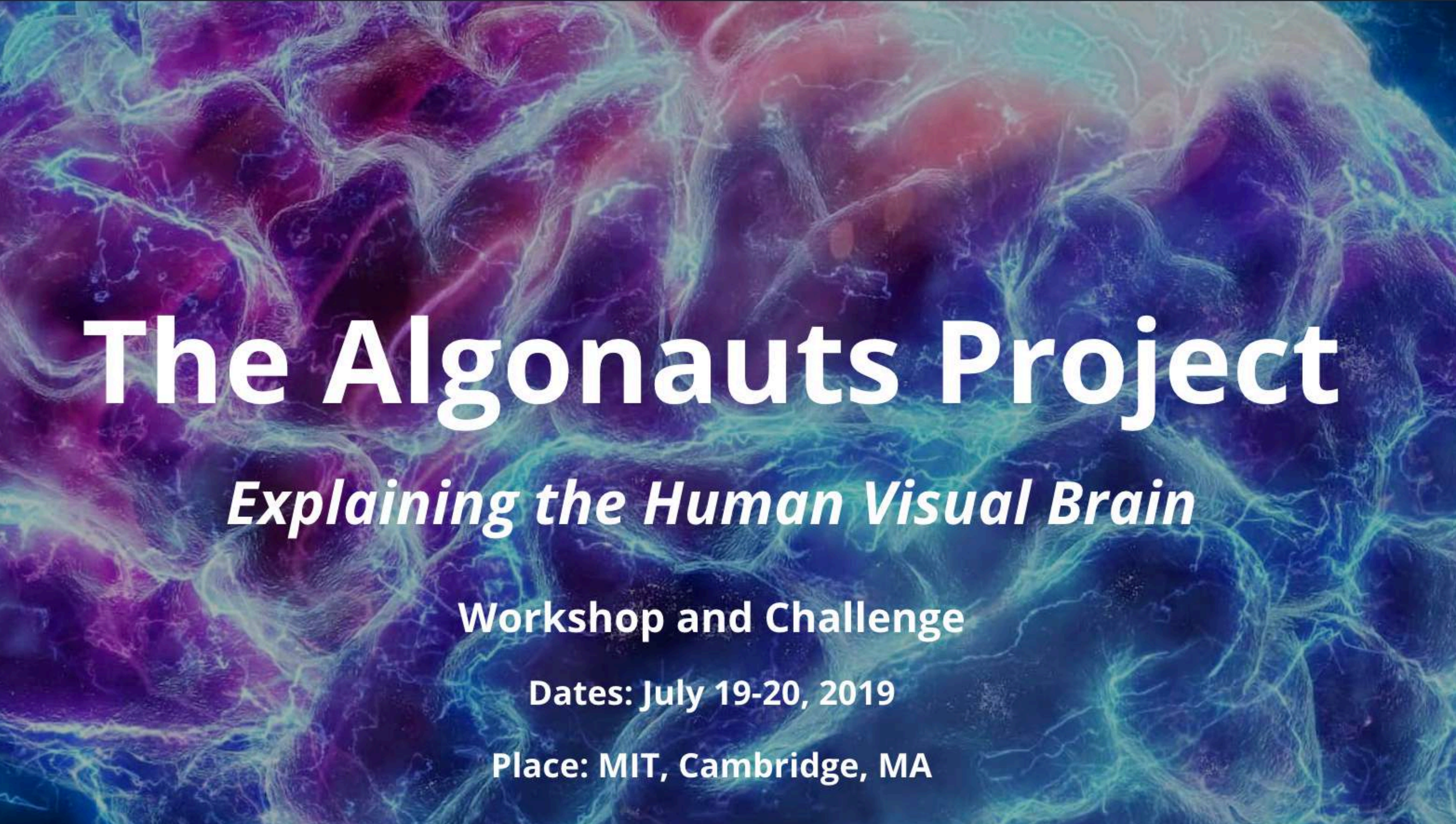


algonauts.csail.mit.edu

Team and Sponsors

Team Leader: Radoslaw Cichy

Research Group Leader, Freie Universität Berlin

Team Leader: Aude Oliva

Principal Research Scientist, MIT

Team Leader: Gemma Roig

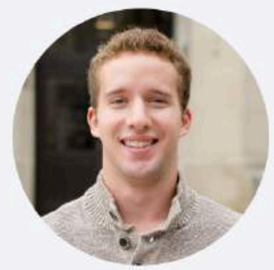
Assistant Professor, SUTD

Alex Andonian

Research Assistant, MIT

Kshitij Dwivedi

PhD Student, SUTD



Benjamin Lahner

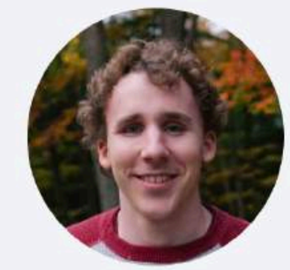
Research Assistant, MIT

Kandan Ramakrishnan

Postdoctoral Researcher, MIT

Yalda Mohsenzadeh

Postdoctoral Researcher, MIT



Alex Lascelles

Research Assistant, MIT

Fern Keniston

Program Coordinator and Assistant to the Directors, MIT

Samantha Smiley

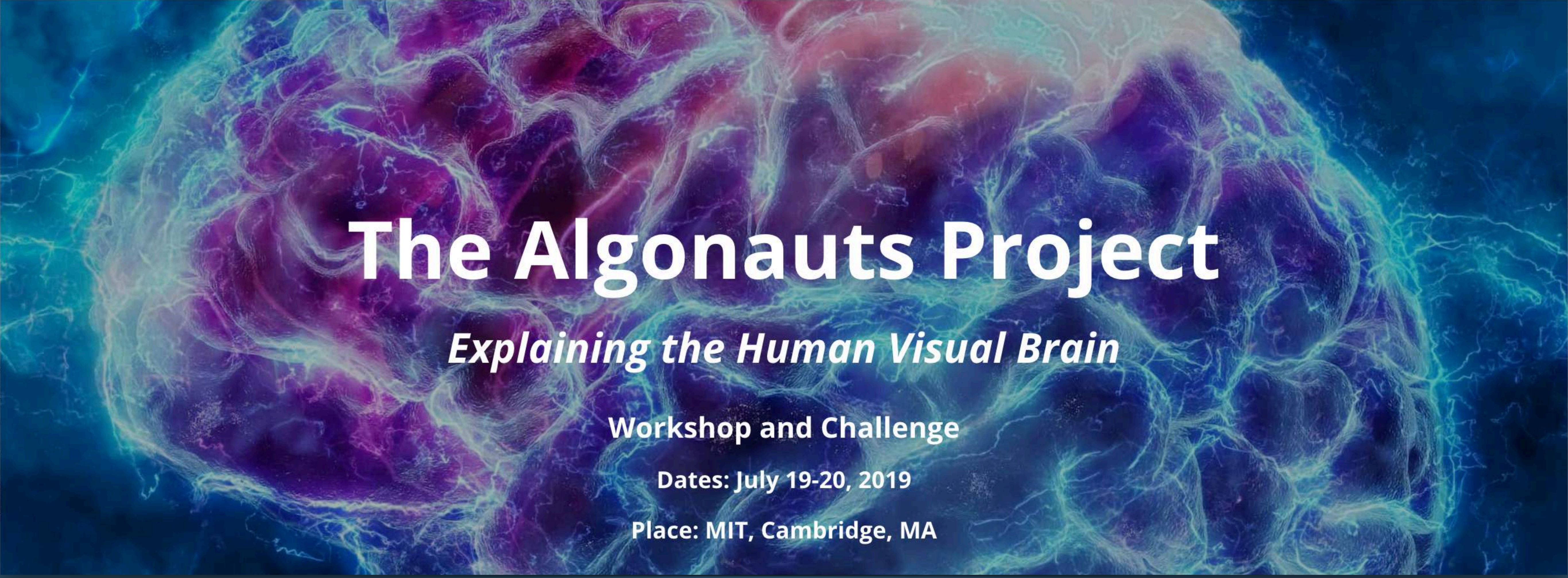
Administrative Assistant, MIT

Kim Martineau

Communications Officer, MIT

MIT-IBM Watson AI Lab

MIT Quest for Intelligence



Workshop for Students Day

19 July 2019

The Algonauts Project

Explaining the Human Visual Brain

Time	Event
12:30 pm - 1:00 pm	Registration / Welcome
1:00 pm - 2:00 pm	**Introduction to Neural Networks** ***Gemma Roig***
2:00 pm - 2:15 pm	BREAK
2:15 pm - 3:15 pm	**Introduction to Brain Imaging: fMRI and MEG/EEG** ***Yalda Mohsenzadeh***
3:15 pm - 3:30 pm	BREAK
3:30 pm - 4:30 pm	**Comparing Brains and DNNs: Methods and Findings** ***Martin Hebart***
4:30 pm - 4:45 pm	BREAK
4:45 pm - 5:45 pm	**Comparing Brains and DNNs: Theory of Science** ***Radoslaw Cichy***
5:45 pm - 6:00pm	Summary

The Algonauts Project

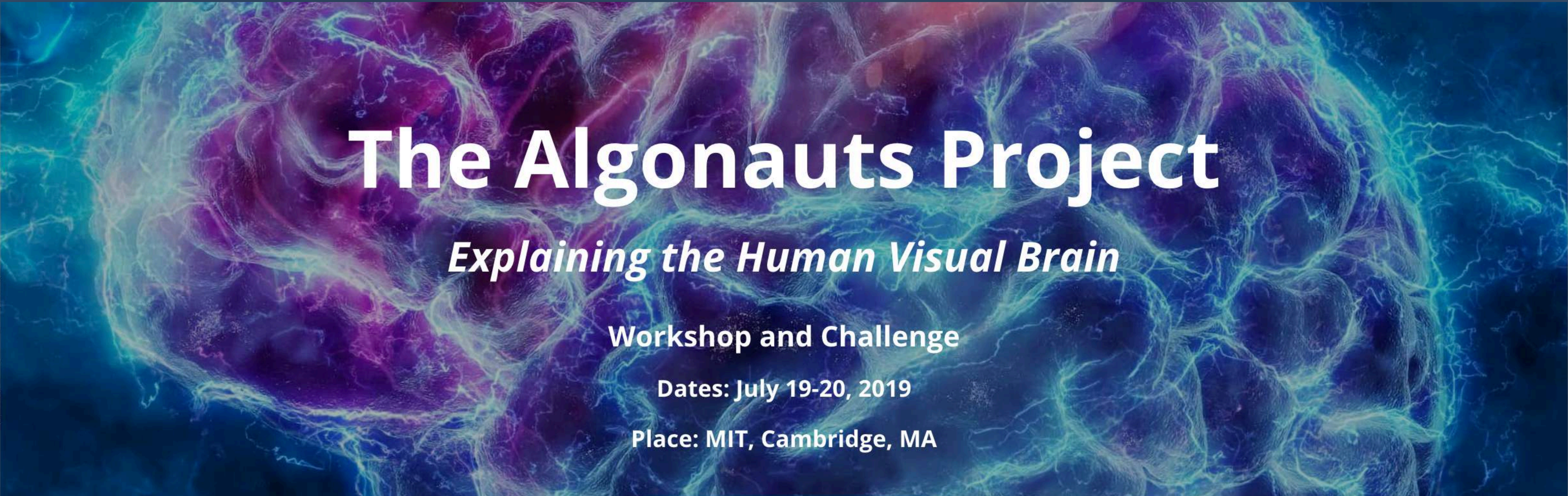
Explaining the Human Visual Brain

20 July Schedule	Event
8:30 am – 9:00 am	Breakfast
9:00 am – 9:15 am	Introduction by Radoslaw Cichy
9:15 am – 9:35 am	Matt Botvinick
9:35 am – 9:55 am	Aude Oliva
9:55 am – 10:15 am	Thomas Naselaris
10:15 am – 11:00 am	Posters and Coffee
11:00 am – 11:20 am	David Cox
11:20 am – 11:40 am	James DiCarlo
11:40 am – 12:00 pm	Kendrick Kay
12:00 pm – 1:30 pm	Lunch on Your Own
1:30 pm – 1:50 pm	Introduction to the Algonauts Challenge by Radoslaw
1:50 pm – 2:50 pm	Invited Talks: Challenge Winners
2:50 pm – 3:30 pm	Posters and Coffee
3:30 pm – 3:50 pm	Talia Konkle
3:50 pm – 4:10 pm	Nikolaus Kriegeskorte
4:10 pm – 4:30 pm	Jack Gallant
4:30 pm – 5:00 pm	Panel Discussion with Speakers
5:00 pm – 6:30 pm	Reception

Introduction to Deep Neural Networks

Tutorial

Gemma Roig



Overview

- **Introduction**
- Artificial Neural Networks
- Computational Models of Object Recognition
- Artificial Neural Networks for Object Recognition
- Applications


Alan Turing
COMPUTING MACHINERY
AND INTELLIGENCE , 1950
"Can machines think?"

Recognition

Object recognition → What is in the image?

→ Bike

→ Train

→ Bird

Recognition

We want the algorithms to **learn** to do object recognition given examples of object categories

Training phase:
The model learns with examples

Testing phase:
Automatic labelling of instances
<u>never seen before by the algorithm</u>

There are different modalities of supervision:
fully supervised, unsupervised, semi-supervised, etc.

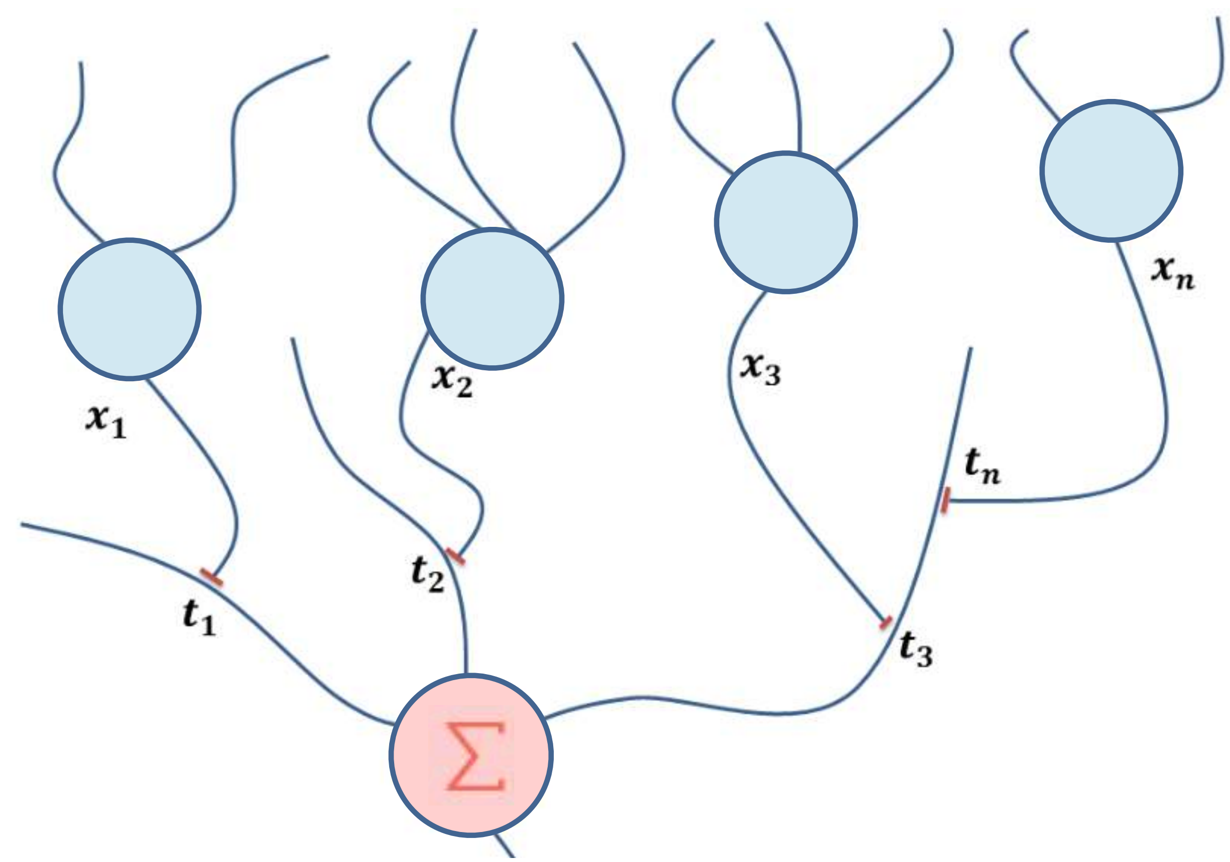
Overview

- Introduction
- **Artificial Neural Networks**
- Computational Models of Object Recognition
- Artificial Neural Networks for Object Recognition
- Applications

Computational Principles

Simplified neuroscience: a neuron computes a dot product between its inputs and the synaptic weights

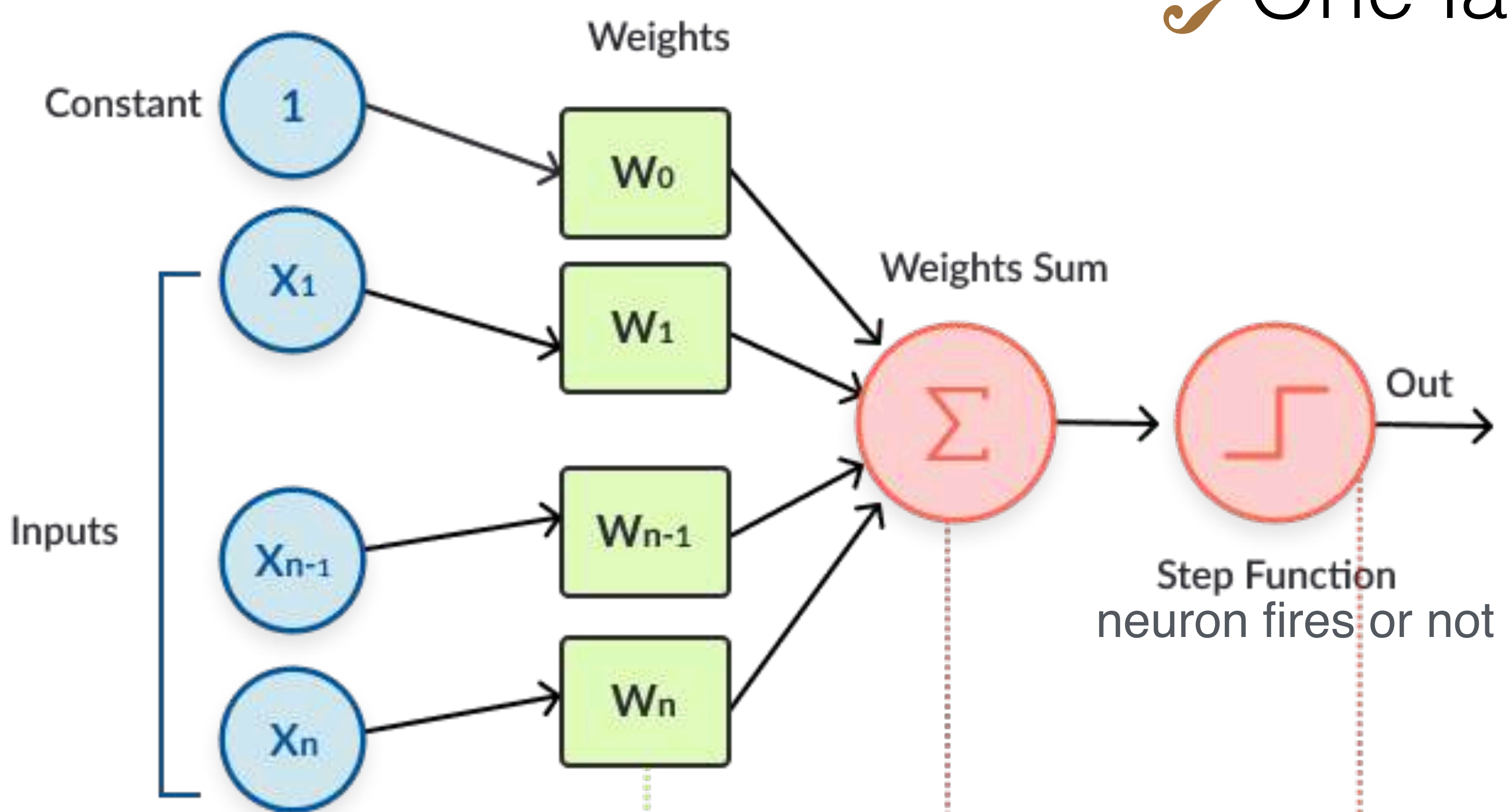
$$< x, t > = \sum_{i=1}^{n} x_i t_i$$



Simple Perceptron

F. Rosenblatt 1957

- One layer NN



Input + constant for bias

Weights Learned

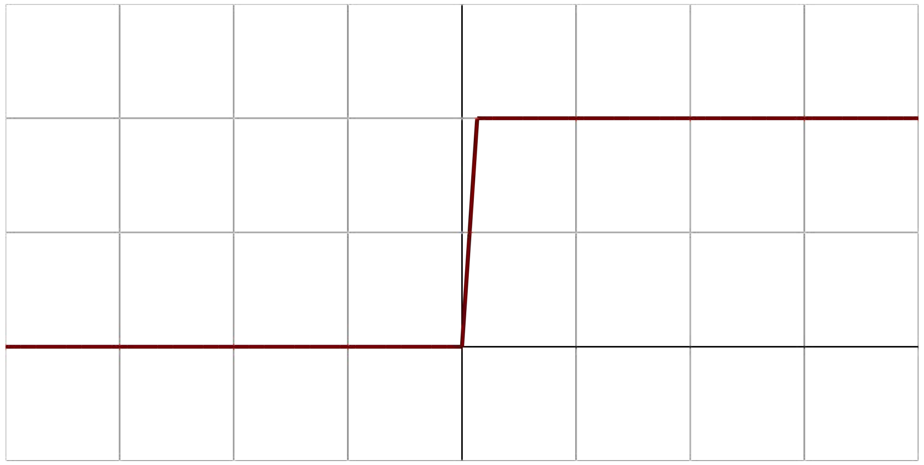
$$\sum_{i=0}^{n} x_i w_i$$

$$Out = sgn(\sum_{i=0}^{n} x_i w_i)$$

Image credit: missinglink.ai

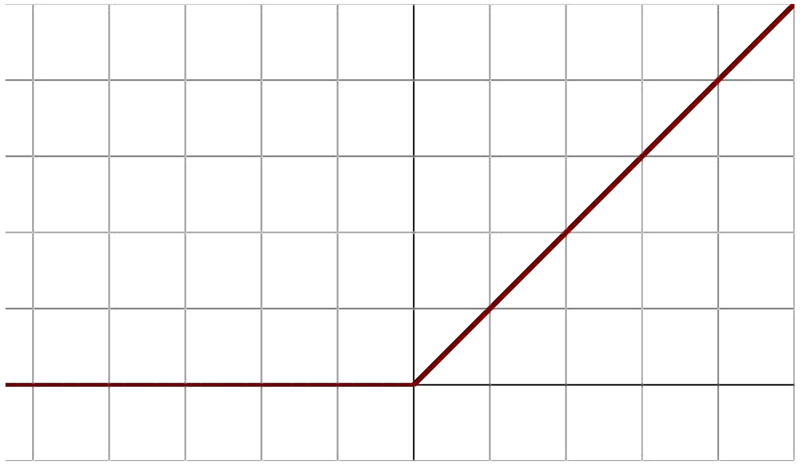
Perceptron

Types of Nonlinearities



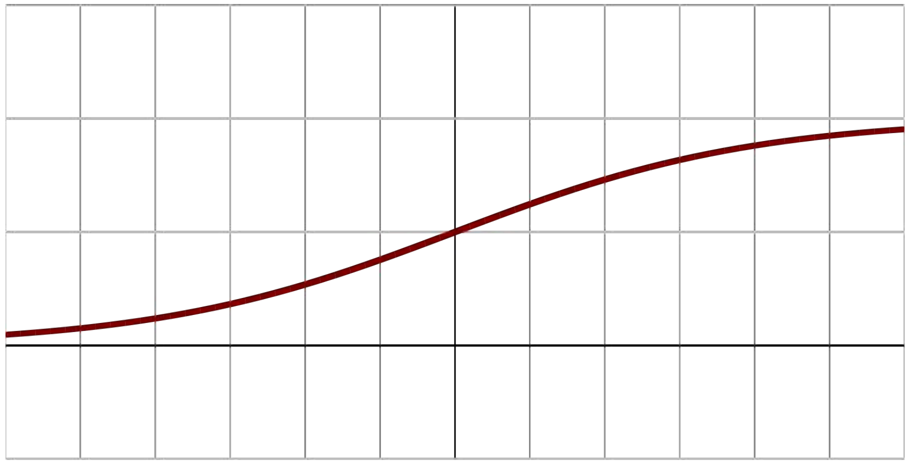
Step function

$$f(x) = \begin{cases} 0 & : x < 0 \\ 1 & : x \geq 0 \end{cases}$$



Linear Rectifier (ReLu)

