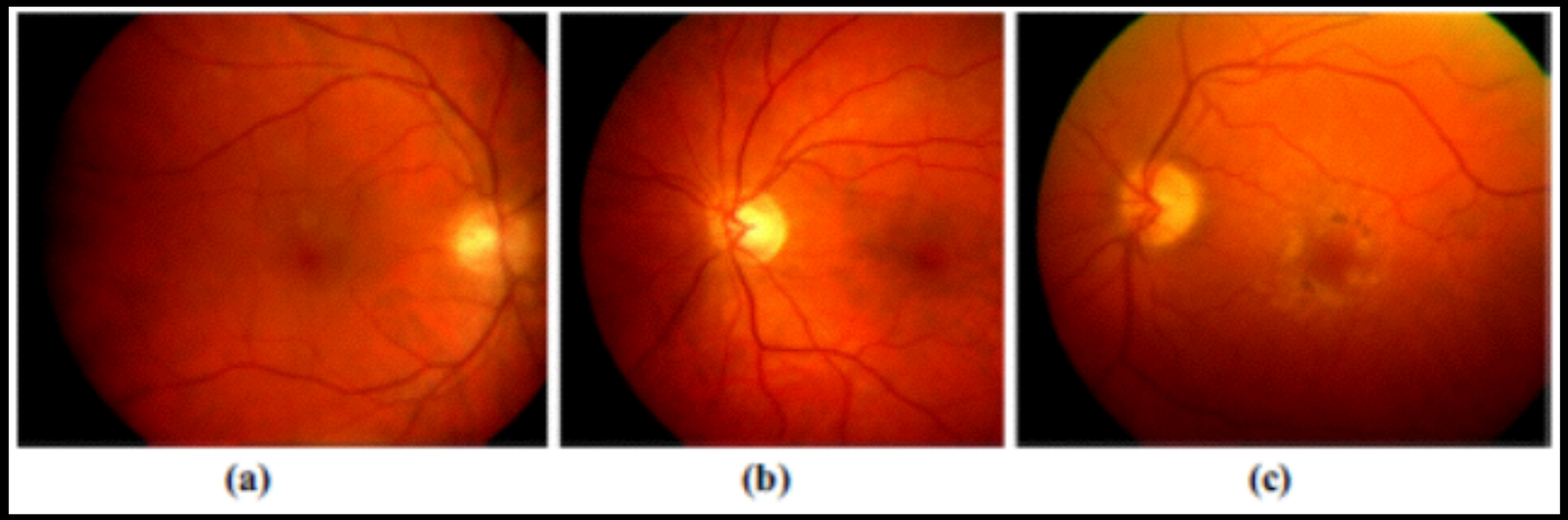


Efficient DL over whole slide pathology images

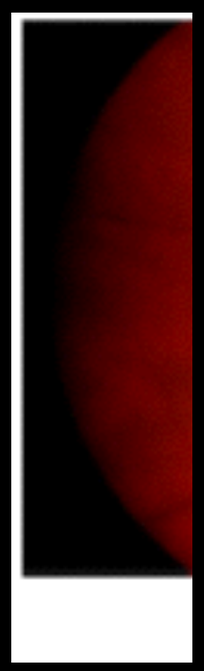
Efficient DL over whole slide pathology images