$$f(x) = \begin{cases} 0 & : x < 0 \\ x & : x \geq 0 \end{cases}$$



Sigmoid

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

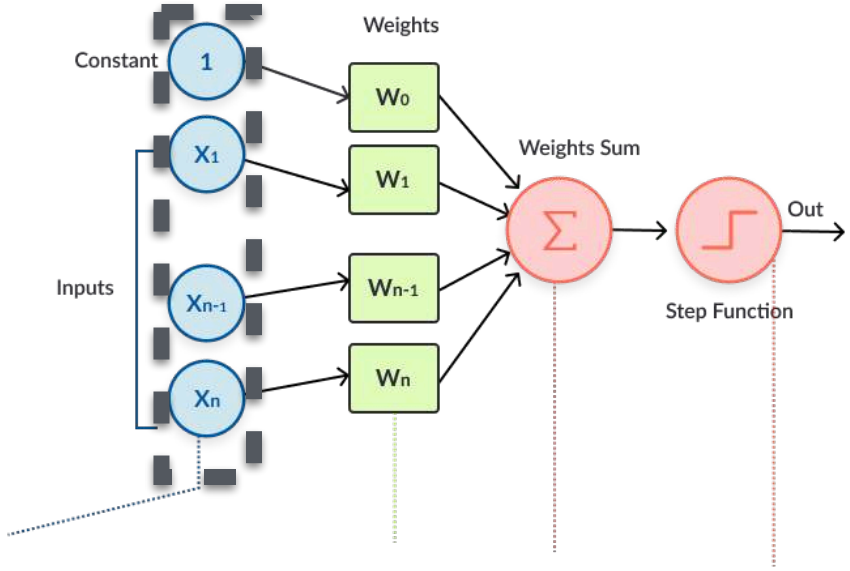
etc.

The Perceptron Learning Rule

Given training samples $\{\mathbf{x}_i, y_i\}_{\forall i}$

$\mathbf{x}_i$ -> input of example *i*,

y_i -> groundtruth target of example *i*

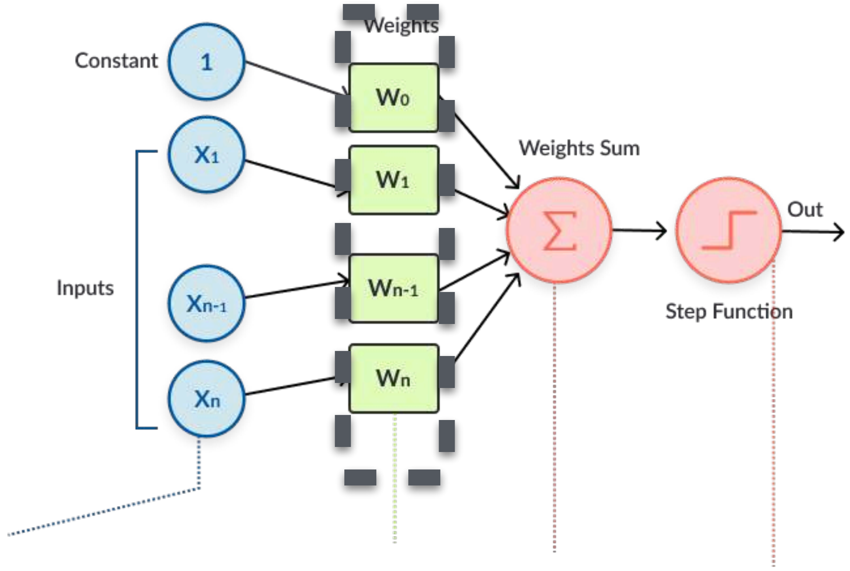


The Perceptron Learning Rule

Given training samples $\{\mathbf{x}_i, y_i\}_{\forall i}$

$\mathbf{x}_i$ -> input of example *i*,

y_i -> groundtruth target of example *i*



Initialization:

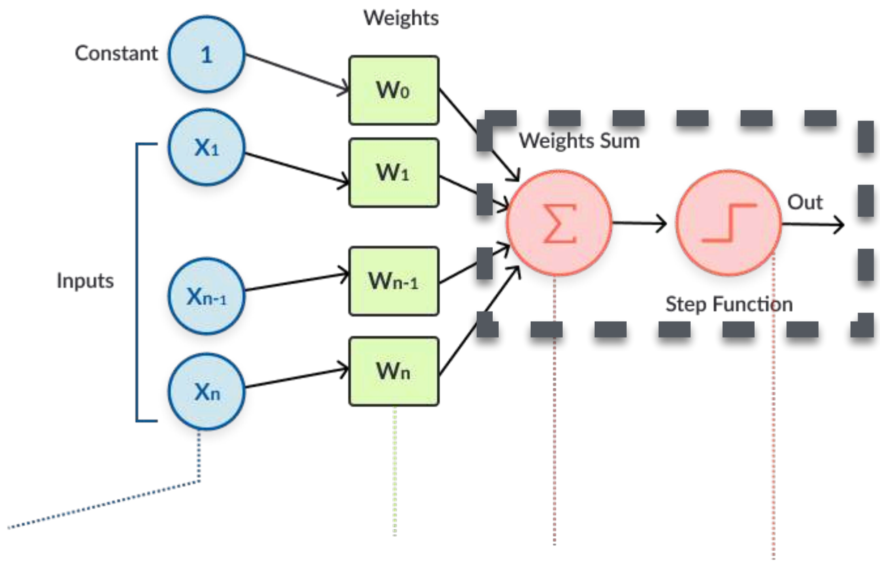
Initialize the weights w to 0 or small random numbers.

The Perceptron Learning Rule

Given training samples $\{\mathbf{x}_i, y_i\}_{\forall i}$

$\mathbf{x}_i$ -> input of example *i*,

y_i -> groundtruth target of example *i*



Initialization:

Initialize the weights w to 0 or small random numbers.

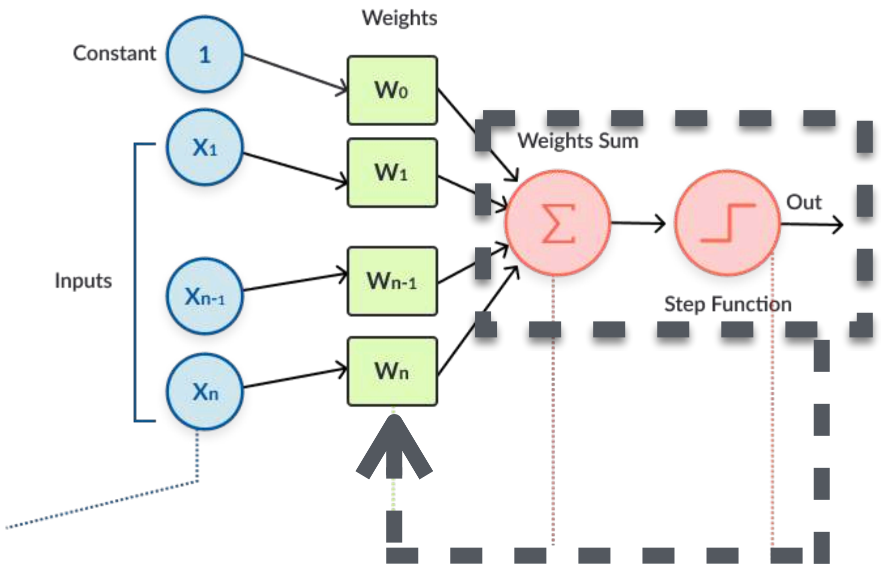
Iterate:

For each training sample $\mathbf{x}_i$:

1.Calculate the output value: $out = sgn\left(\sum_{i=0}^{n} x_i w_i\right)$

The Perceptron Learning Rule

Given training samples $\{\mathbf{x}_i, y_i\}_{\forall i}$
$\mathbf{x}_i$ -> input of example *i*,
y_i -> groundtruth target of example *i*



Initialization:

Initialize the weights w to 0 or small random numbers.

Iterate:

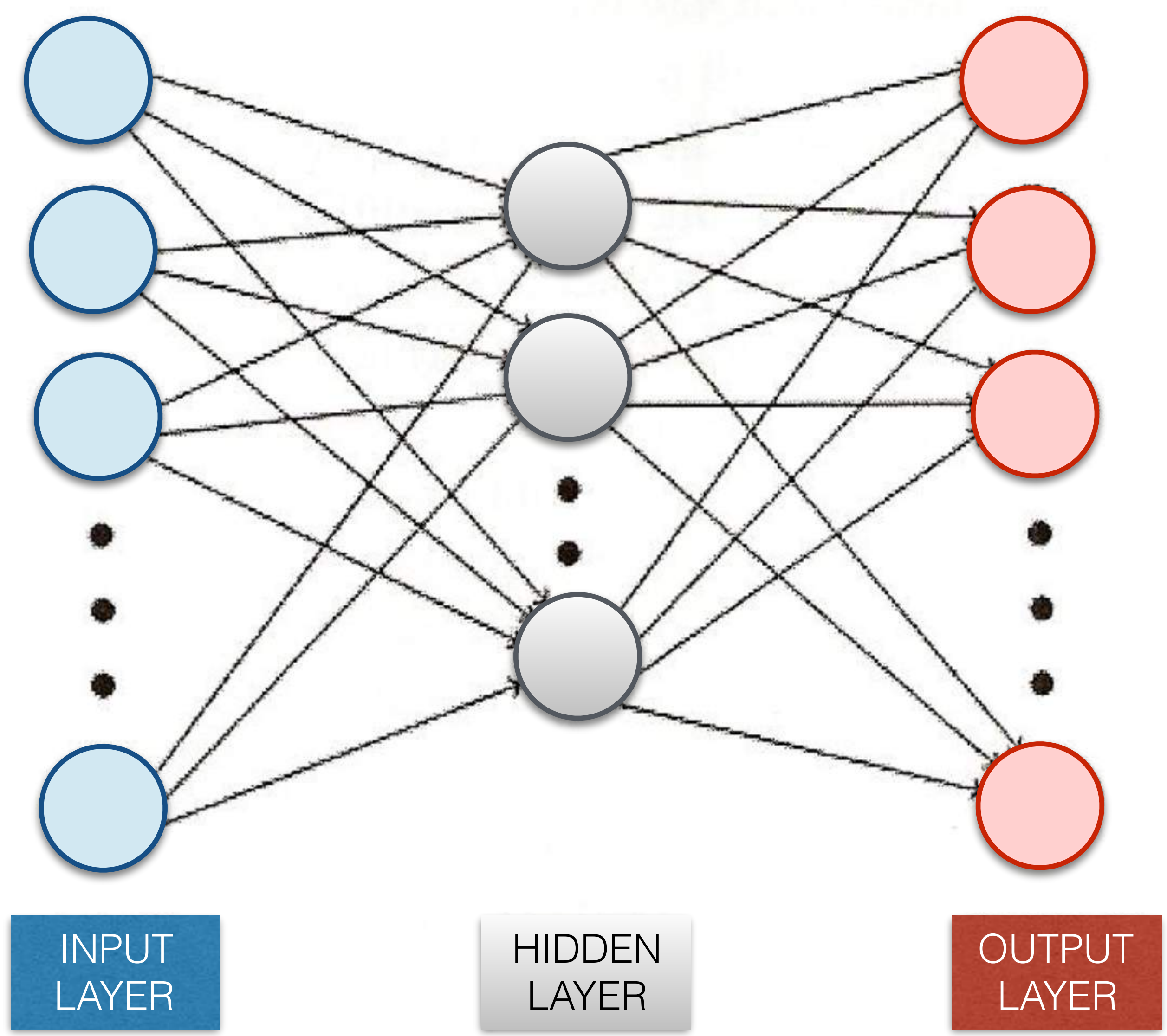
For each training sample $\mathbf{x}_i$:

1. Calculate the output value: $out = sgn\left(\sum_{i=0}^{n} x_i w_i\right)$

2. Update the weights. $\mathbf{w} = \mathbf{w} + \eta\, \mathbf{x_i}\, (y_i - out)$

Multi-layer Perceptron

Rumelhart et al. 1986

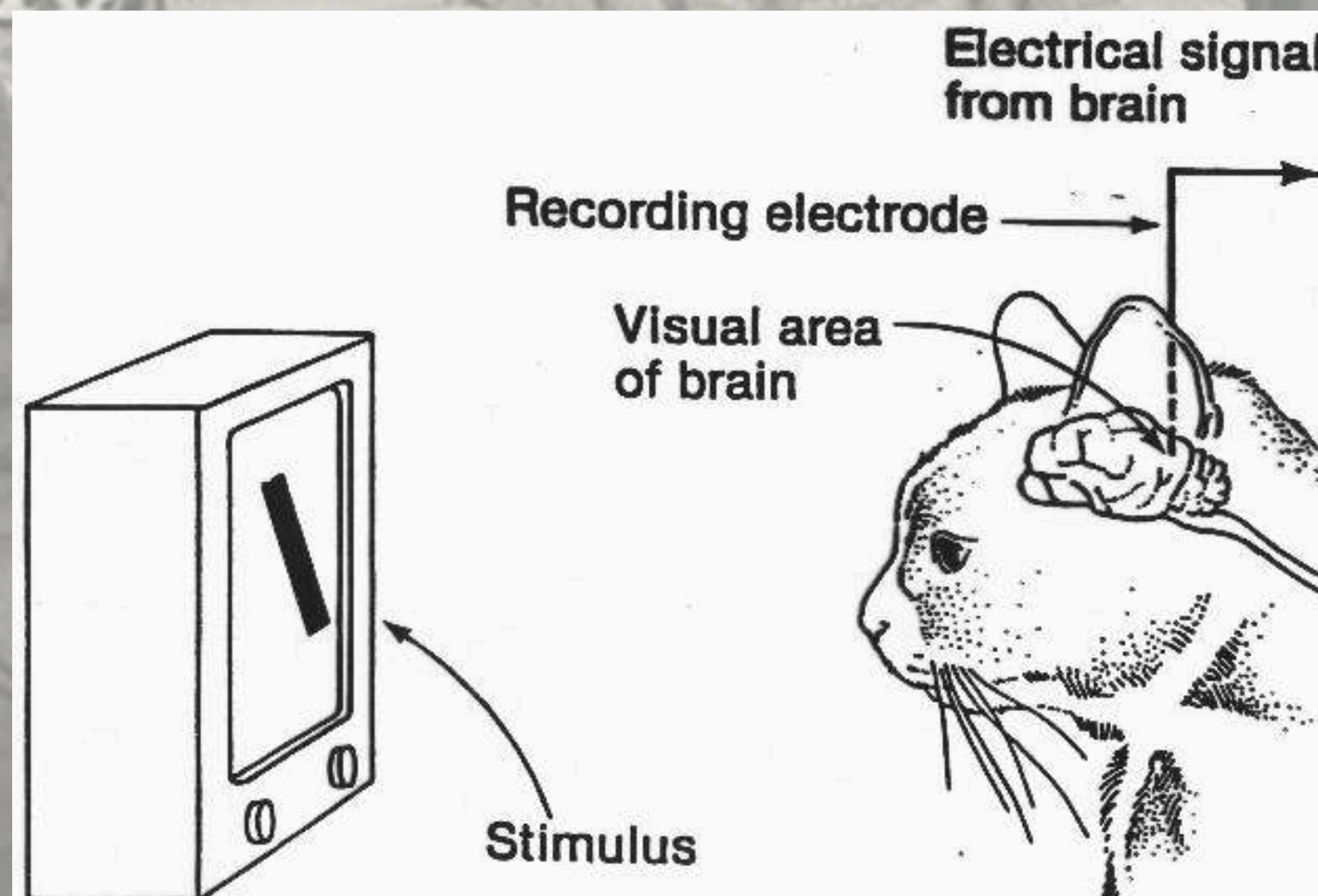


possibly many more layers → learning with back-propagation

Overview

- Introduction
- Artificial Neural Networks
- **Computational Models of Object Recognition**
- Artificial Neural Networks for Object Recognition
- Applications

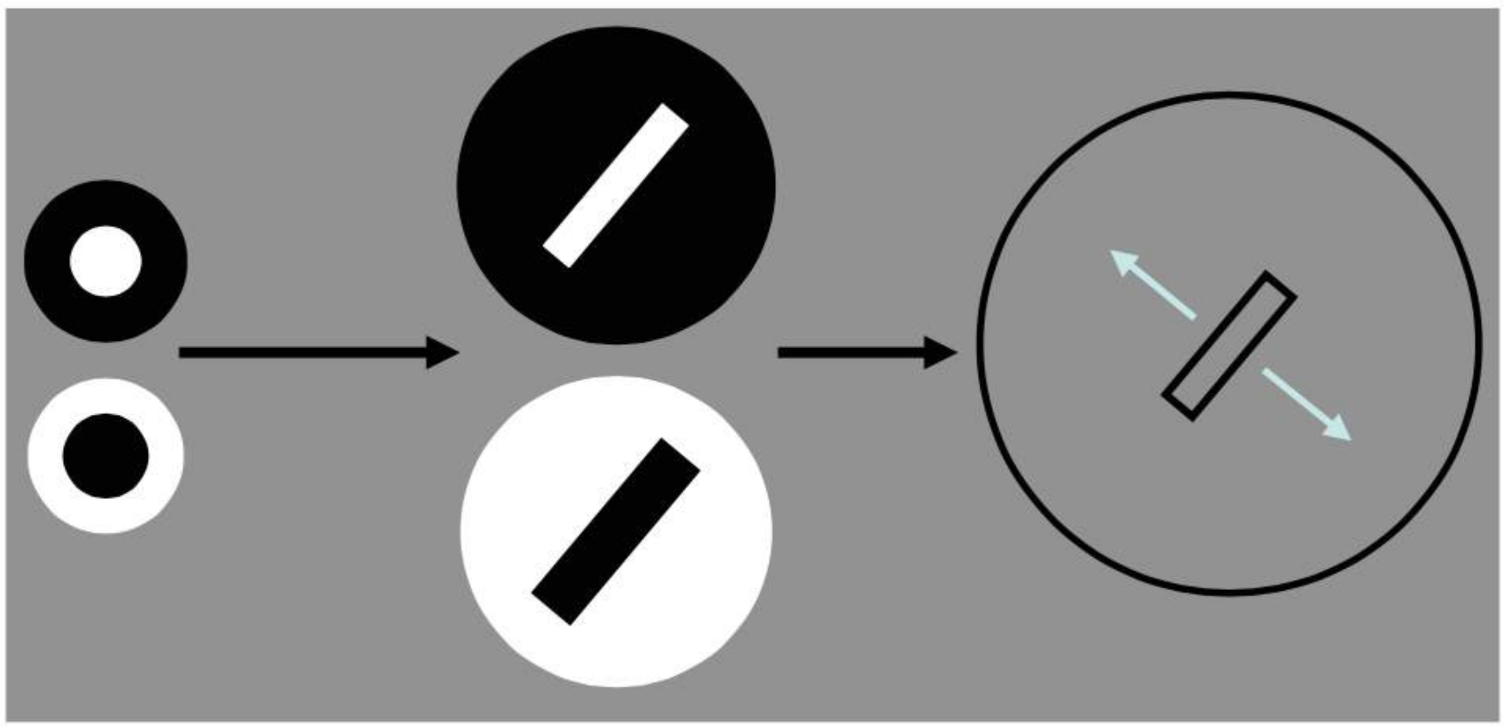
Hubel and Wiesel

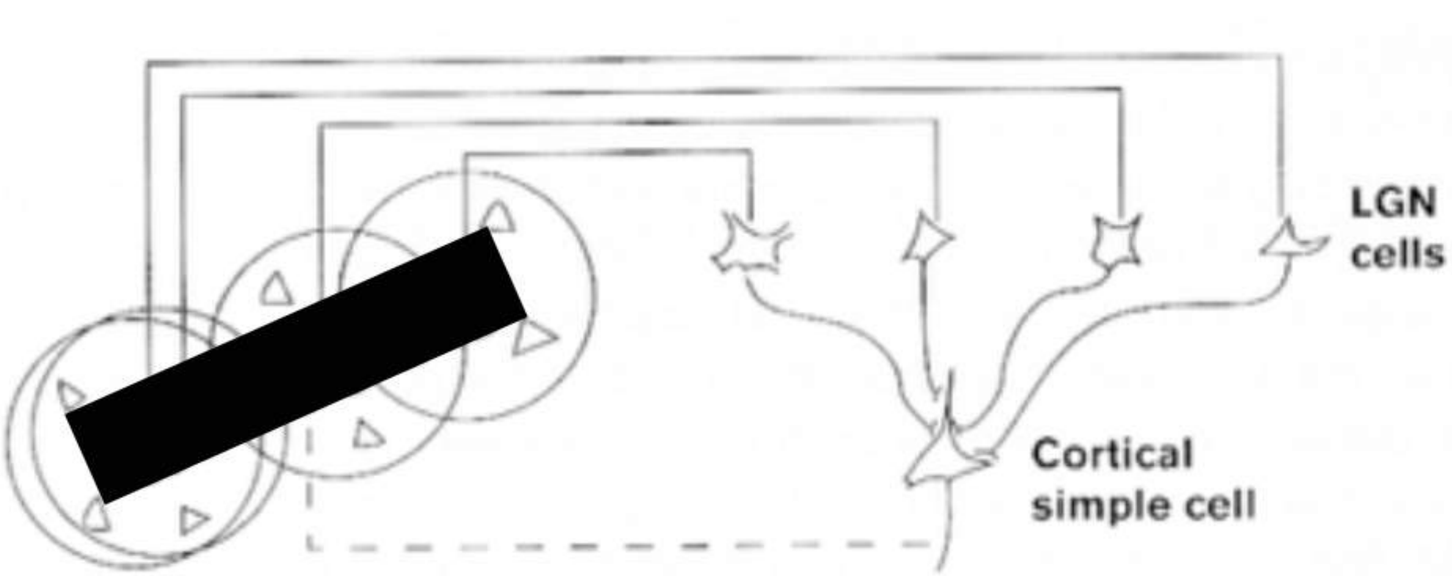


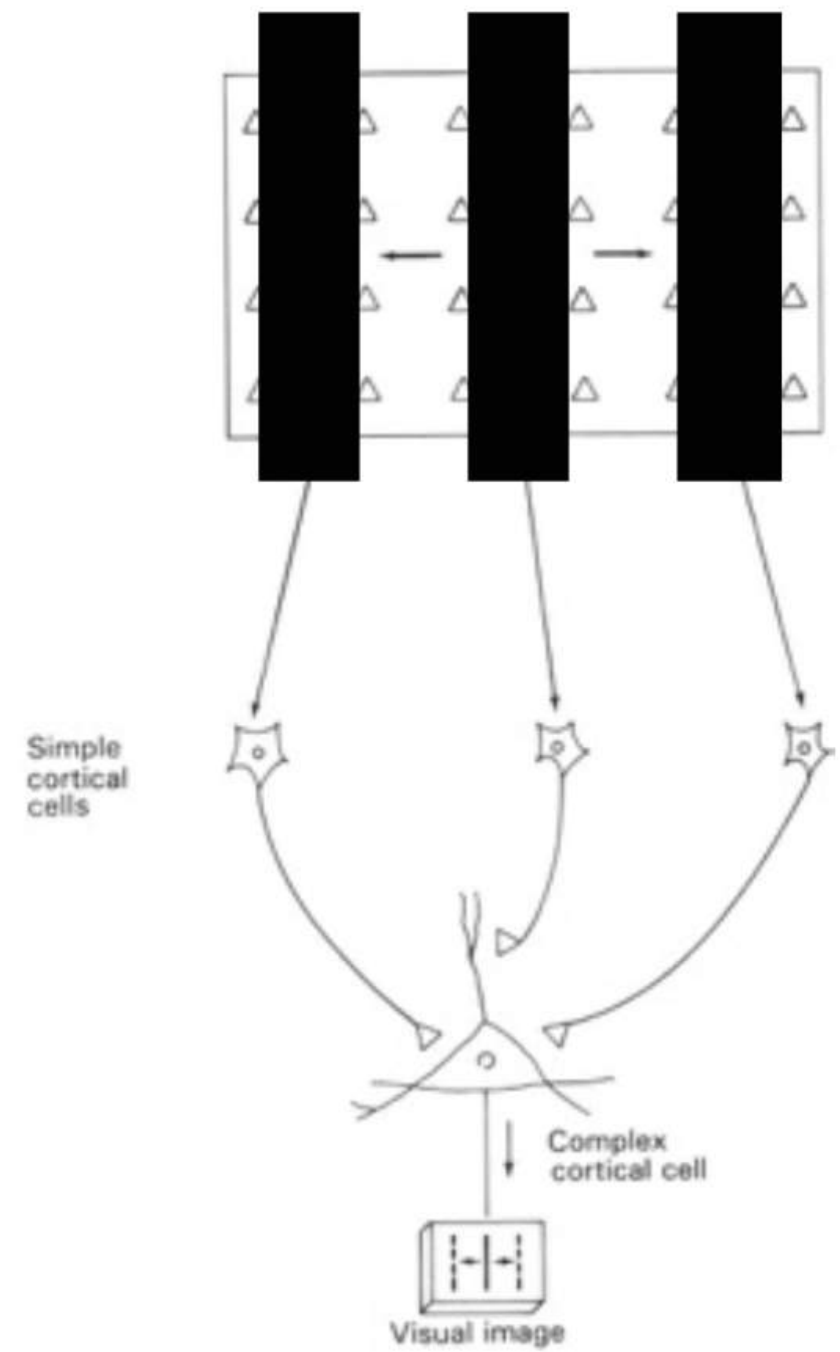
Nobel prize
(1981)

Hubel and Wiesel

LGN–type cells | Simple cells | Complex cells

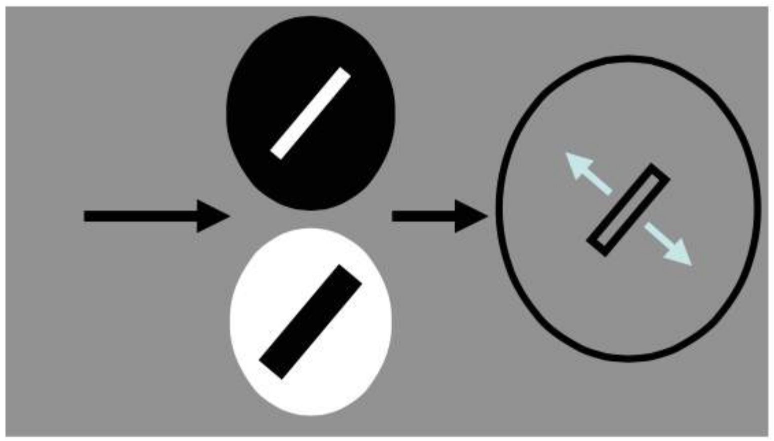


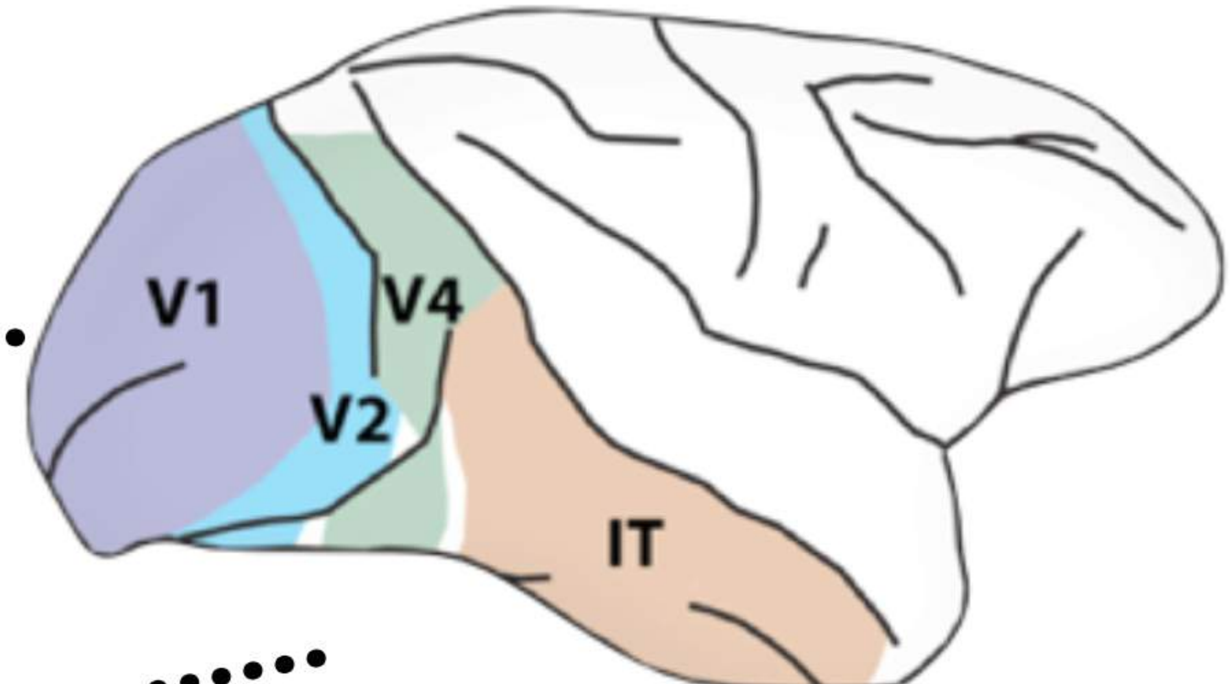


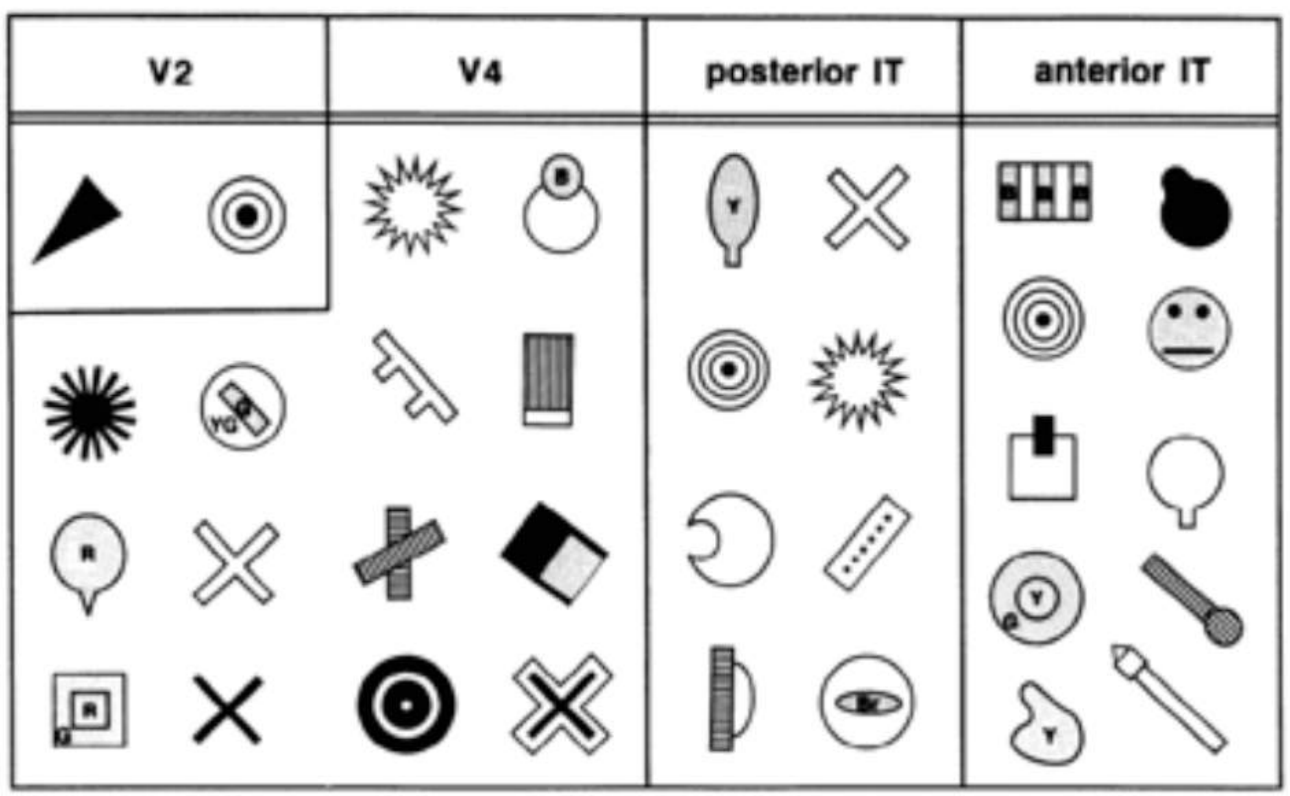


(Hubel & Wiesel 1959)

The visual ventral stream





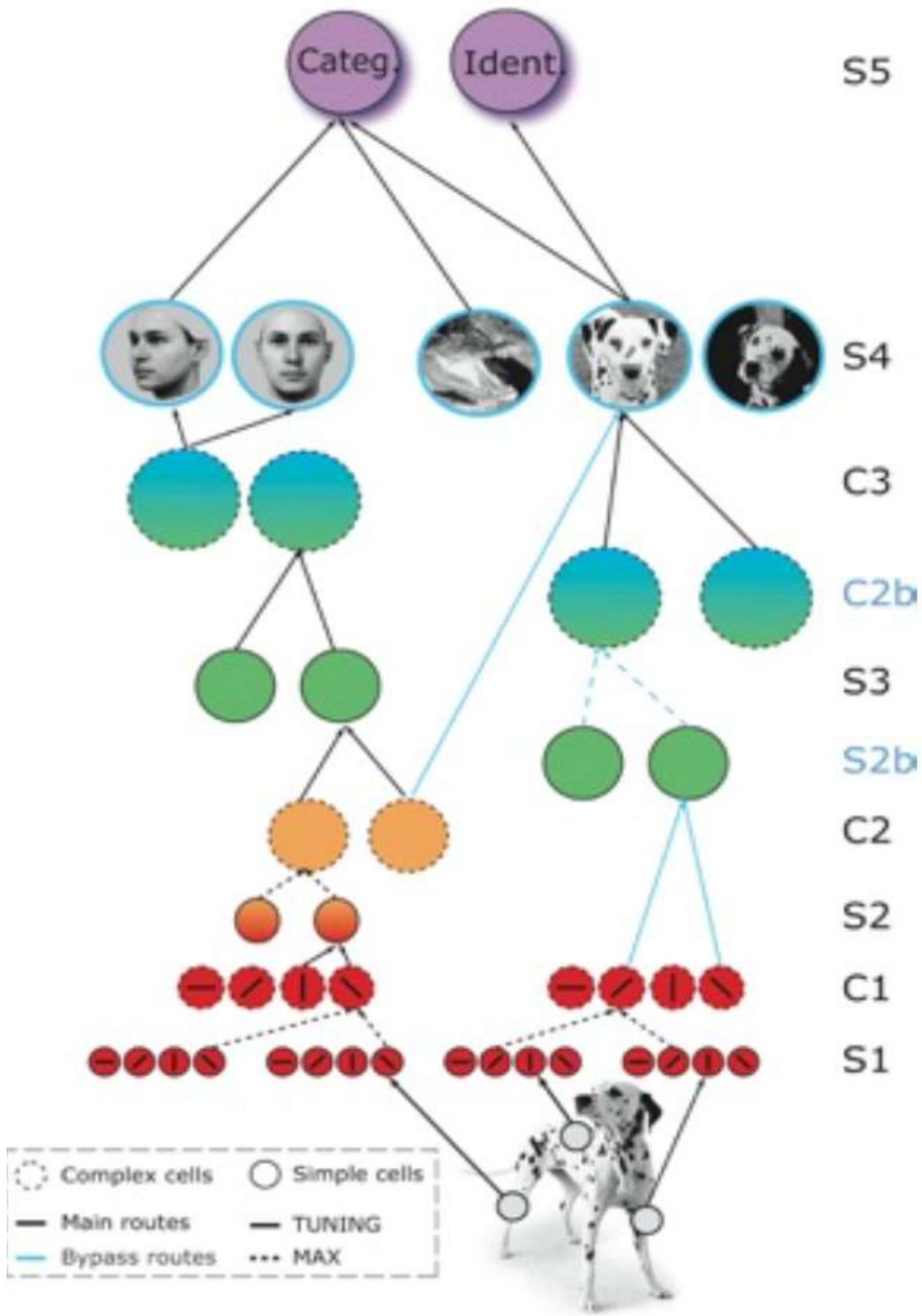


The ventral stream hierarchy: V1, V2, V4, IT

A gradual increase in the receptive field size, in the complexity of the preferred stimulus, in tolerance to position and scale changes

Kobatake & Tanaka, 1994

HMAX



Riesenhuber & Poggio 1999, 2000; Serre Kouh Cadieu Knoblich Kreiman & Poggio 2005; Serre Oliva Poggio 2007

Two operations (~OR, ~AND): disjunctions of conjunctions

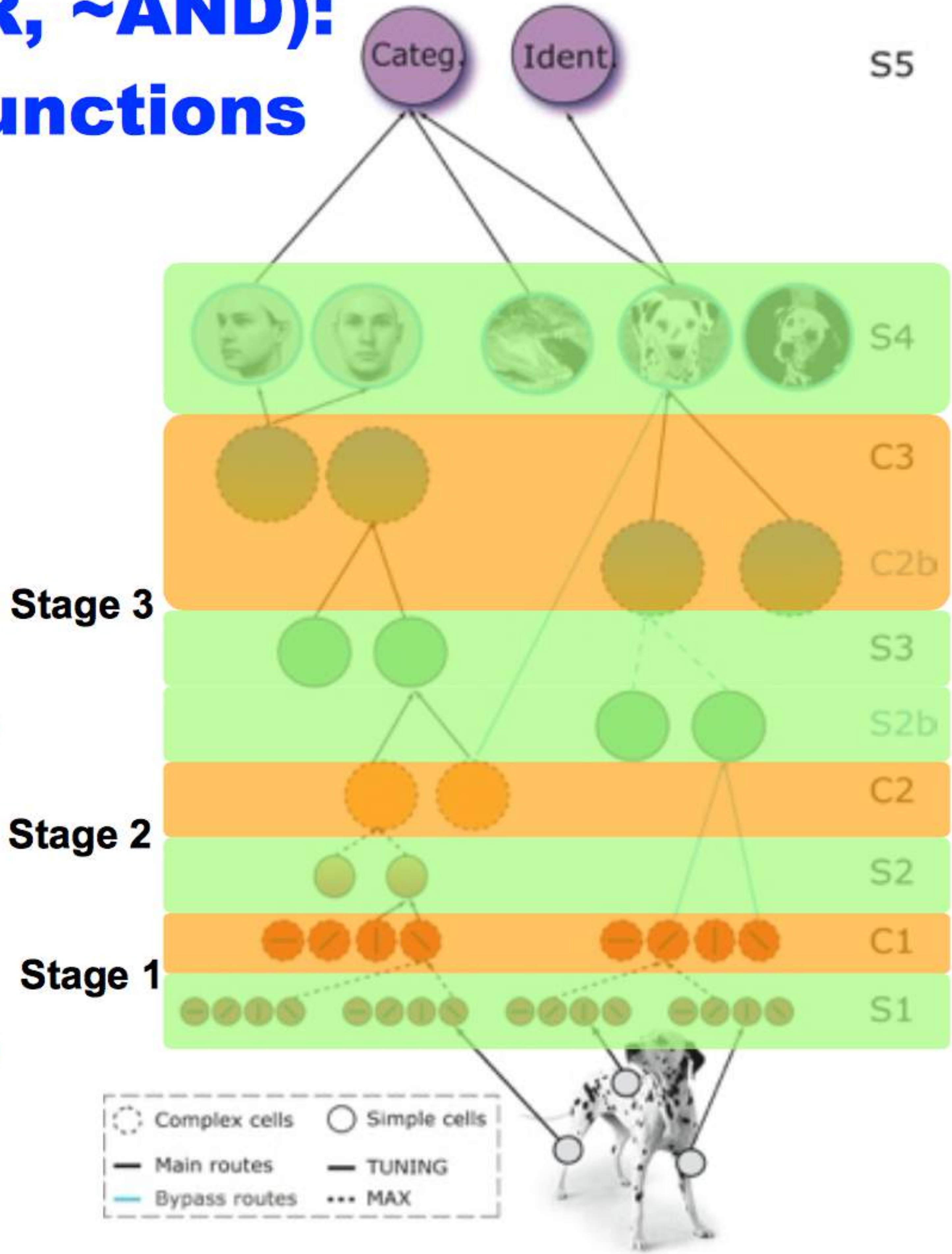
- Tuning operation (Gaussian-like, AND-like) $y = e^{-|x-w|^2}$ *or* $y \sim \frac{x \bullet w}{|x|}$
- Simple units