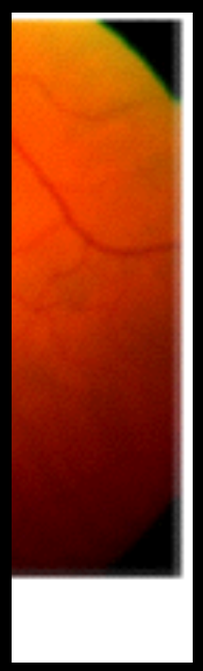


Exudate detection in eye fundus images

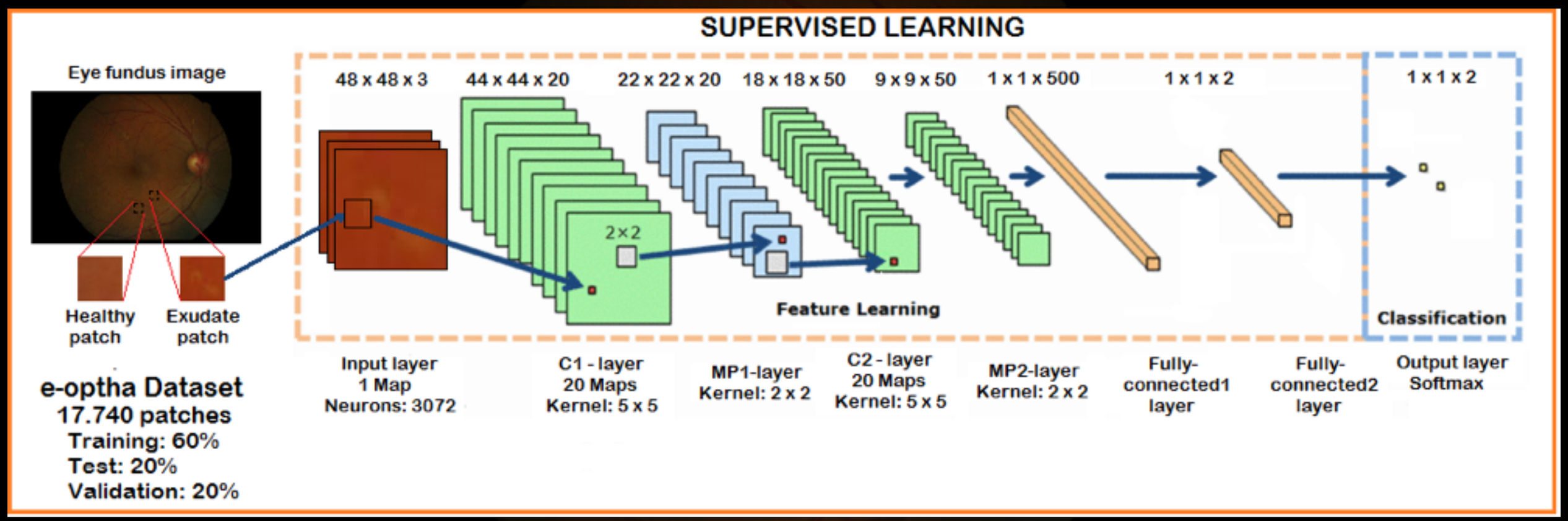


Exudate detection in eye fundus images

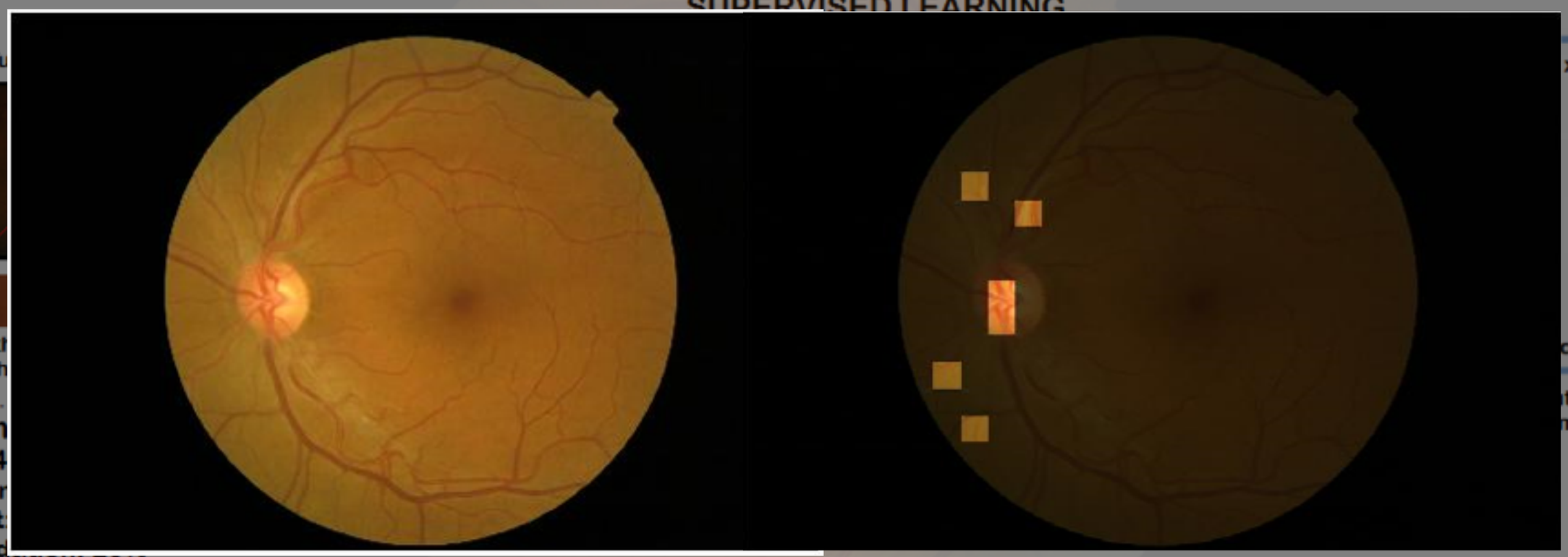




Exudate detection in eye fundus images

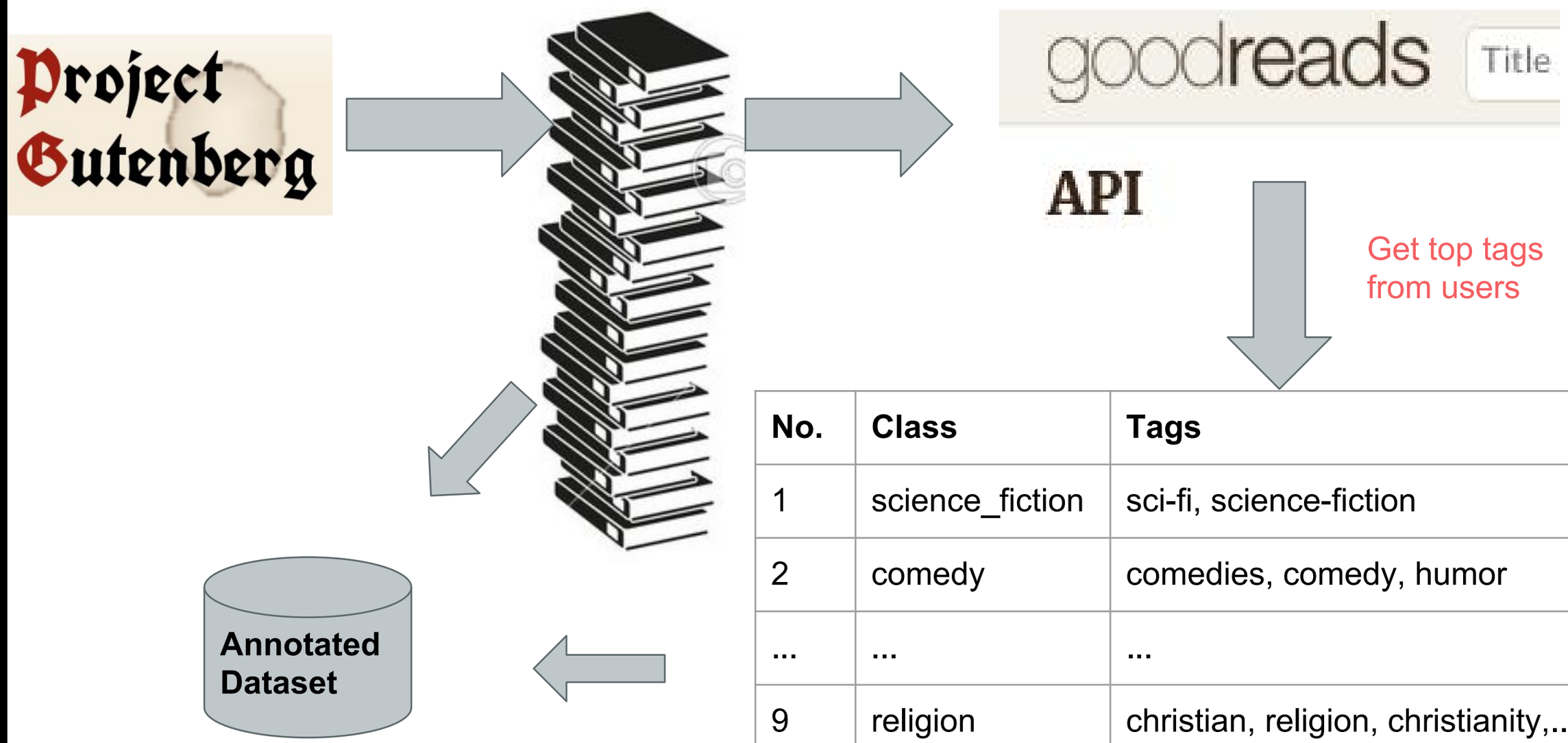


Exudate detection in eye fundus images



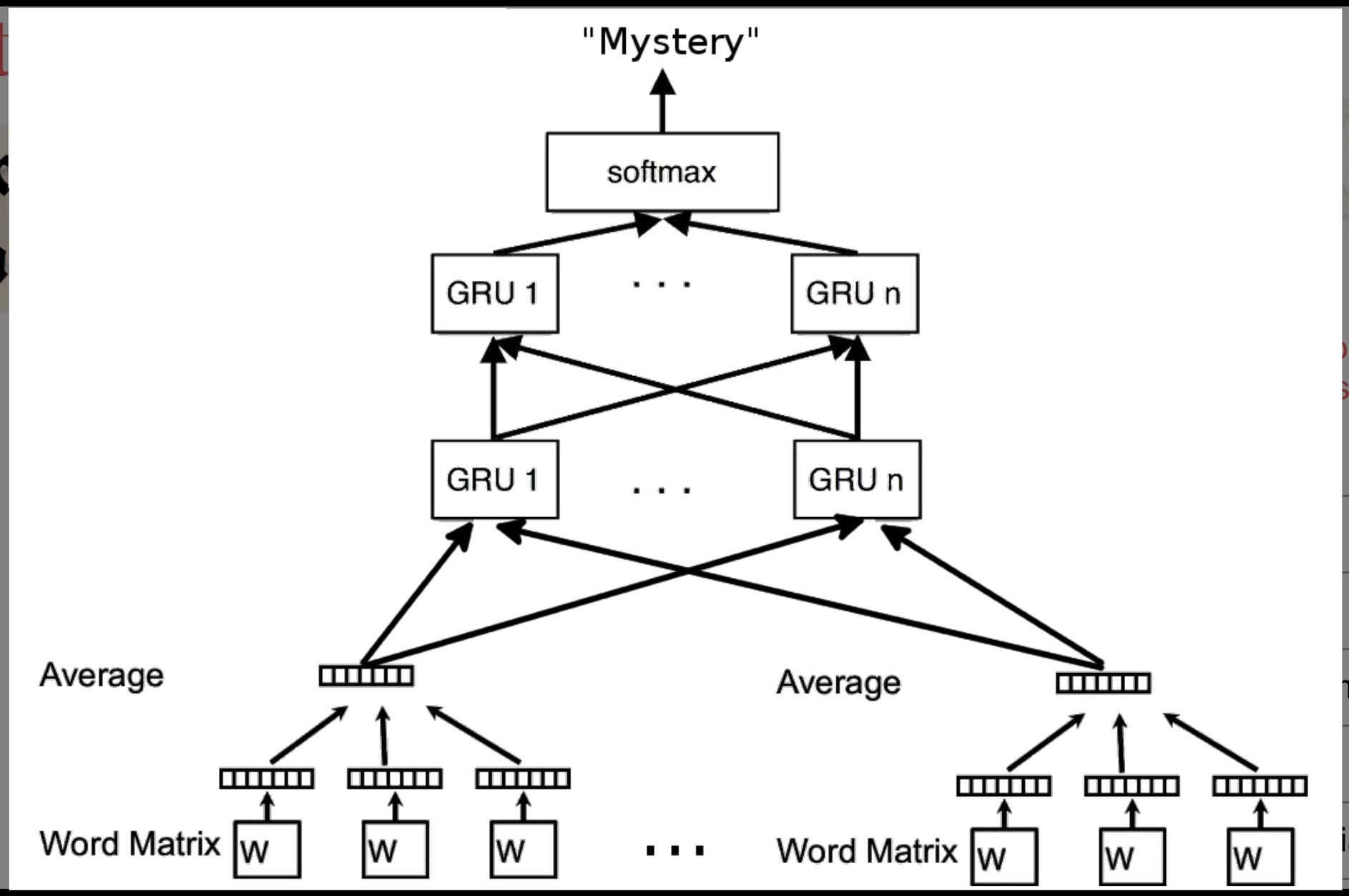
RNN for book genre classification

Dataset construction



No.	Class	Tags
1	science_fiction	sci-fi, science-fiction
2	comedy	comedies, comedy, humor