- Max-like operation (OR-like)

$$y = \max\{x1, x2, \ldots\}$$

- Complex units

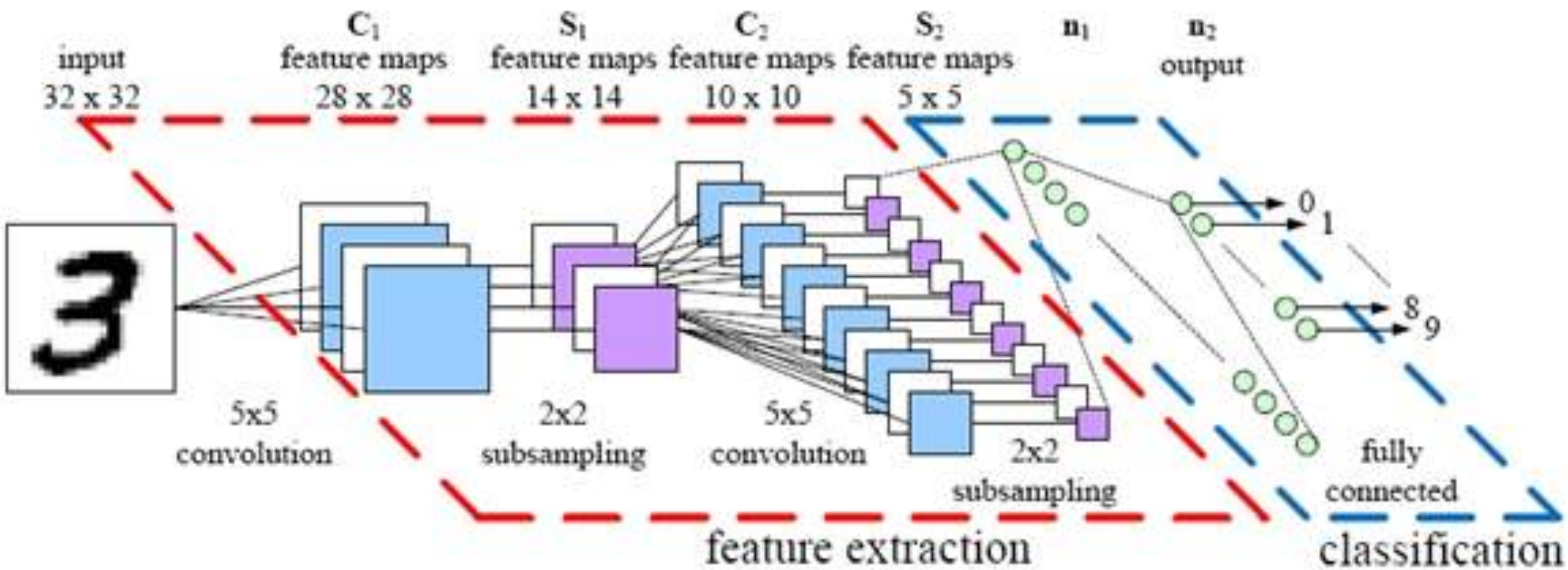
Each operation ~microcircuits of ~100 neurons



Overview

- Introduction
- Artificial Neural Networks
- Computational Models of Object Recognition
- **Artificial Neural Networks for Object Recognition**
- Applications

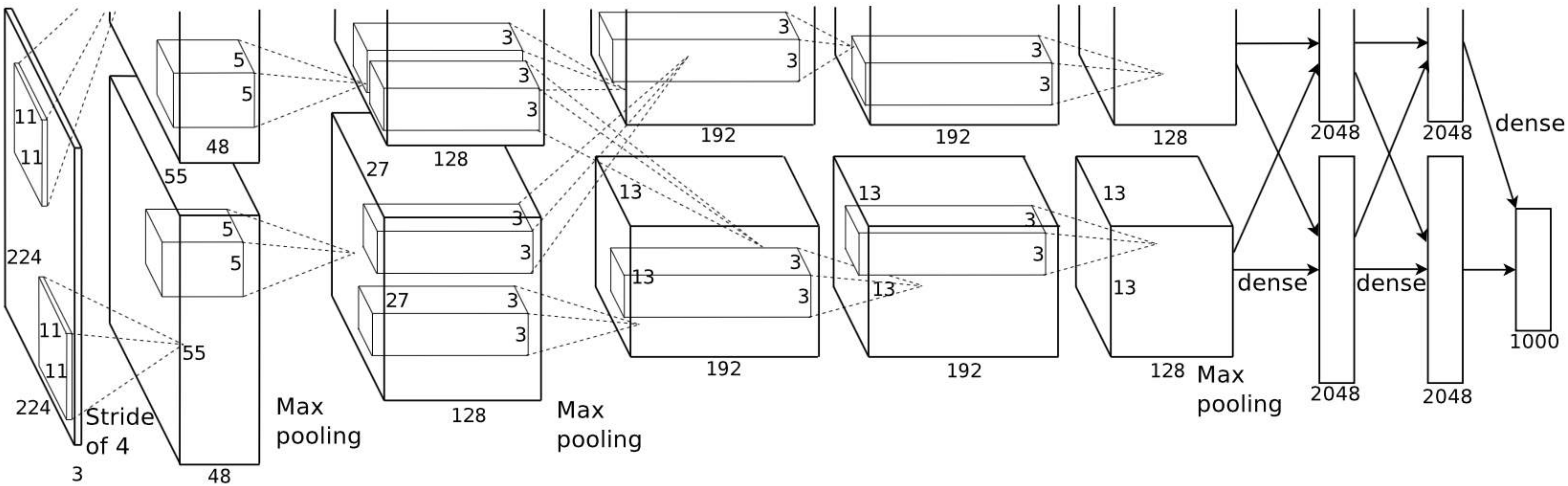
Convolutional Neural Networks (CNNs)



Convolutional assumption

LeCun et al. 98

Deep CNN (2012)



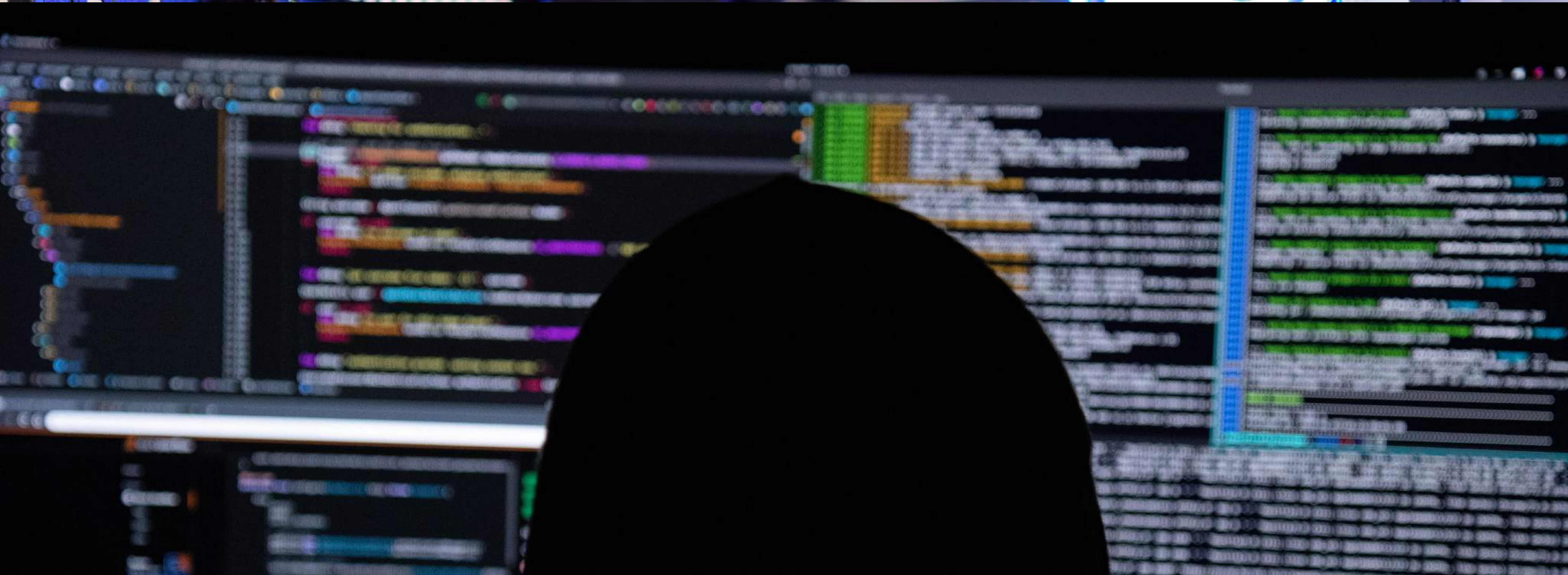
Learned with back propagation on GPUs (7 days)

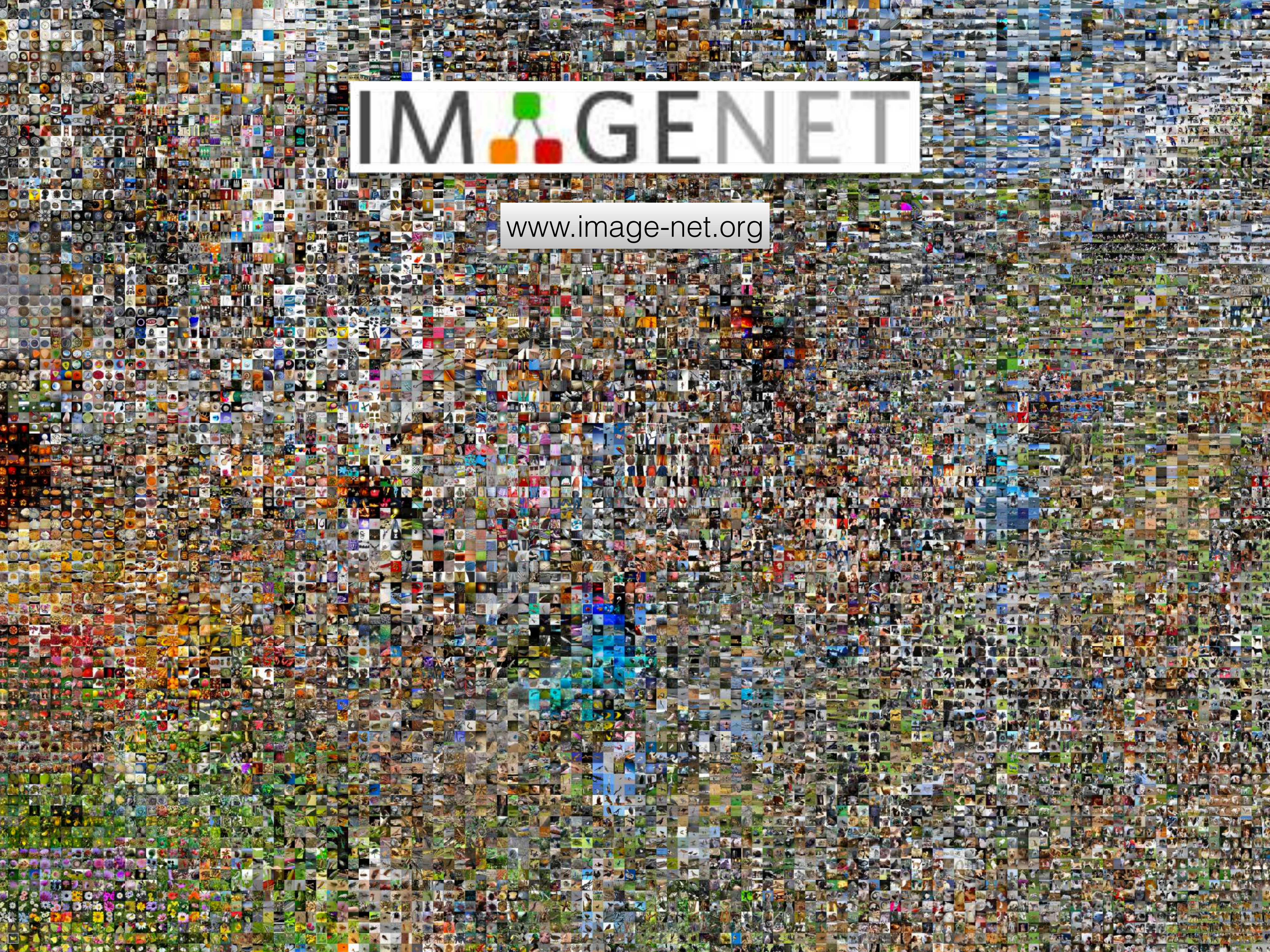
ImageNet dataset (1 million labeled images available)

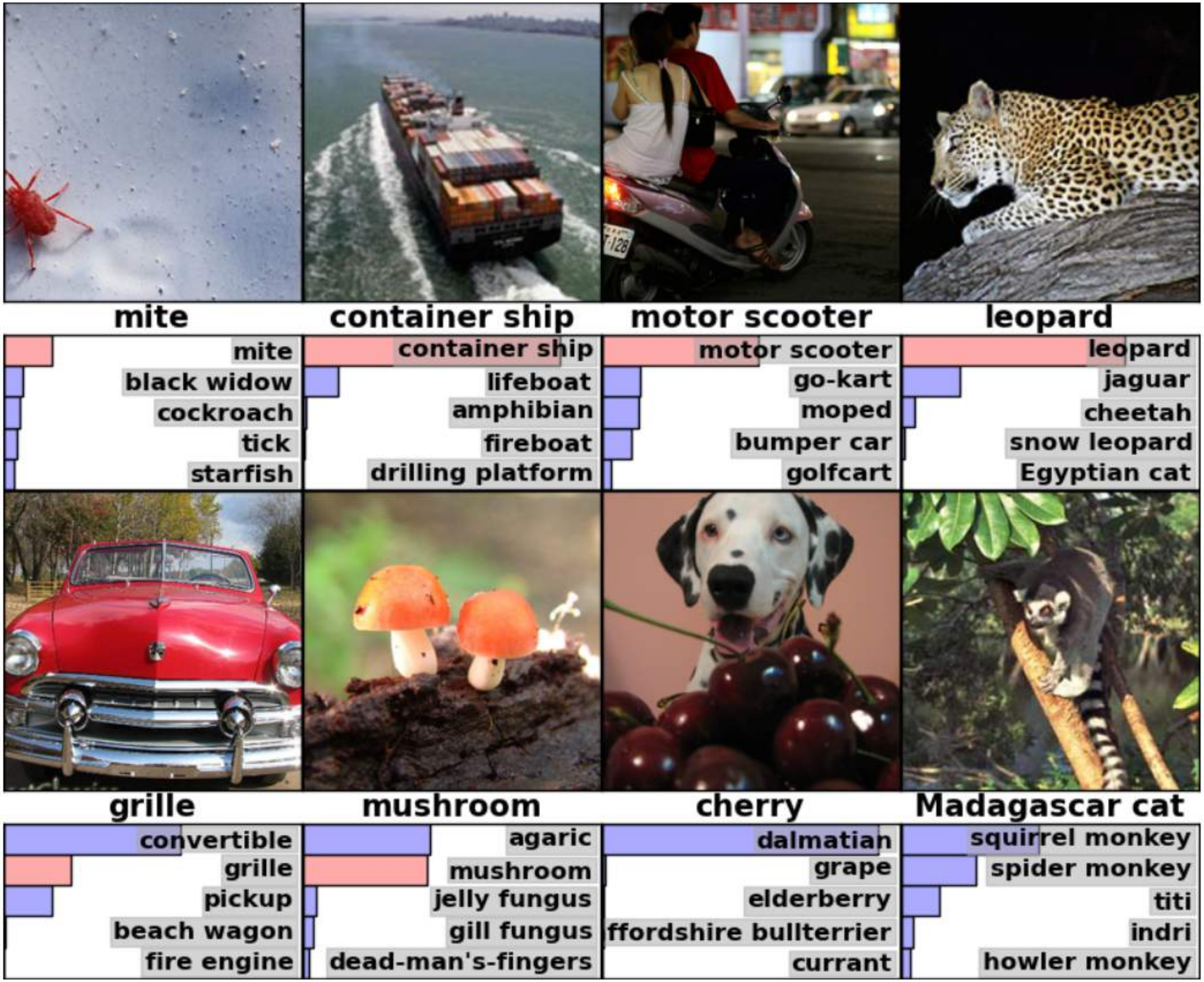
Techniques to avoid overfitting

Krizhevsky et al. 12

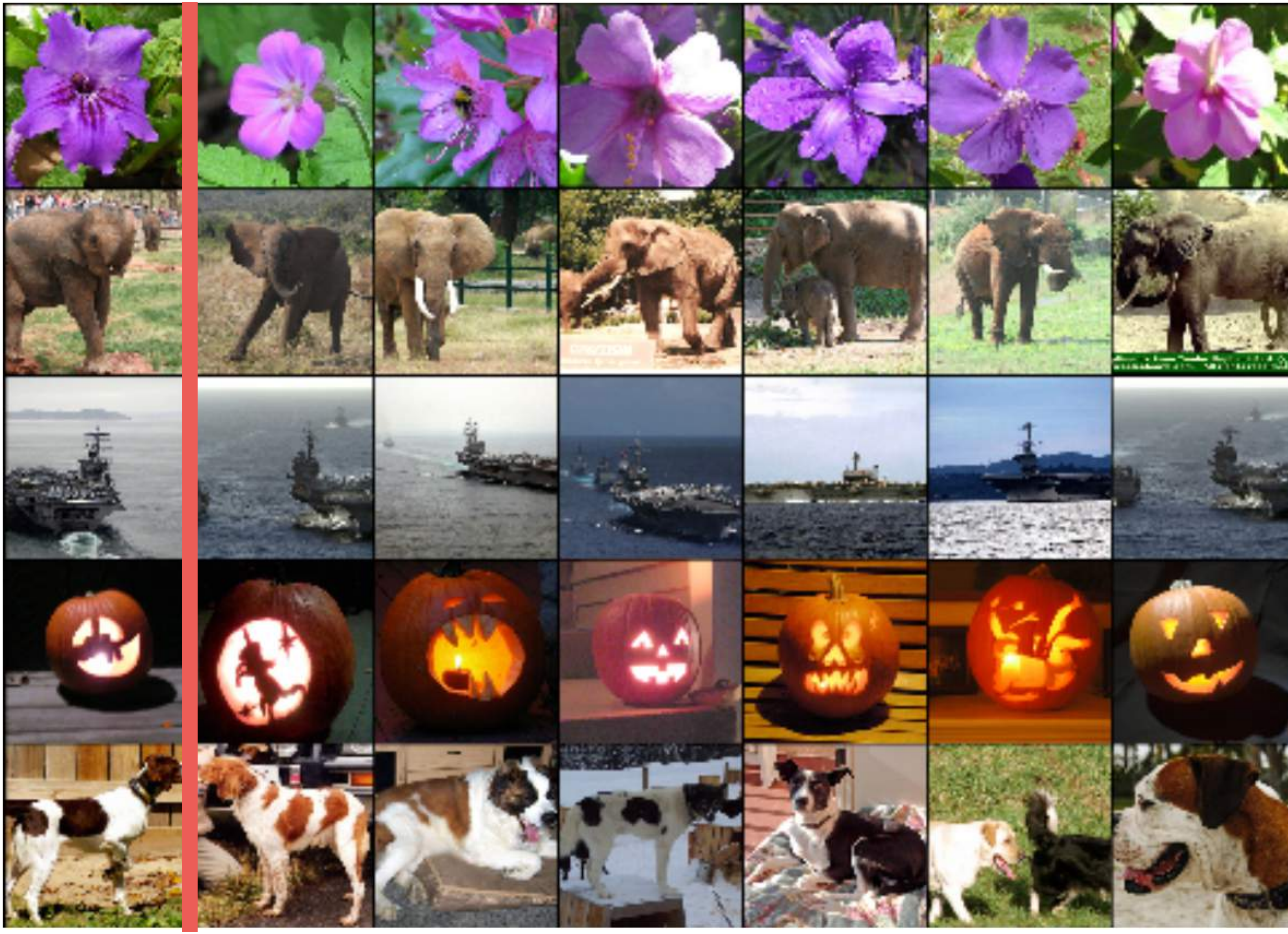
Learned with back propagation on GPUs (7 days)