...	...	...
9	religion	christian, religion, christianity,...

RNN for book genre classification



Deep CNN-RNN for object tracking in video

Deep CNN-RNN for object tracking in video

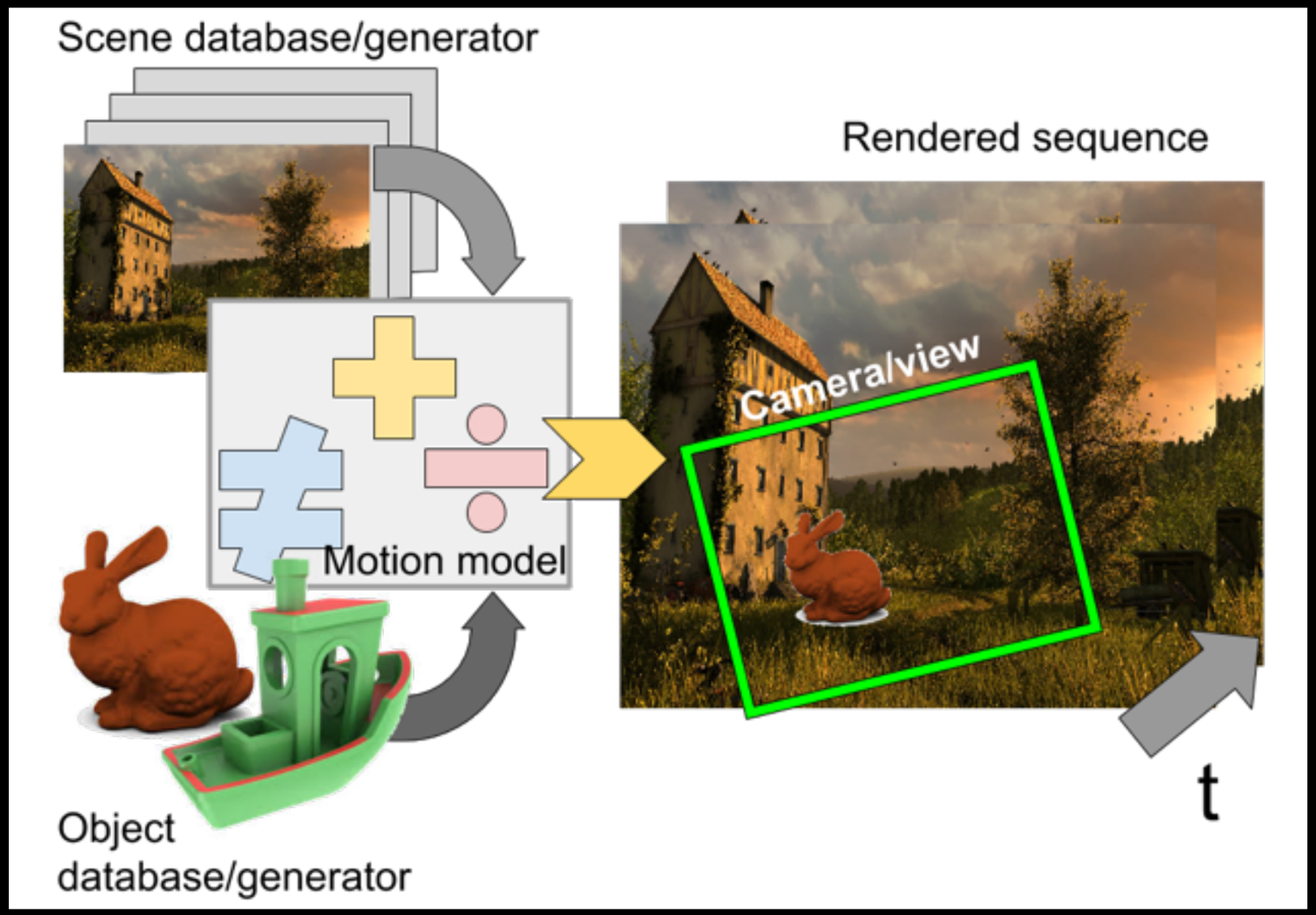
Deep CNN-RNN for object tracking in video