IMAGENET
www.image-net.org



Krizhevsky et al. 12



Krizhevsky et al. 12

Results on ImageNet

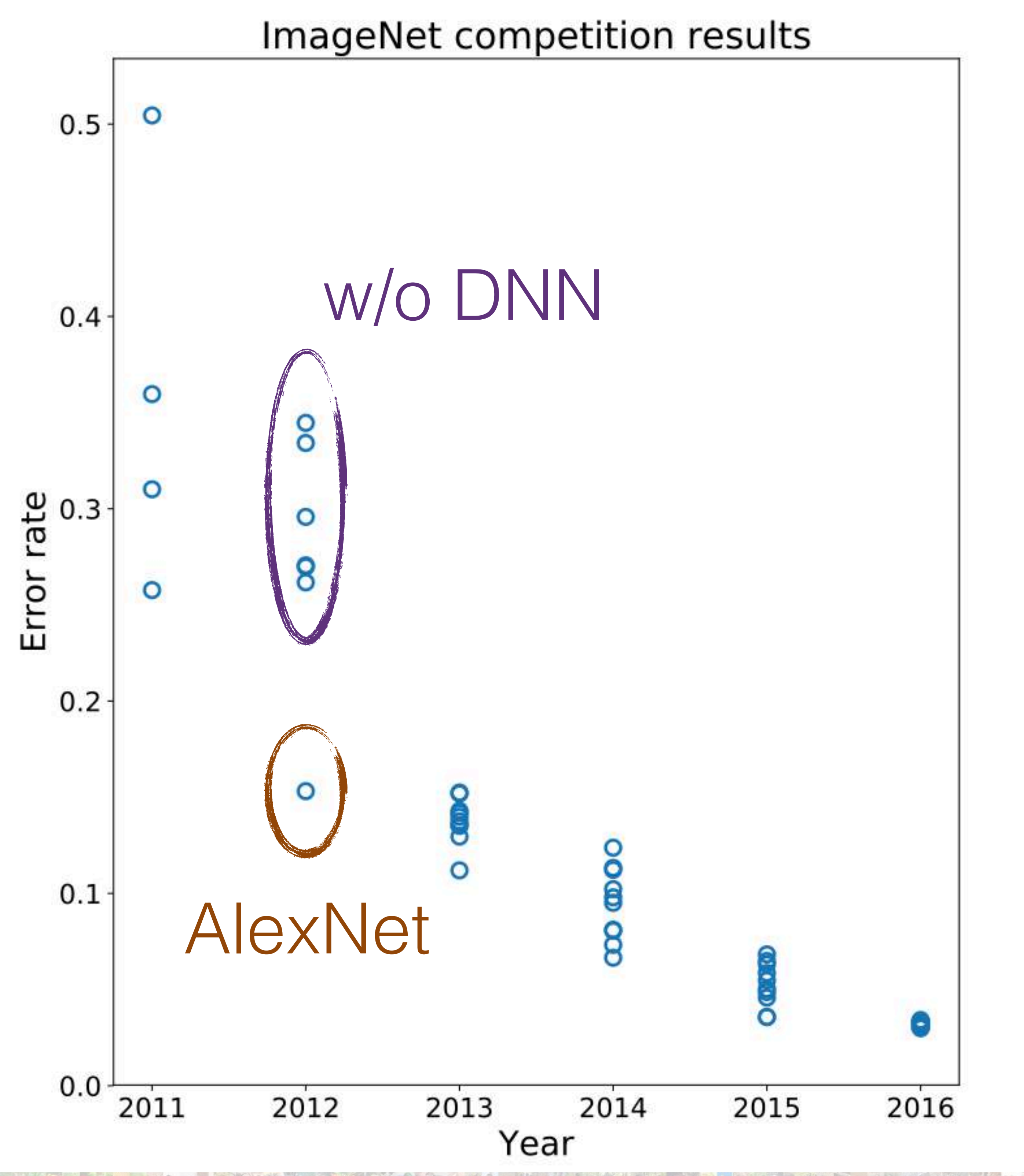


Image credit: wikipedia

Results on ImageNet

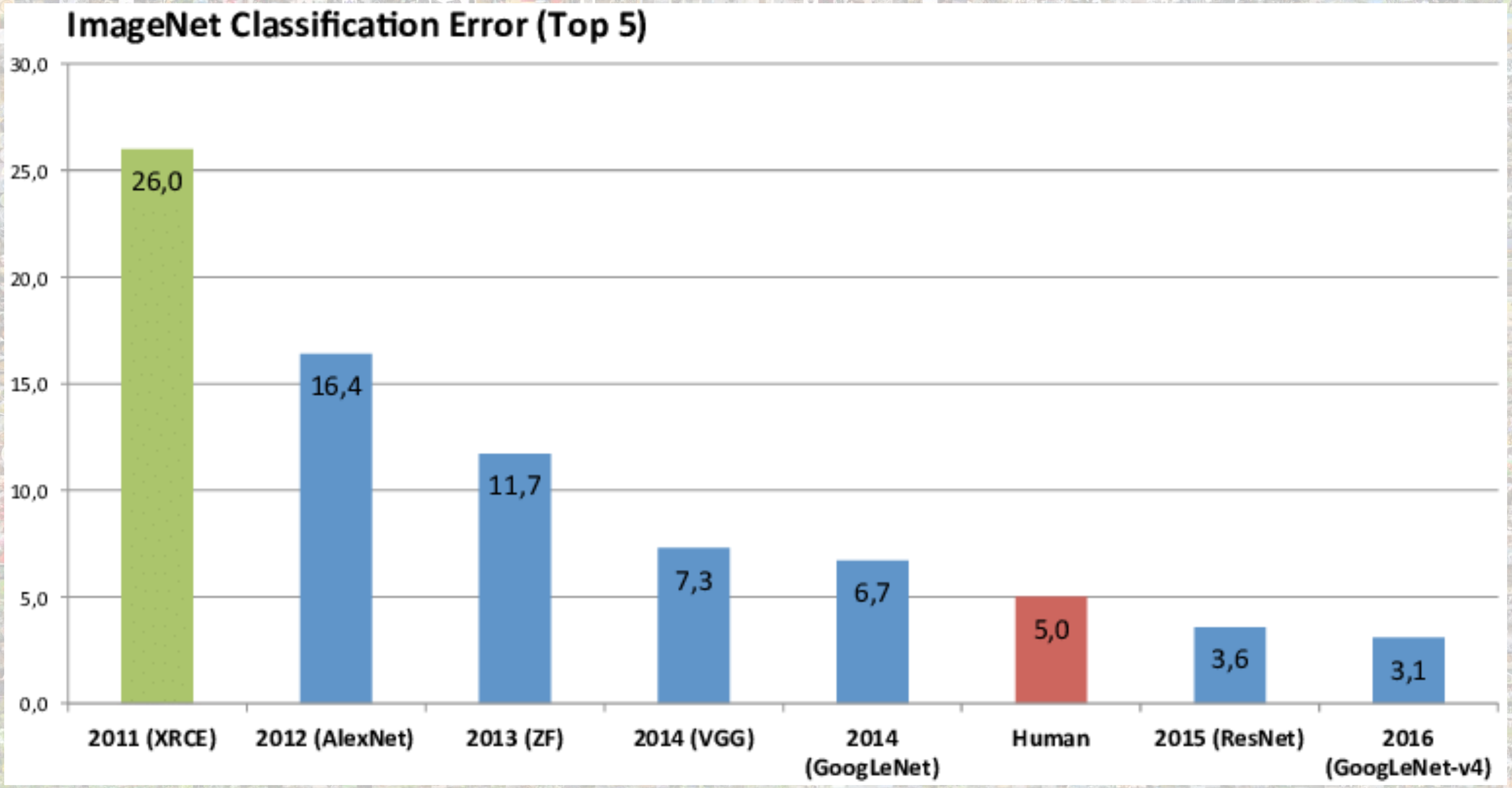
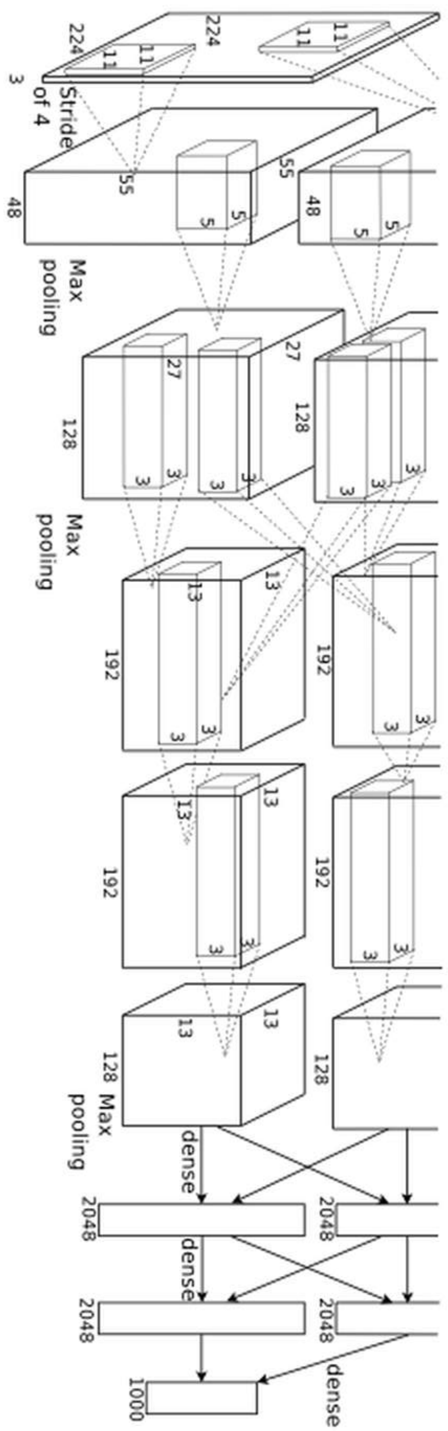
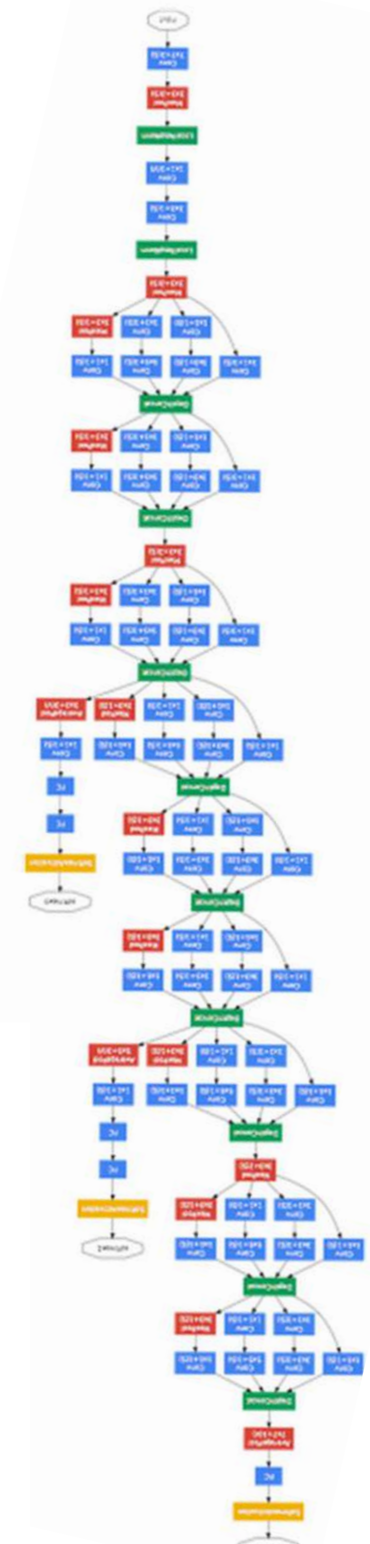
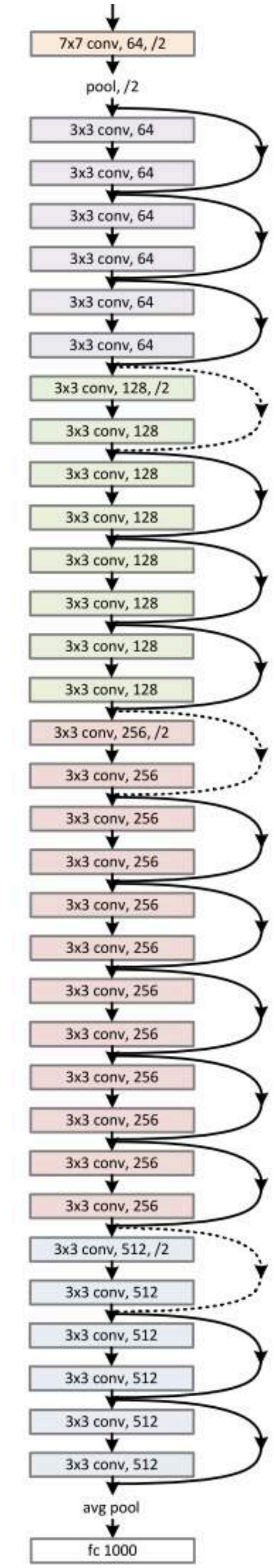