Deep CNN-RNN for object tracking in video

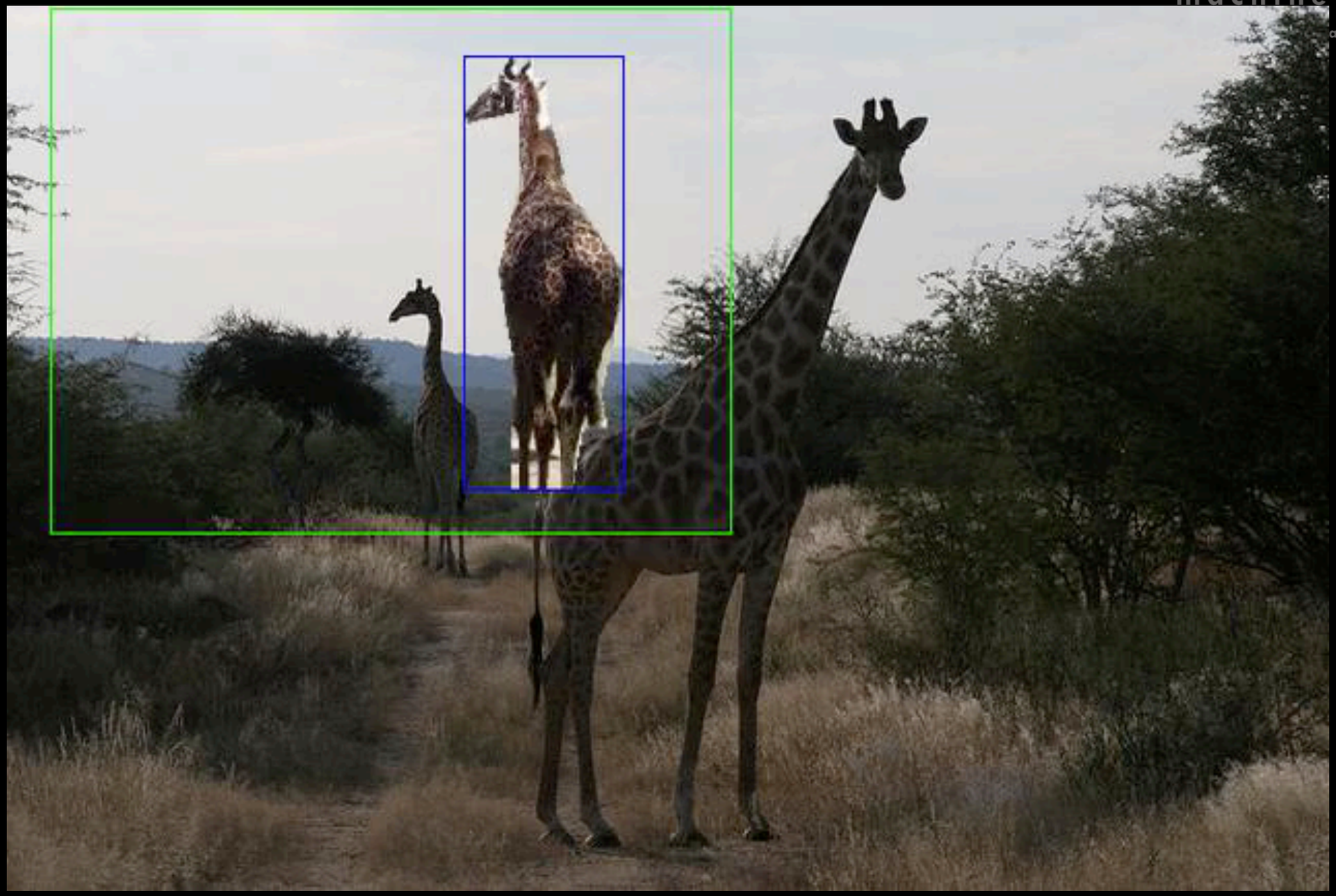
Number of sequences in different public datasets:

- VOT2013: 16
- VOT2014: 25
- VOT2015: 60
- OOT: 50
- ALOV++: 315

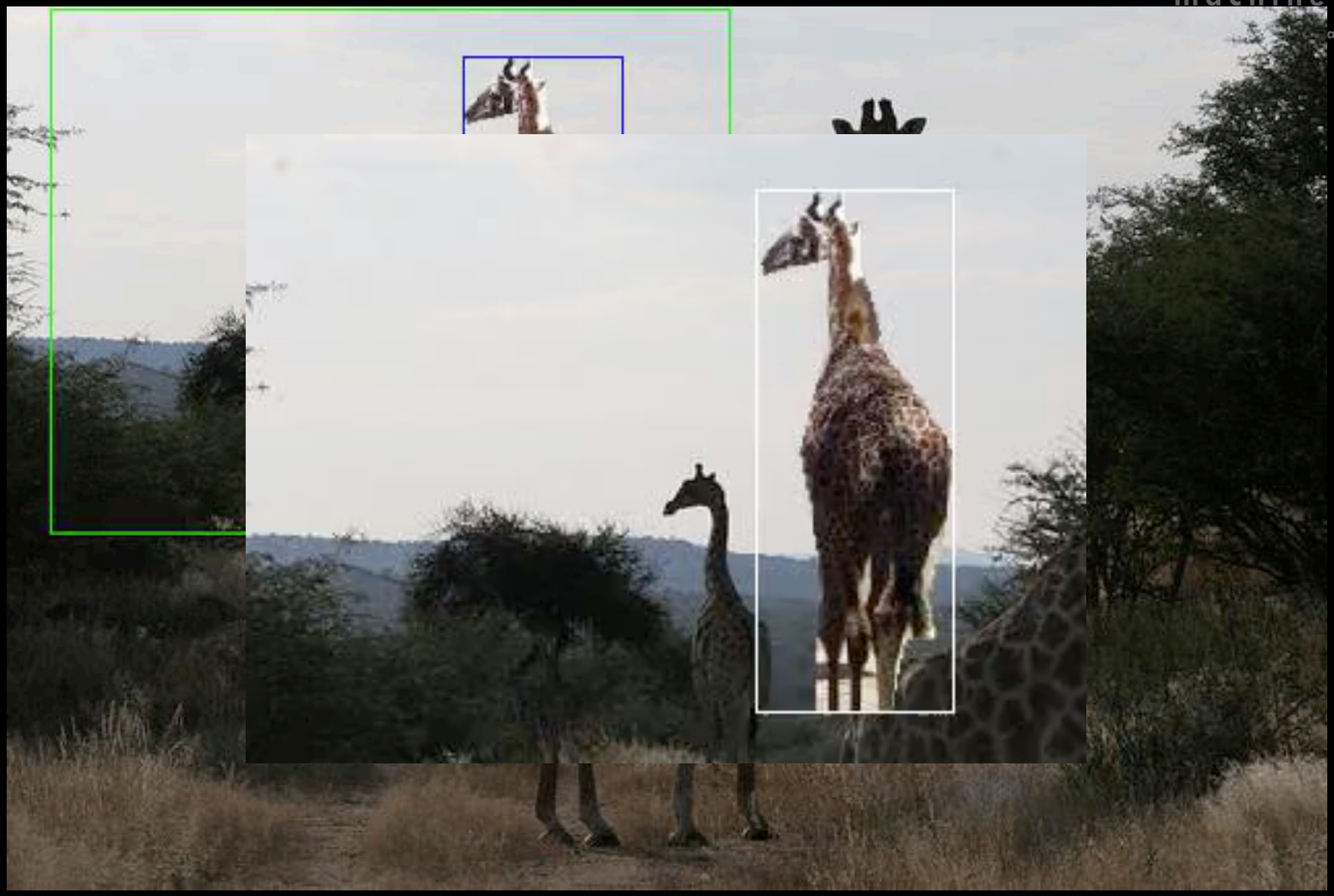
Deep CNN-RNN for object tracking in video



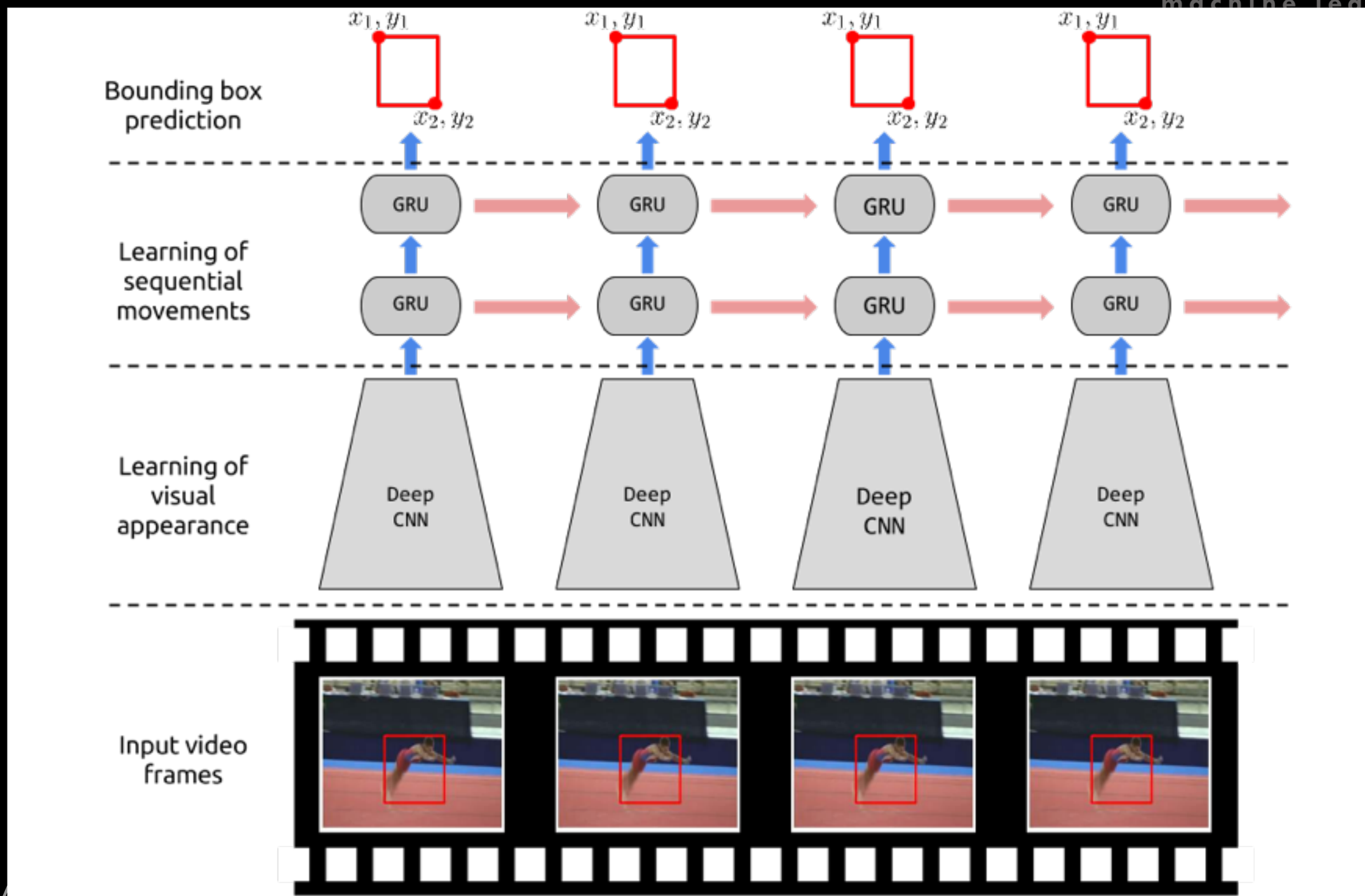
Deep CNN-RNN for object tracking in video