Image credit: von Zitzwewitz, 2017

Object classification

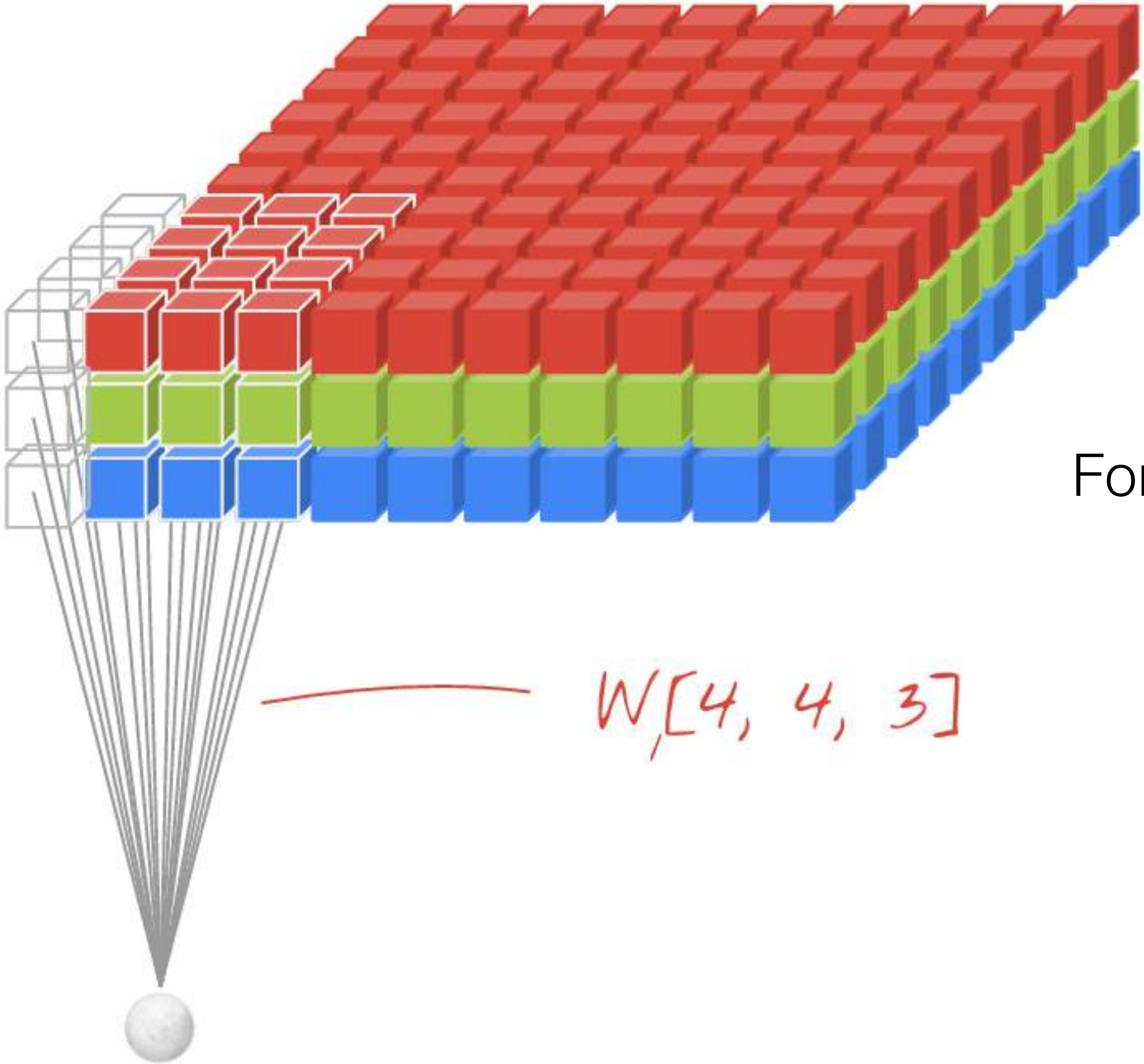






AlexNet 12 VGG 14 GoogLeNet 14 ResNet 15

Convolution 1st Layer



$$\sum_{i=1,j=1,c=1}^{i=4,j=4,c=3} x^{p}_{i,j,c} w^{k}_{i,j,c}$$

For each image patch p, $\mathbf{x}^p$
and kernel k, $\mathbf{w}^k$

Image credit: codelabs google

Convolution in Deeper Layers

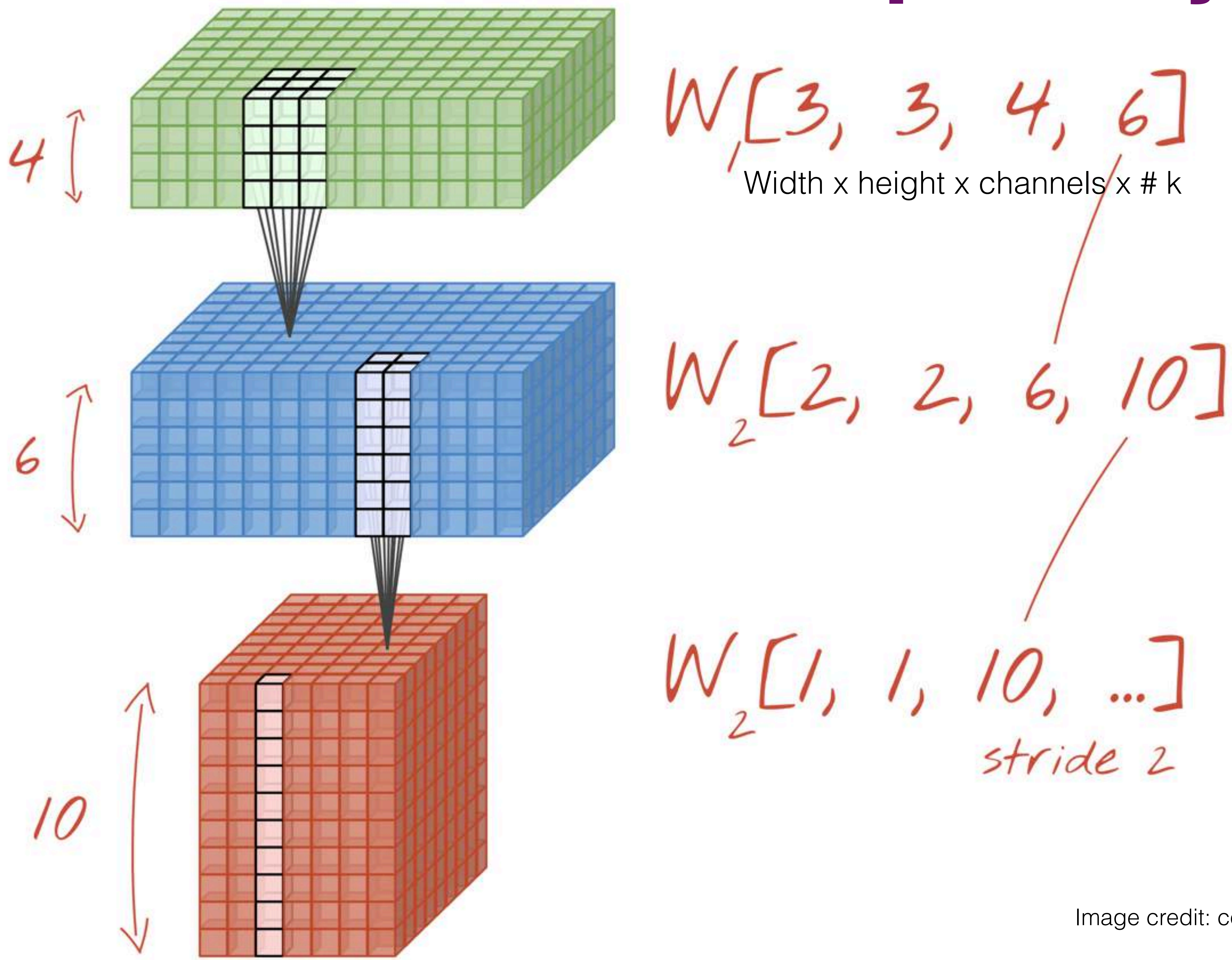


Image credit: codelabs google

Max or Average Pooling

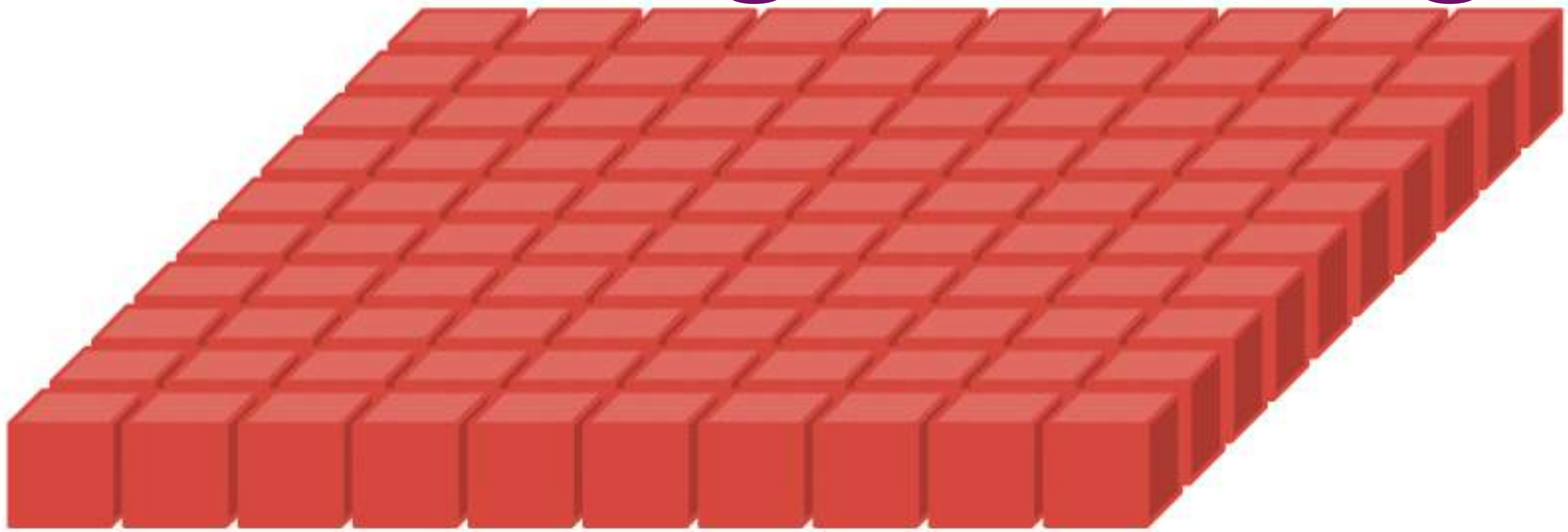


Image credit: codelabs google

Max or Average Pooling

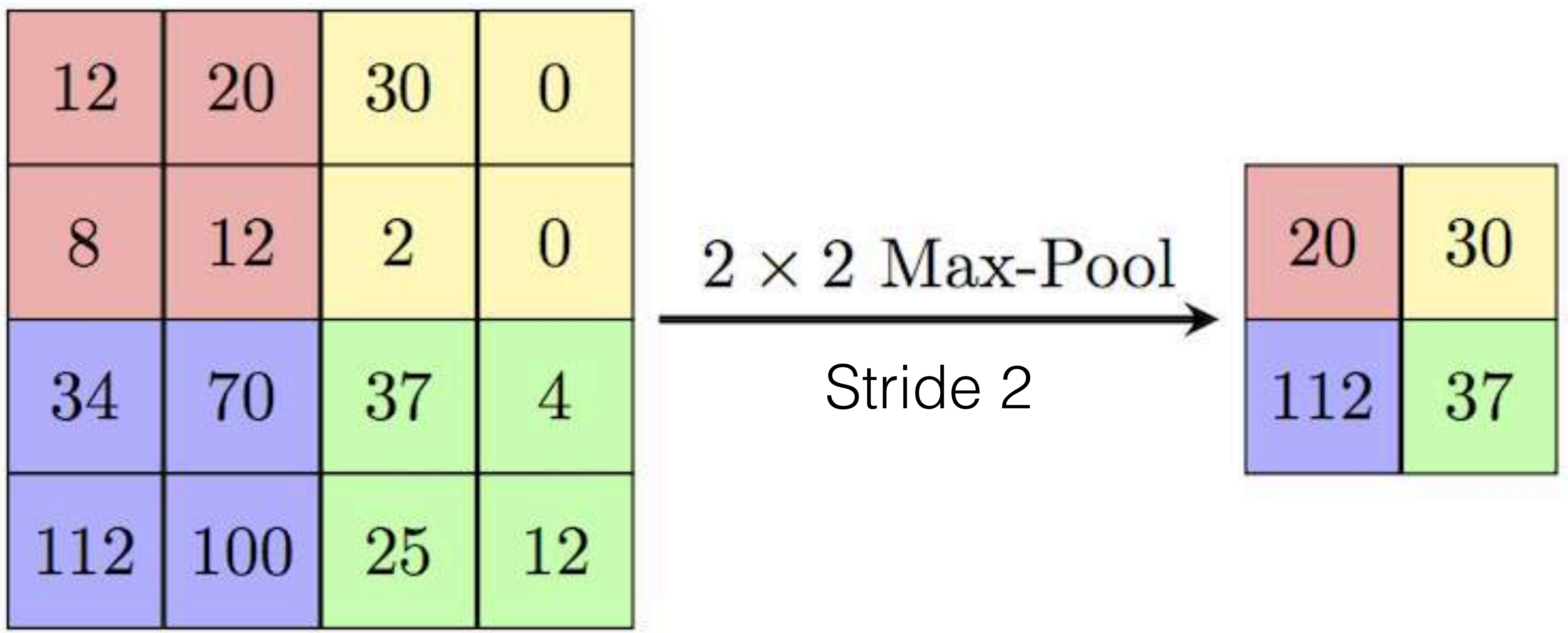


Image credit: computersciecewiki.org

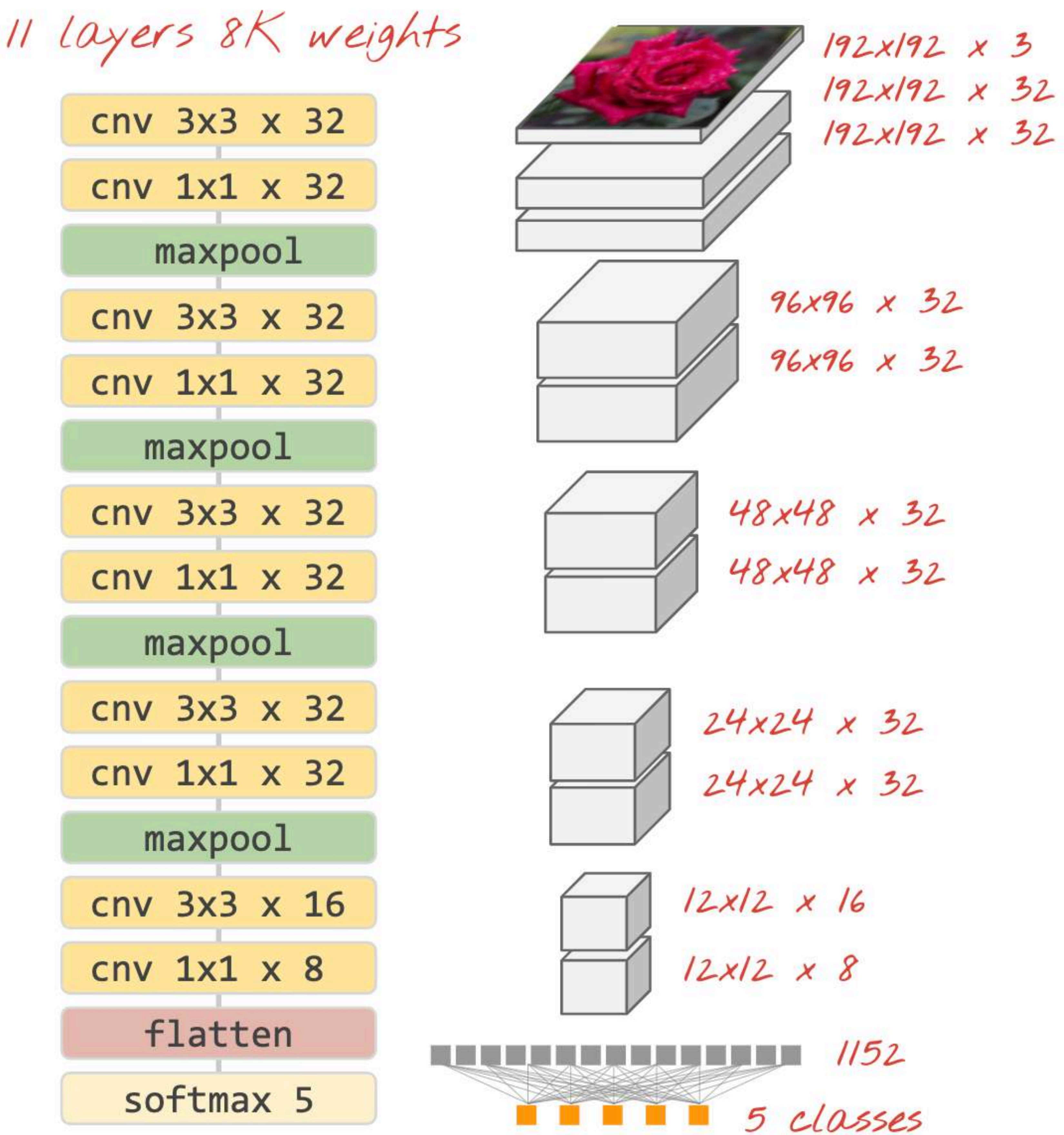


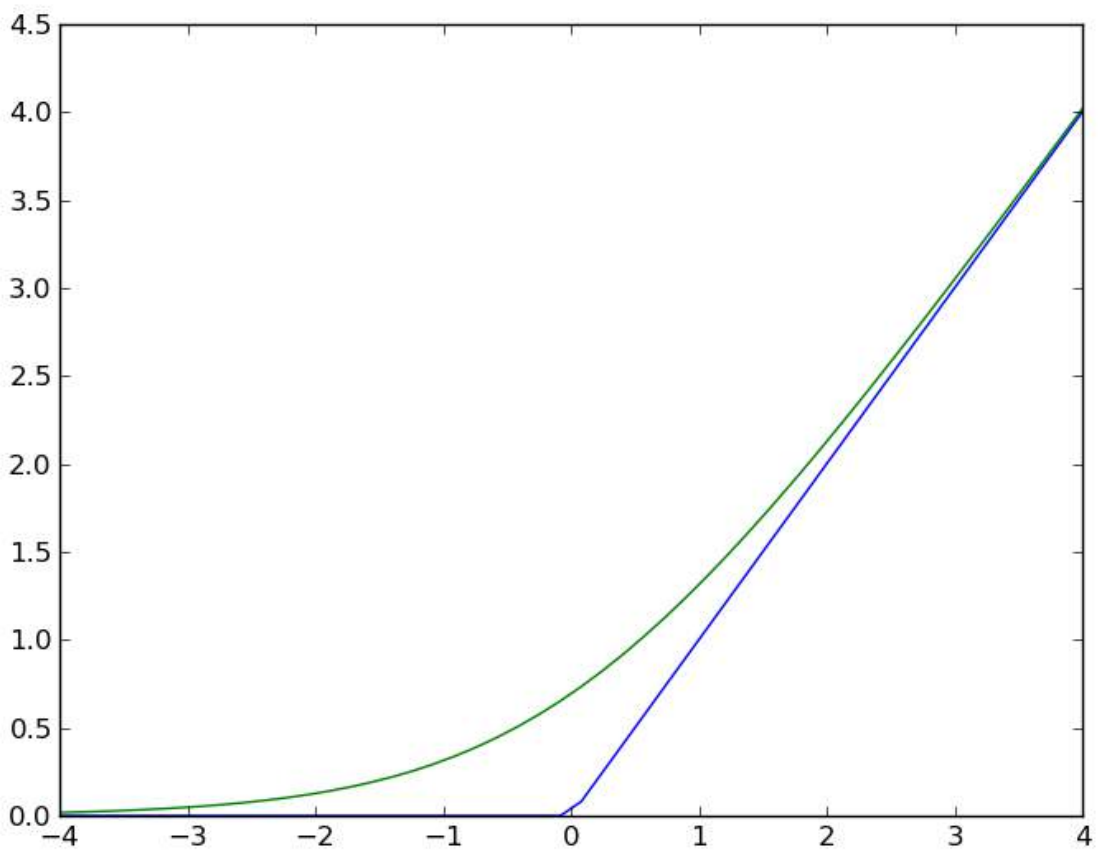
Image credit: codelabs google

Avoid Overfitting

- Architecture of the network as prior:
  - Convolutions
  - Non-linear activation, e.g., ReLU
- Use data augmentation in the training
  - Affine transformations
- Dropout
- Batch Normalization

Rectified Linear Unit

ReLU (blue line)



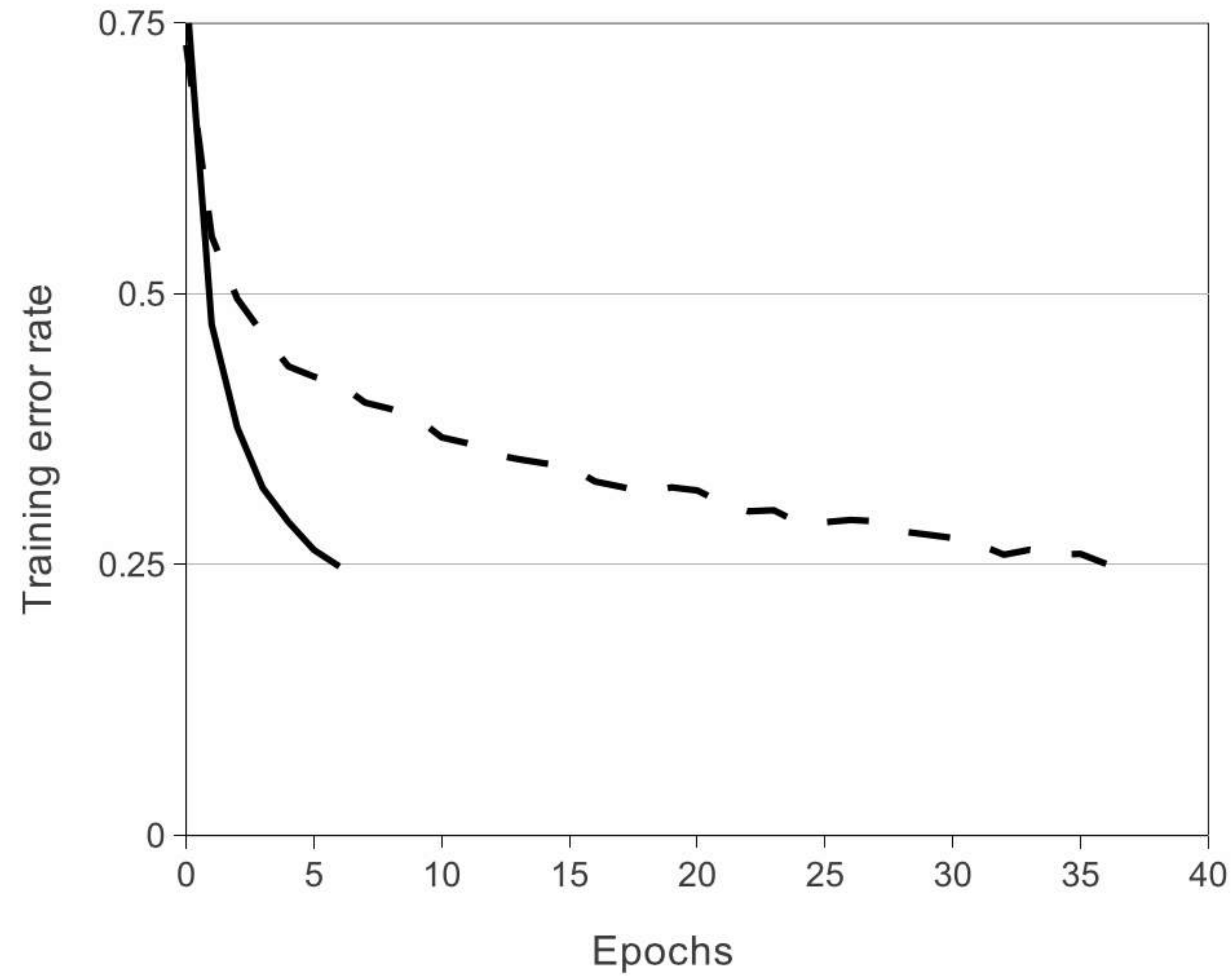


Figure 1: A four-layer convolutional neural network with ReLUs **(solid line)** reaches a 25% training error rate on CIFAR-10 six times faster than an equivalent network with tanh neurons **(dashed line)**. The learning rates for each net-

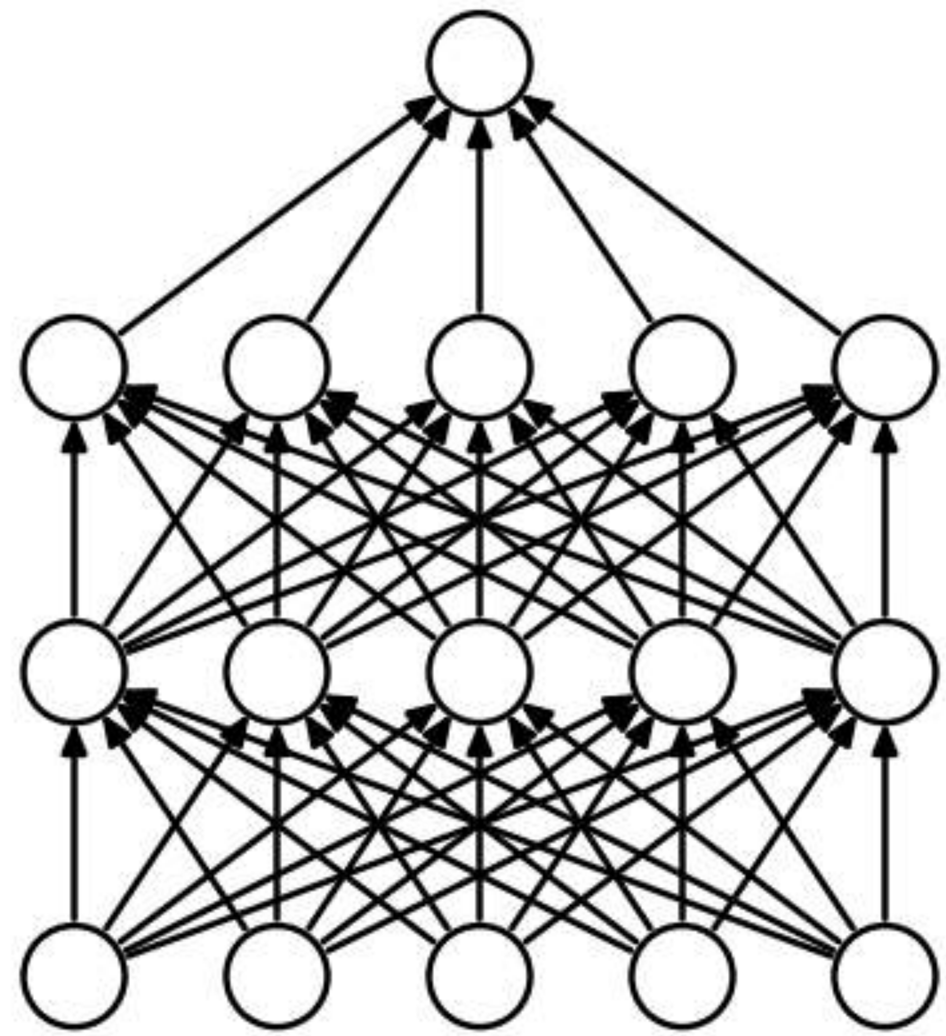
Krizhevsky et al. 12

Avoid Overfitting

- Dropout

training phase:
remove stochastically hidden units

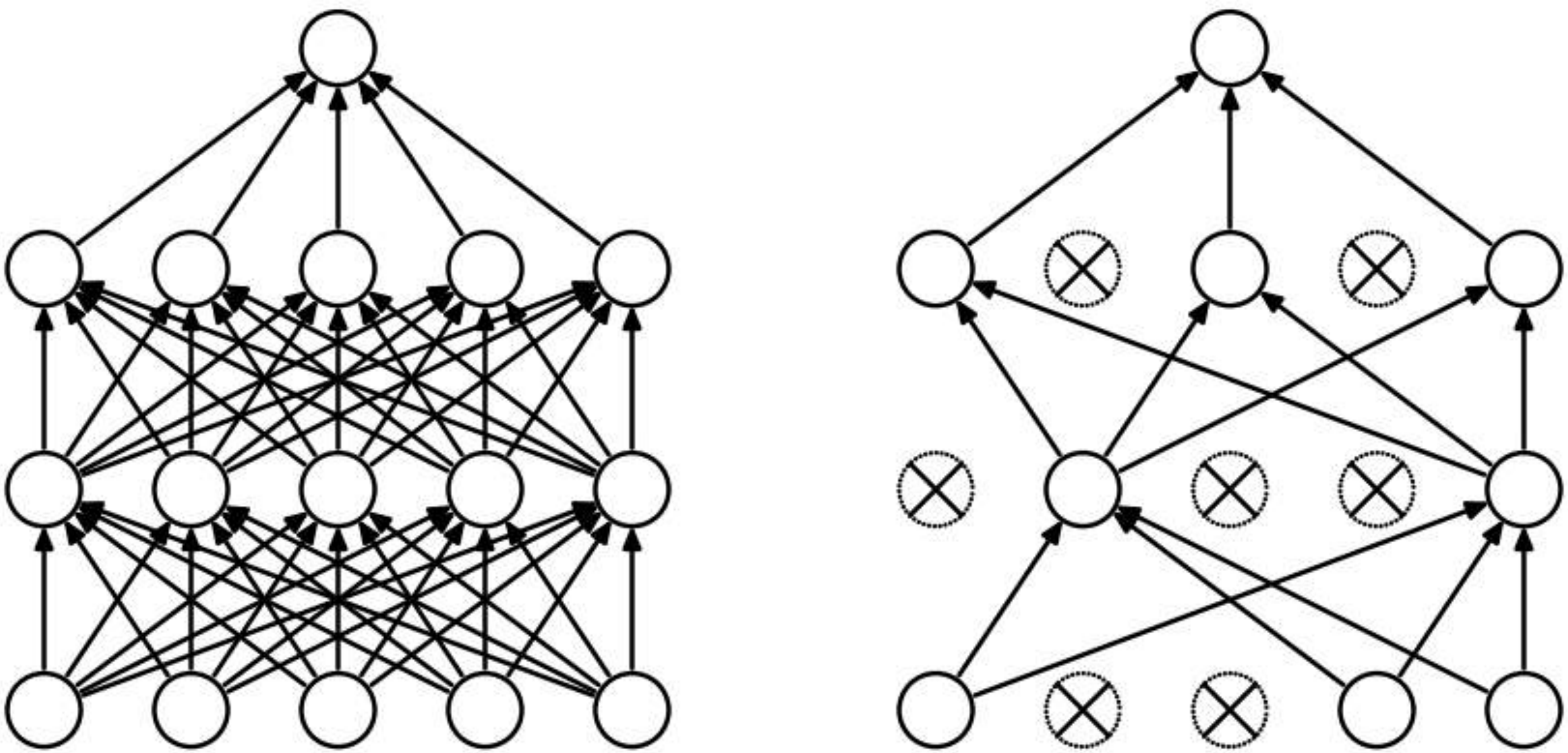
- Hidden units set to 0 with a probability (0.5, changes stochastically)
- Hidden units can not co-adapt to other hidden units



(a) Standard Neural Net

Avoid Overfitting

- Dropout



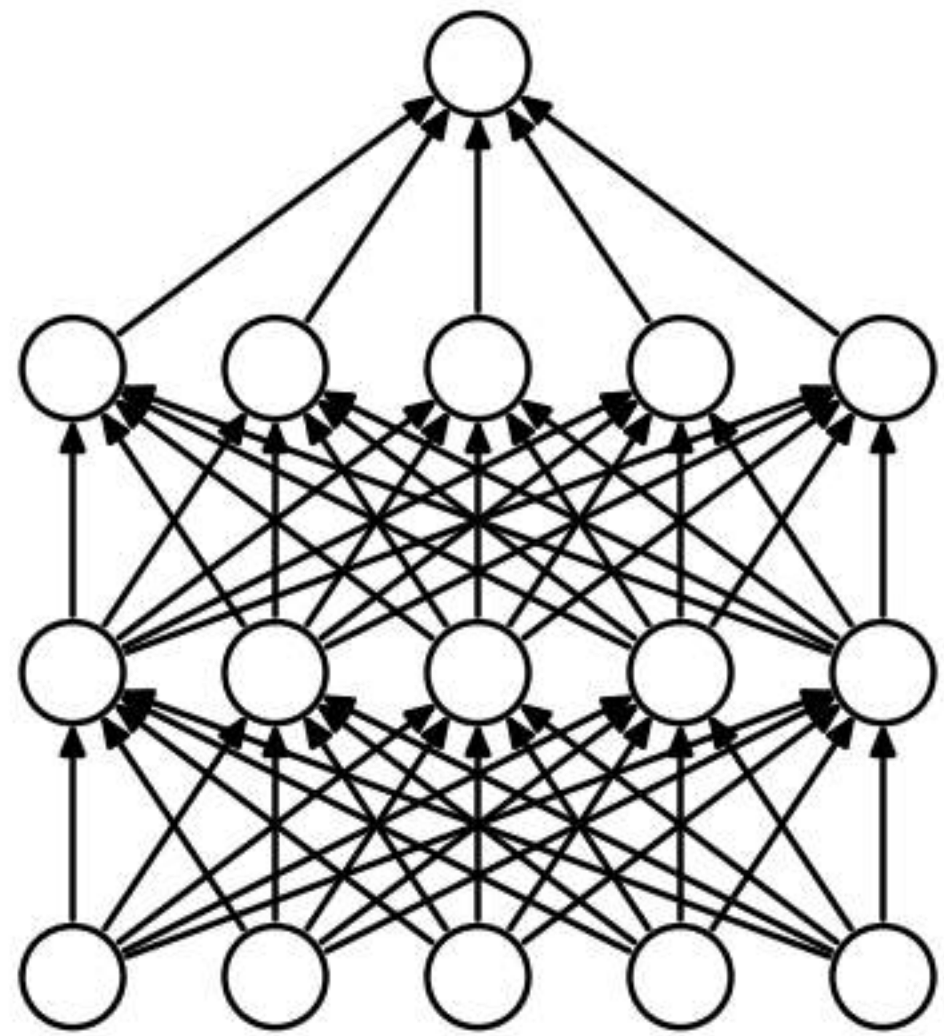
(a) Standard Neural Net (b) After applying dropout.

Avoid Overfitting

- Dropout

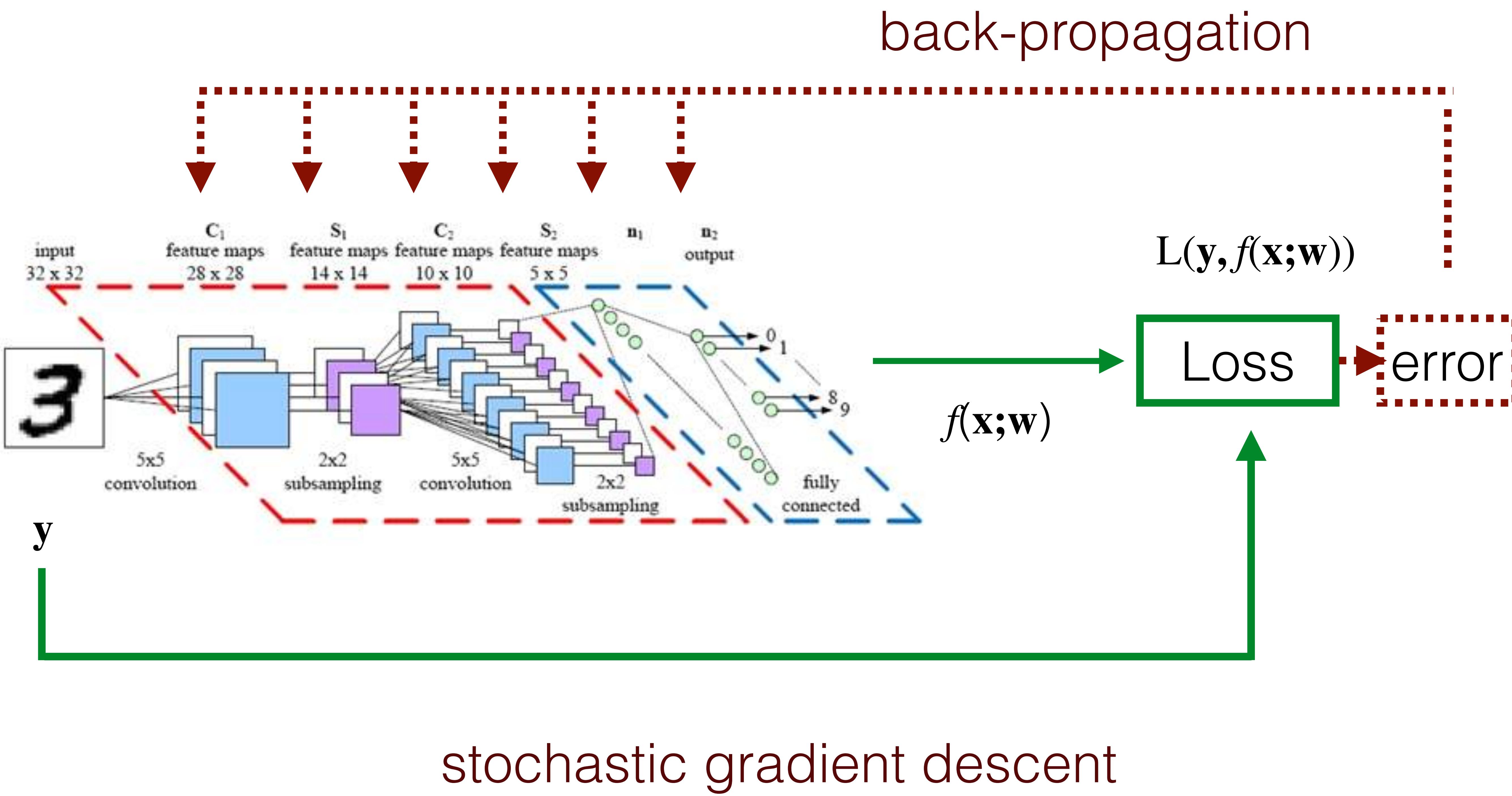
testing phase:
all hidden units used

- Multiply hidden layers by the dropout probability (0.5, not stochastic)
- Better generalization



(a) Standard Neural Net

Learning


back-propagation
input
32 x 32
C1
feature maps
28 x 28
S1
feature maps
14 x 14
C2
feature maps
10 x 10
S2
feature maps
5 x 5
n1
n2
output
5x5
convolution
2x2
subsampling
5x5
convolution
2x2
subsampling
fully
connected
0
1
8
9
y
f(x;w)
L(y, f(x;w))
Loss
error
stochastic gradient descent

Back-propagation

Learning based on iterating between:

1. Propagation

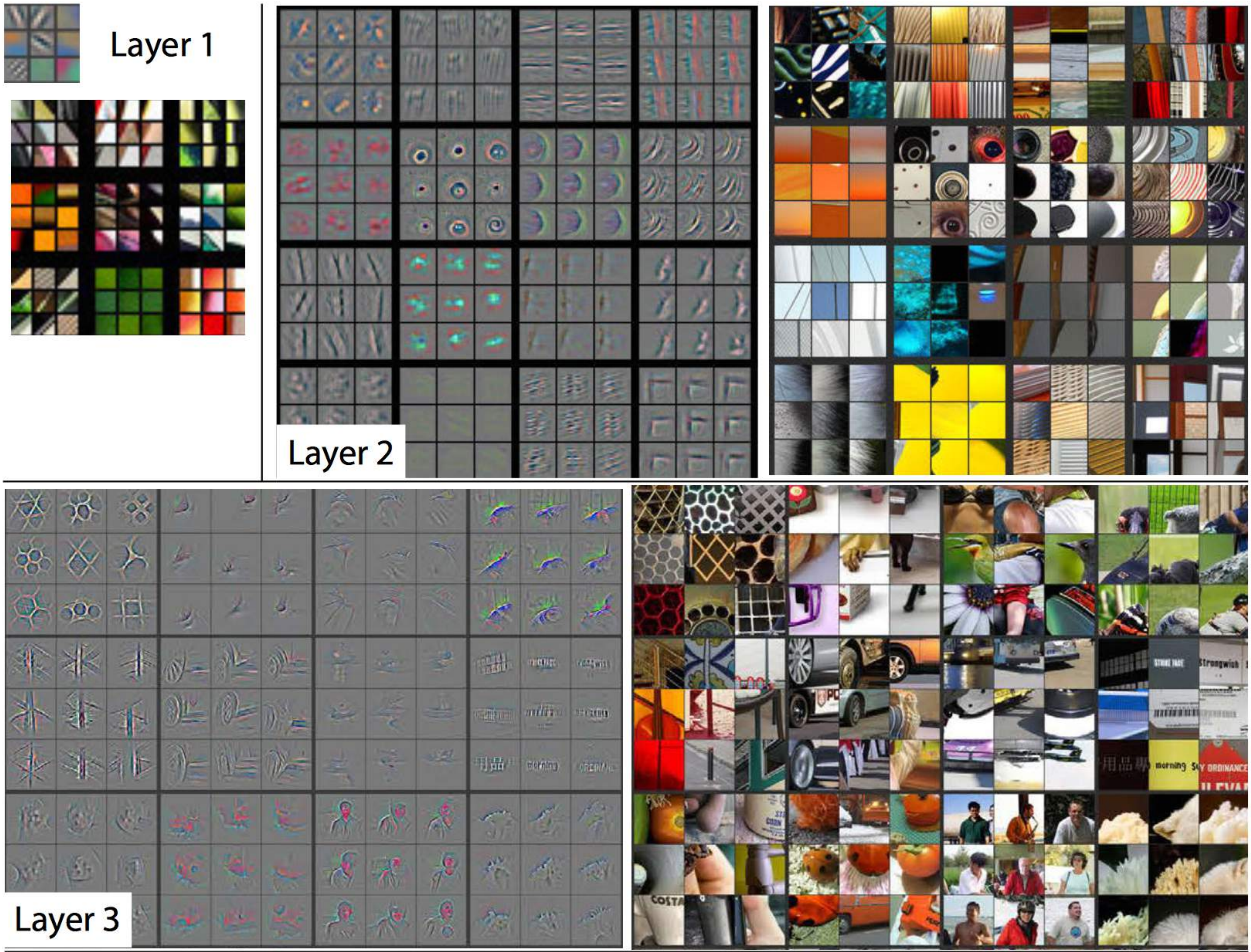
   1.1. Forward pass through NN

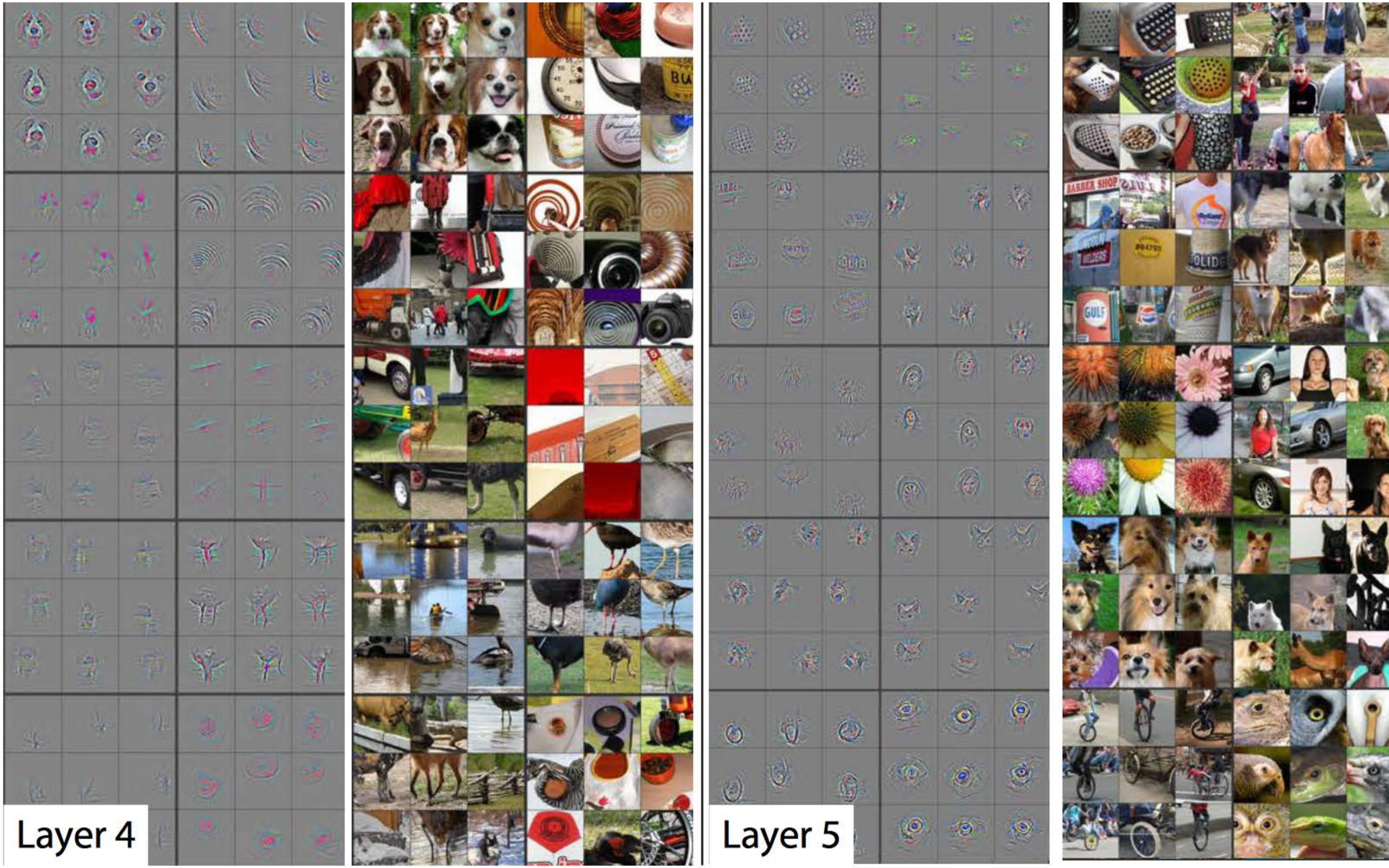
   1.2 Backward pass using partial derivatives

2. Weights updates

(stochastic gradient descend — with mini-batches)

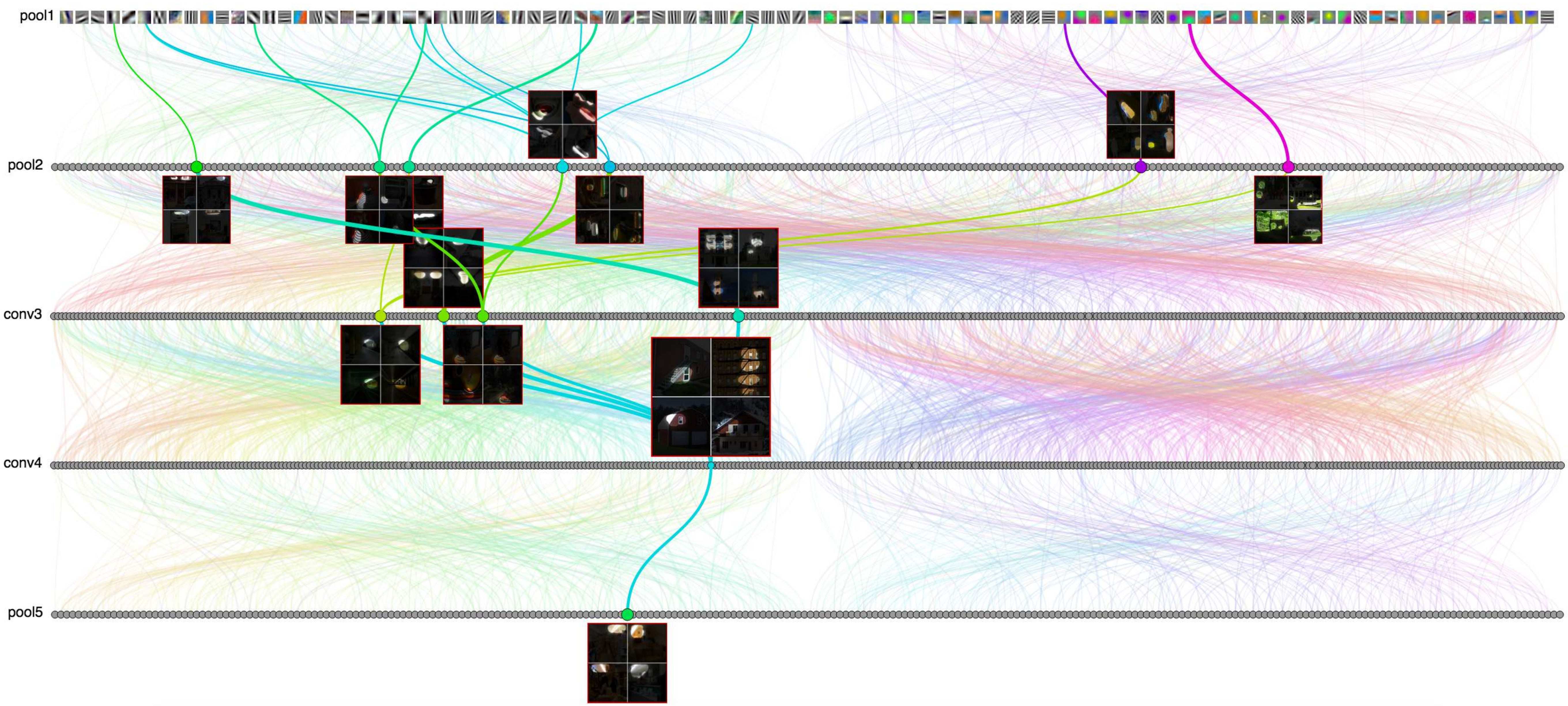
Visualization
of learned filters


Layer 1
Layer 2
Layer 3



Zeiler and Fergus 13

Visualization of learned filters



http://people.csail.mit.edu/torralba/research/drawCNN/drawNet.html

Invariance Properties

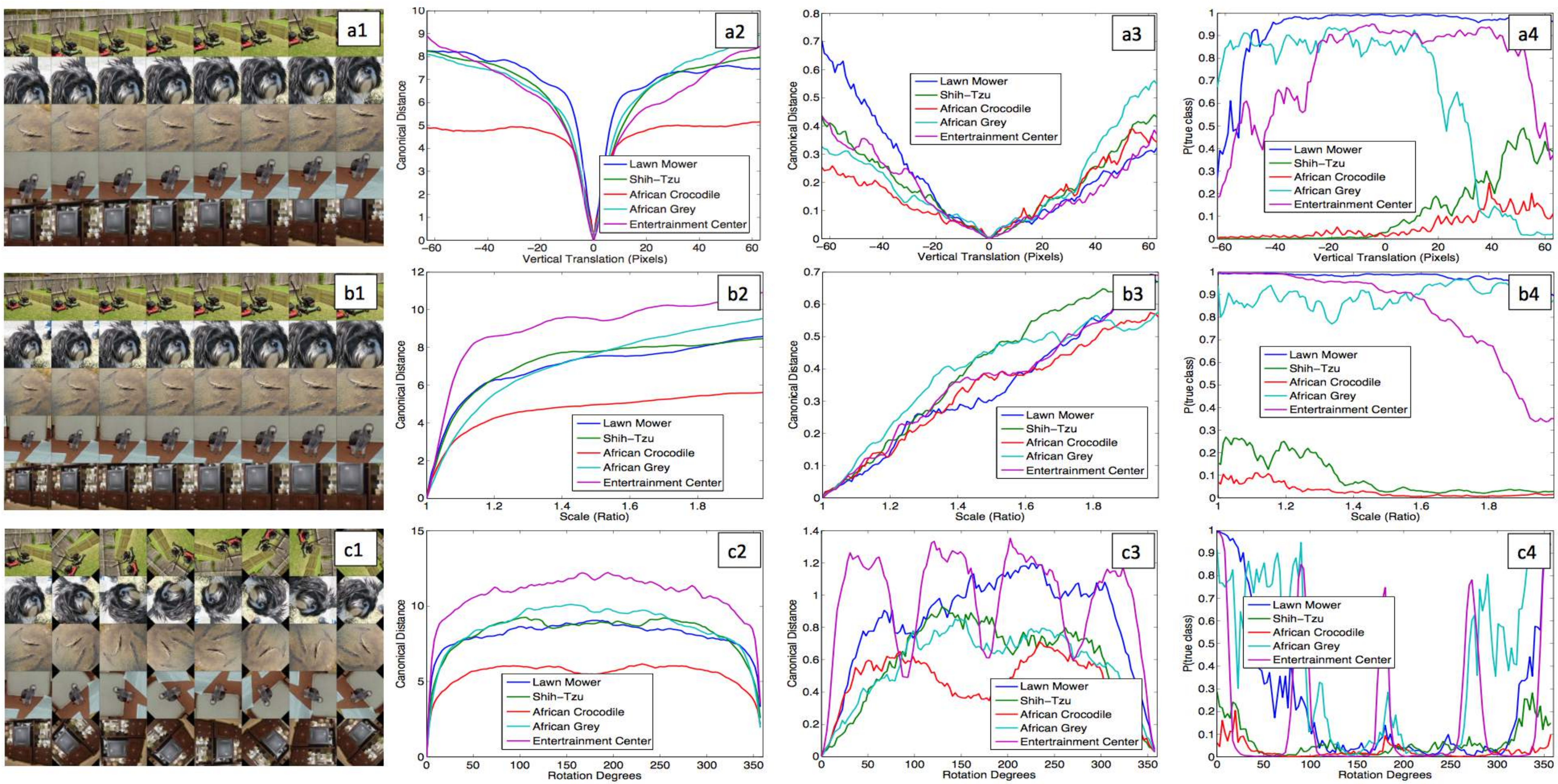


Figure 5. Analysis of vertical translation, scale, and rotation invariance within the model (rows a-c respectively). Col 1: 5 example images undergoing the transformations. Col 2 & 3: Euclidean distance between feature vectors from the original and transformed images in layers 1 and 7 respectively. Col 4: the probability of the true label for each image, as the image is transformed.