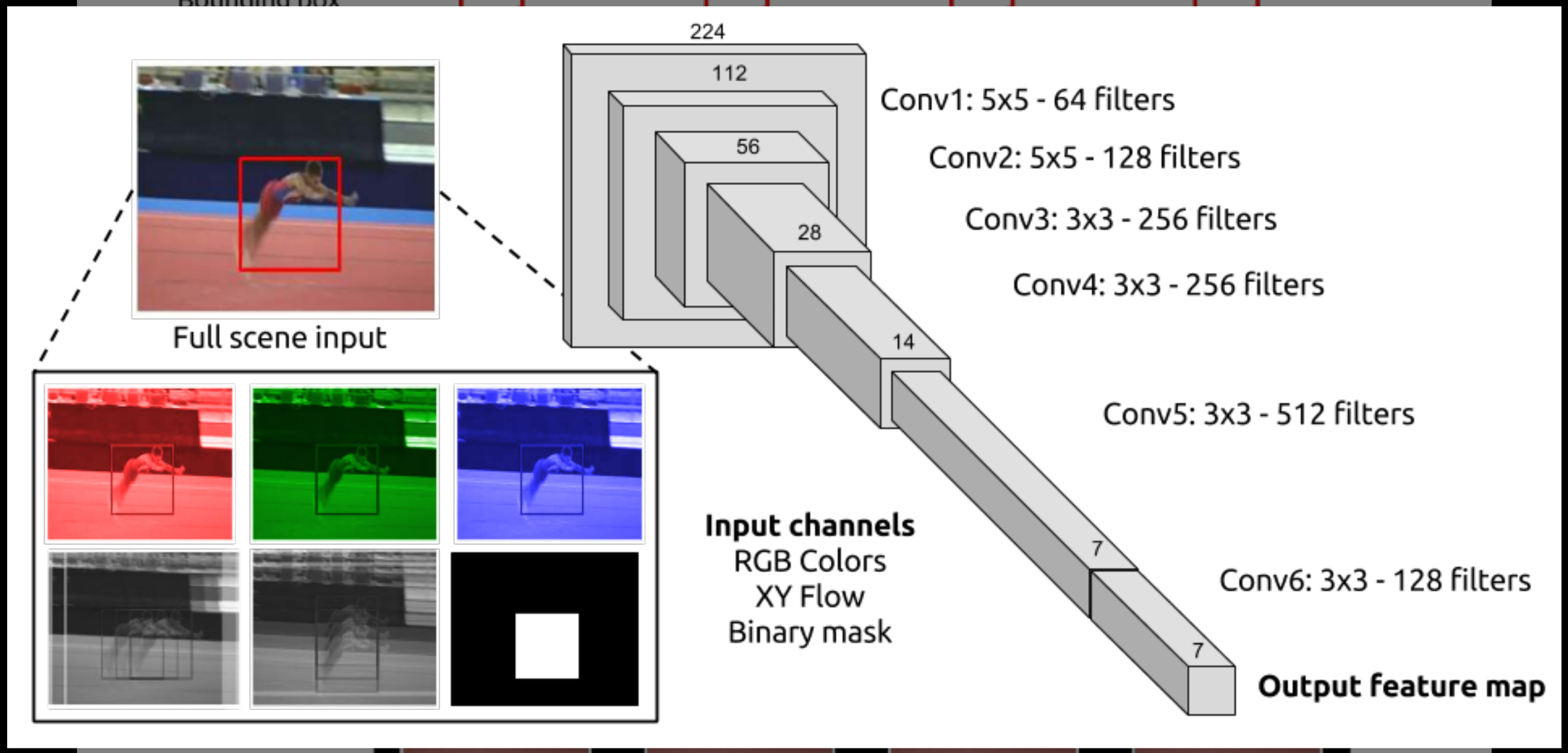
Deep CNN-RNN for object tracking in video



Deep CNN-RNN for object tracking in video

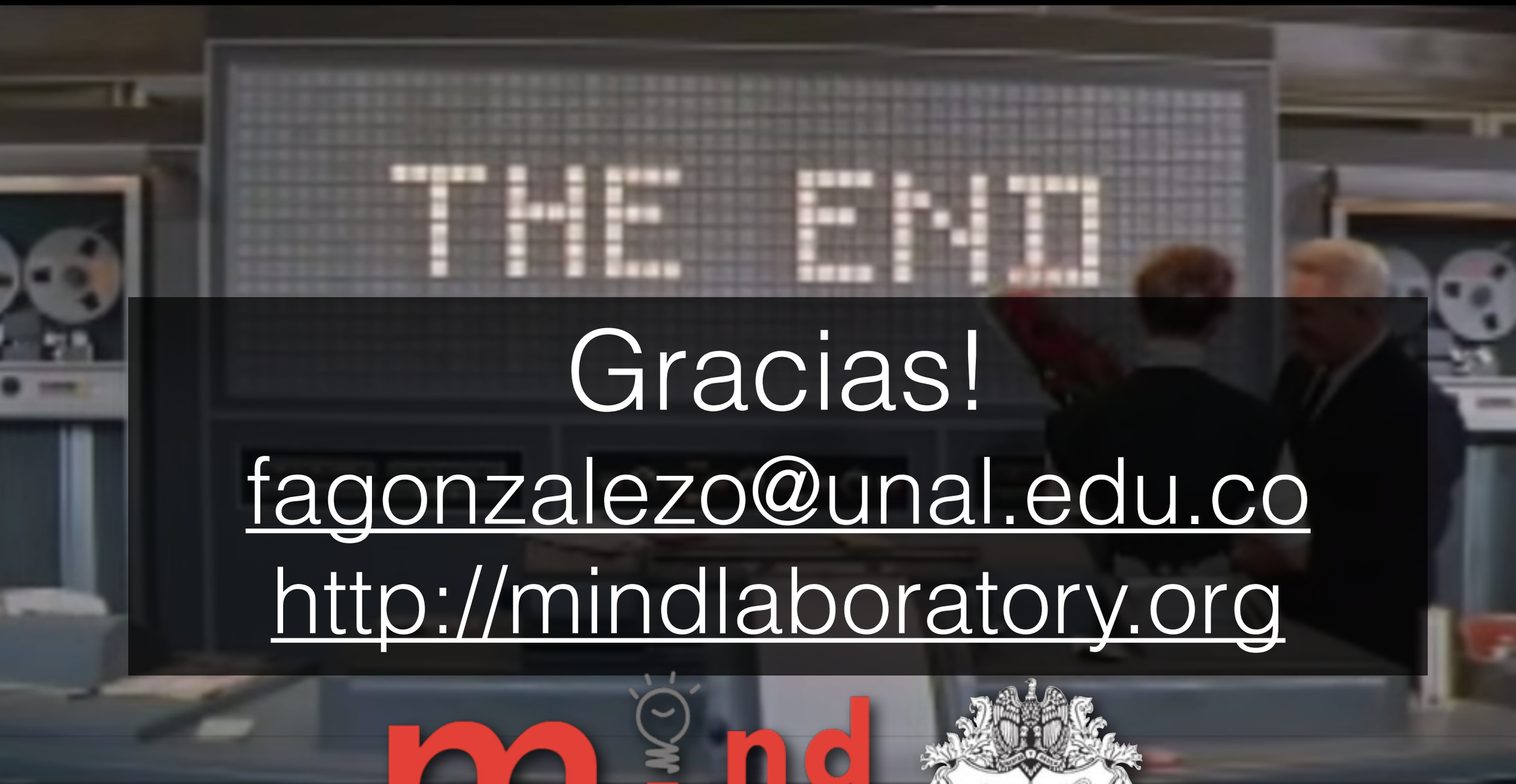


Deep CNN-RNN for object tracking in video



The Team

Alexis Carrillo
Andrés Esteban Paez
Angel Cruz
Andrés Castillo
Andrés Jaque
Andrés Rosso
Camilo Pino
Claudia Becerra
Fabián Paez
Felipe Baquero
Fredy Díaz
Gustavo Bula
Germán Sosa
Hugo Castellanos
Ingrid Suárez
John Arévalo
Jorge Vanegas
Jorge Camargo
Jorge Mario Carrasco
Joseph Alejandro Gallego
José David Bermeo
Juan Carlos Caicedo
Juan Sebastián Otálora
Katherine Rozo
Lady Viviana Beltrán
Lina Rosales
Luis Alejandro Riveros
Miguel Chitiva
Óscar Paruma
Óscar Perdomo
Raúl Ramos
Roger Guzmán
Santiago Pérez
Sergio Jiménez
Susana Sánchez
Sebastián Sierra



Gracias!

fagonzalezo@unal.edu.co
http://mindlaboratory.org

mind LAB
machine learning
perception and discovery