Zeiler and Fergus 13

Overview

- Introduction
- Artificial Neural Networks
- Computational Models of Object Recognition
- Artificial Neural Networks for Object Recognition
- **Applications**

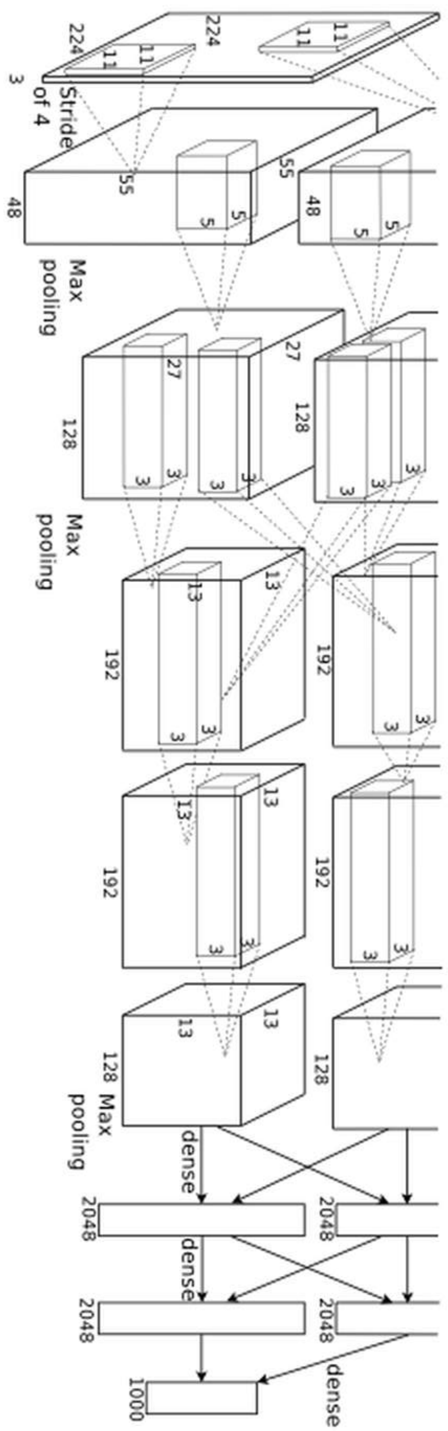
Applications

- Use a pre-trained CNN as a feature extractor
- Fine-tune on limited data
- Train from scratch on big data

Applications

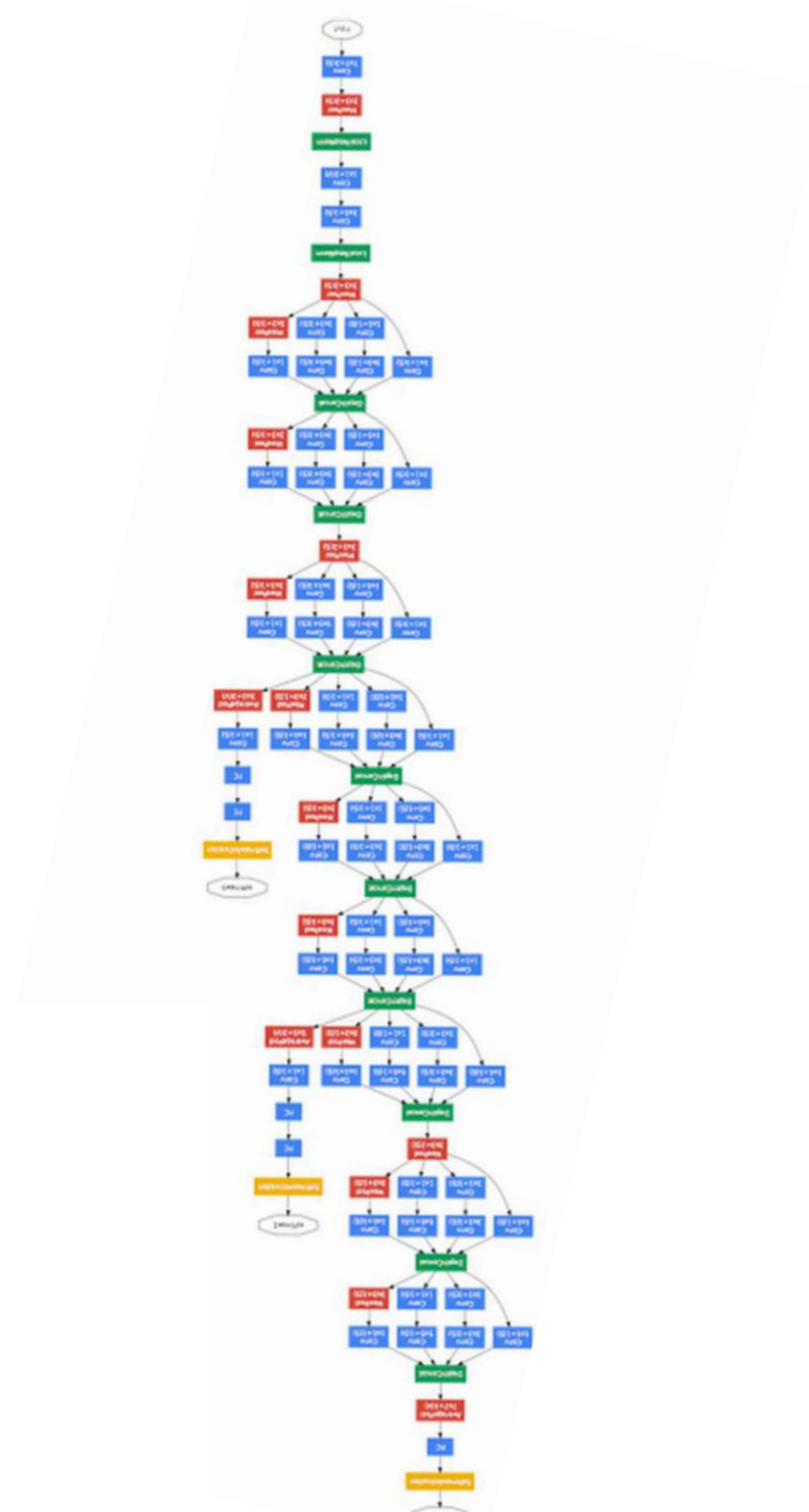
- **Use a pre-trained CNN as a feature extractor**
- Fine-tune on limited data
- Train from scratch on big data

Object classification

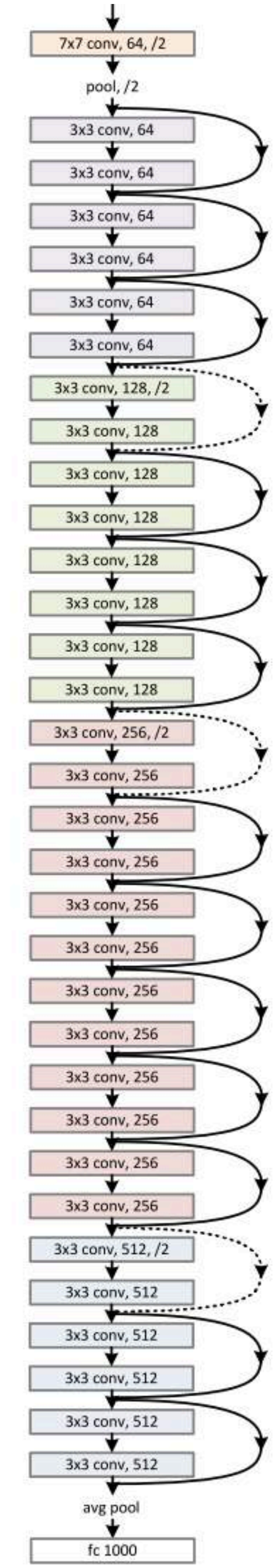


AlexNet 12

VGG 14



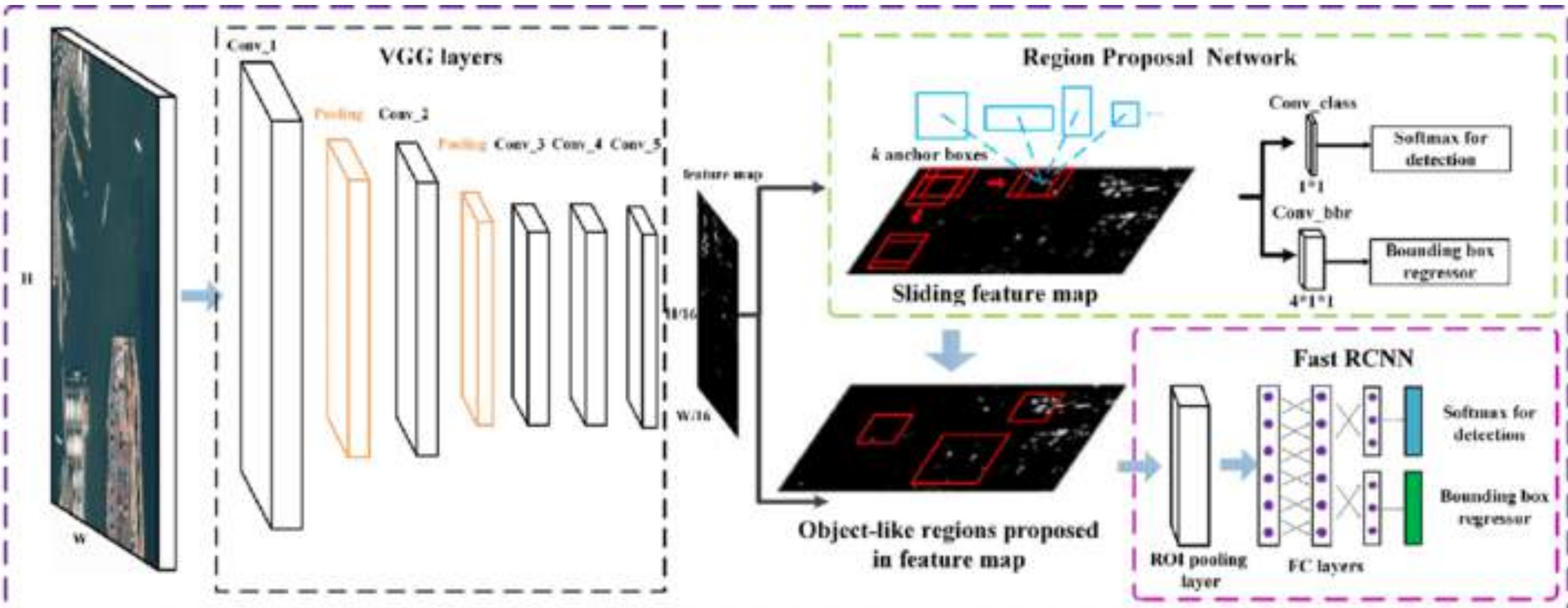
GoogLeNet 14



ResNet 15

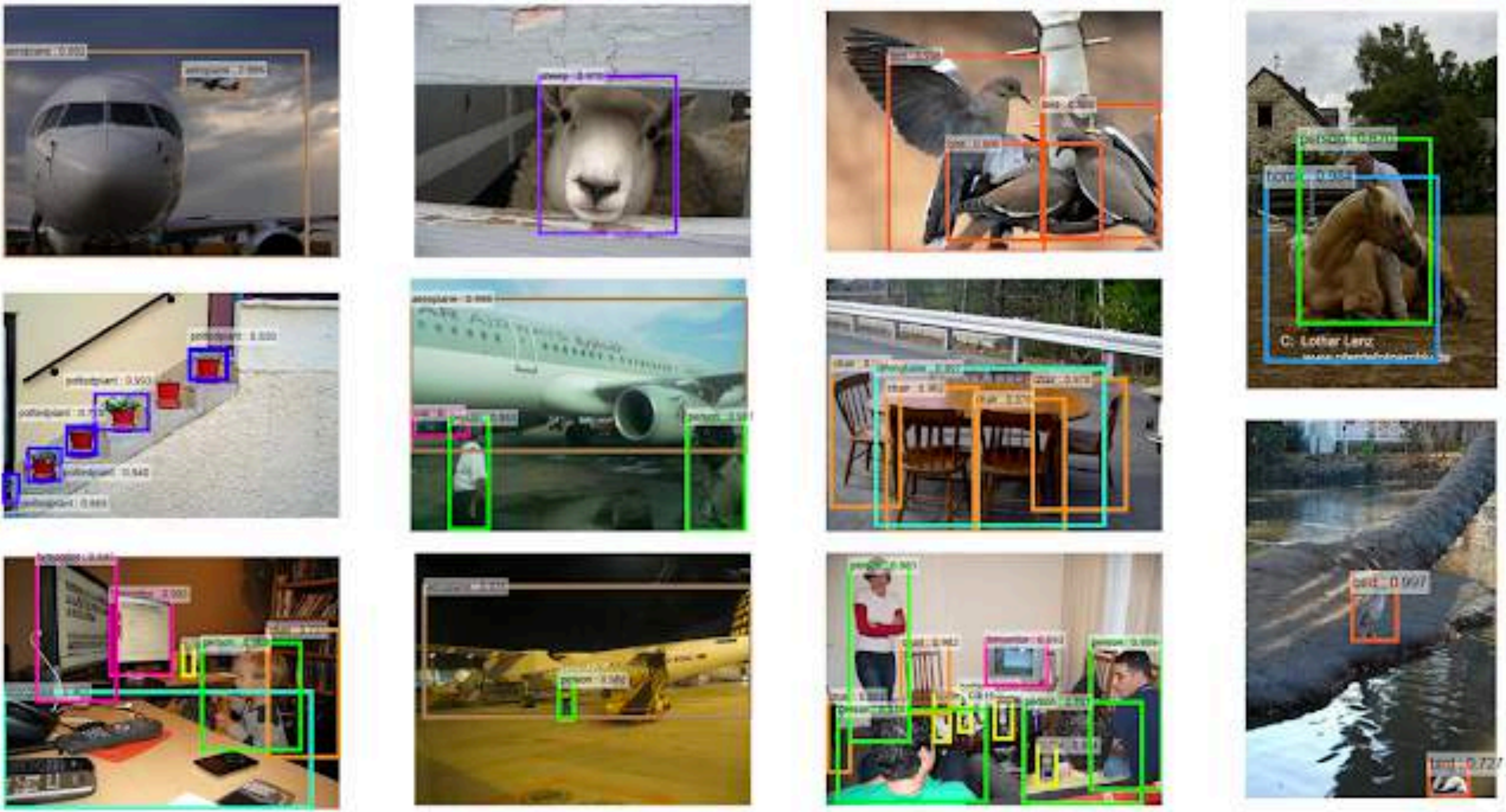
Object Detection

Faster Region CNN



Ren et al. 16

Object Detection

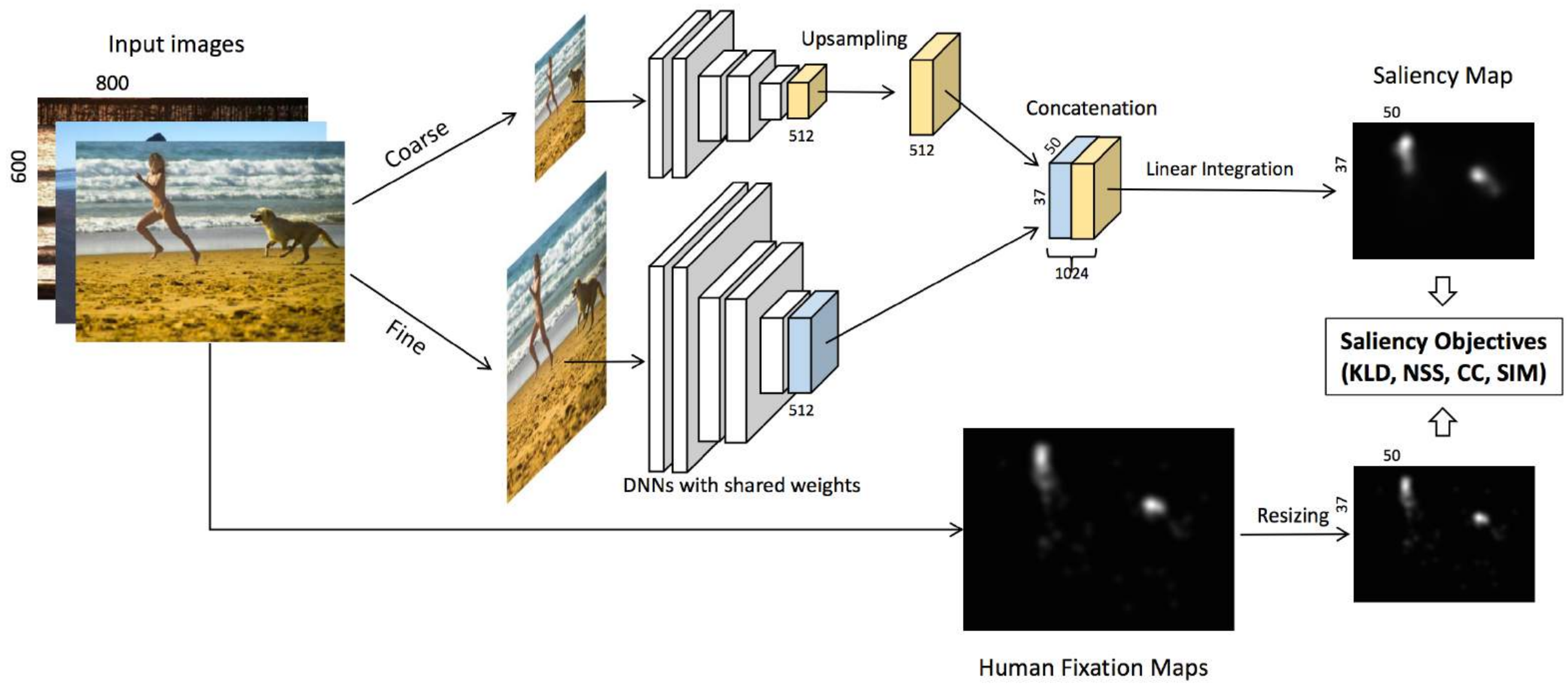


Applications

- Use a pre-trained CNN as a feature extractor
- **Fine-tune on limited data**
- Train from scratch on big data

Saliency Prediction

Reducing the Semantic Gap in Saliency Prediction by Adapting Neural Networks

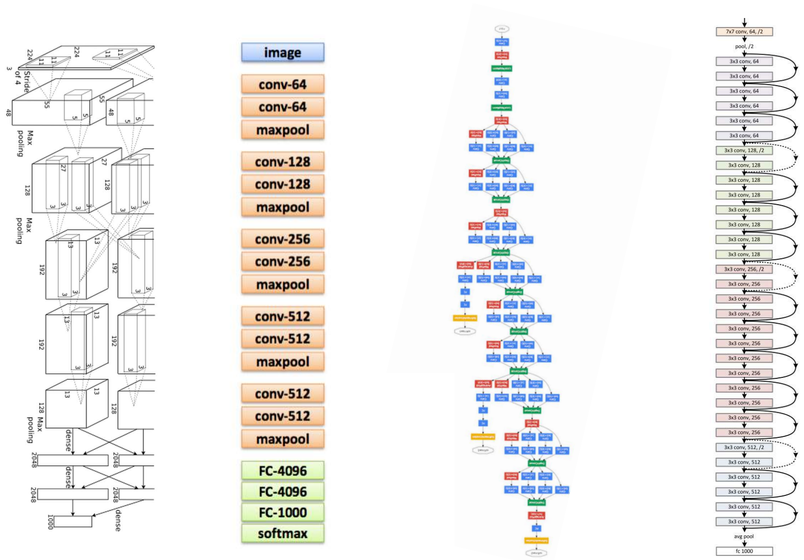


Huang et al. 15

Applications

- Use a pre-trained CNN as a feature extractor
- Fine-tune on limited data
- **Train from scratch on big data**

Places Recognition

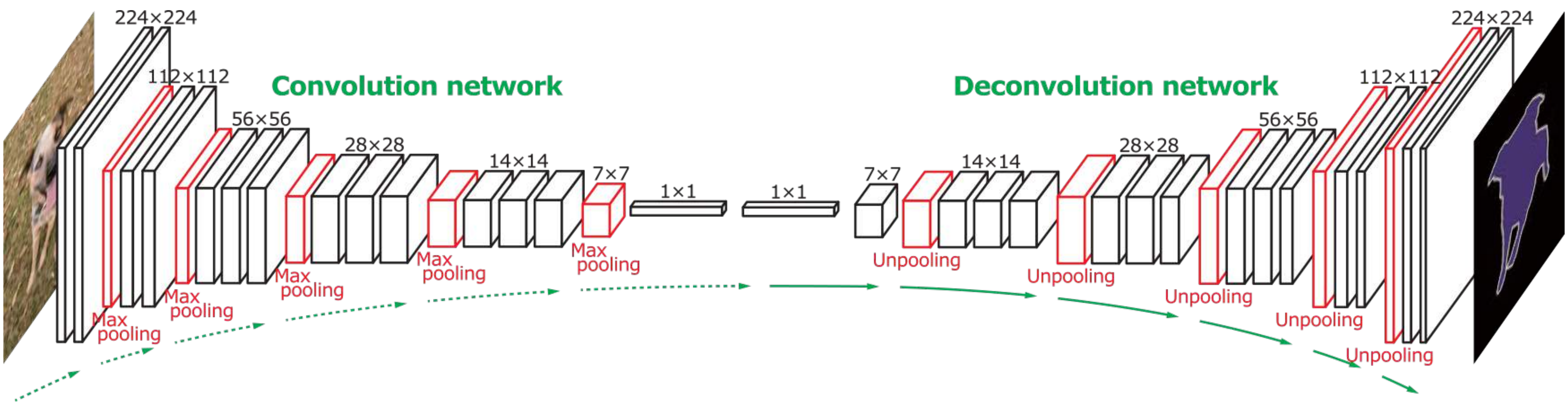


10 million images with
400+ unique scene categories

places2.csail.mit.edu

Semantic Segmentation

Learning Deconvolution Network for
Semantic Segmentation



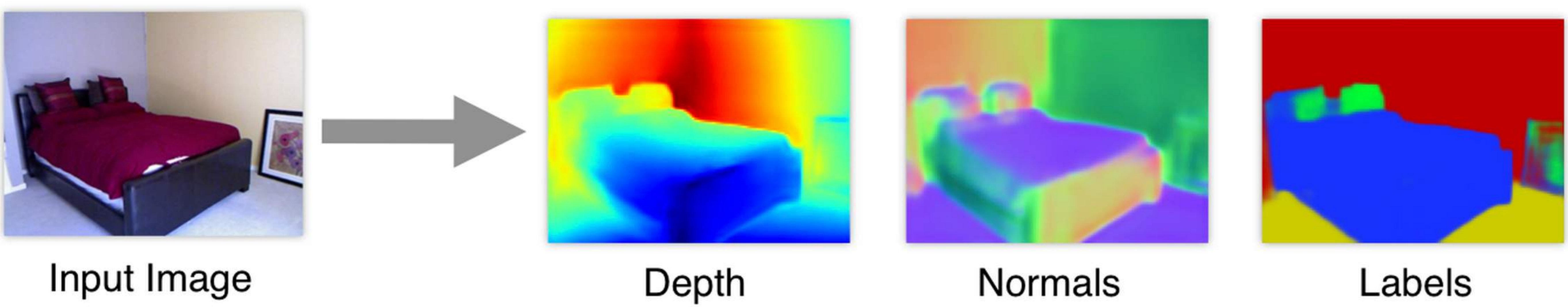
Noh et al. 15

Semantic Segmentation

Noh et al. 15

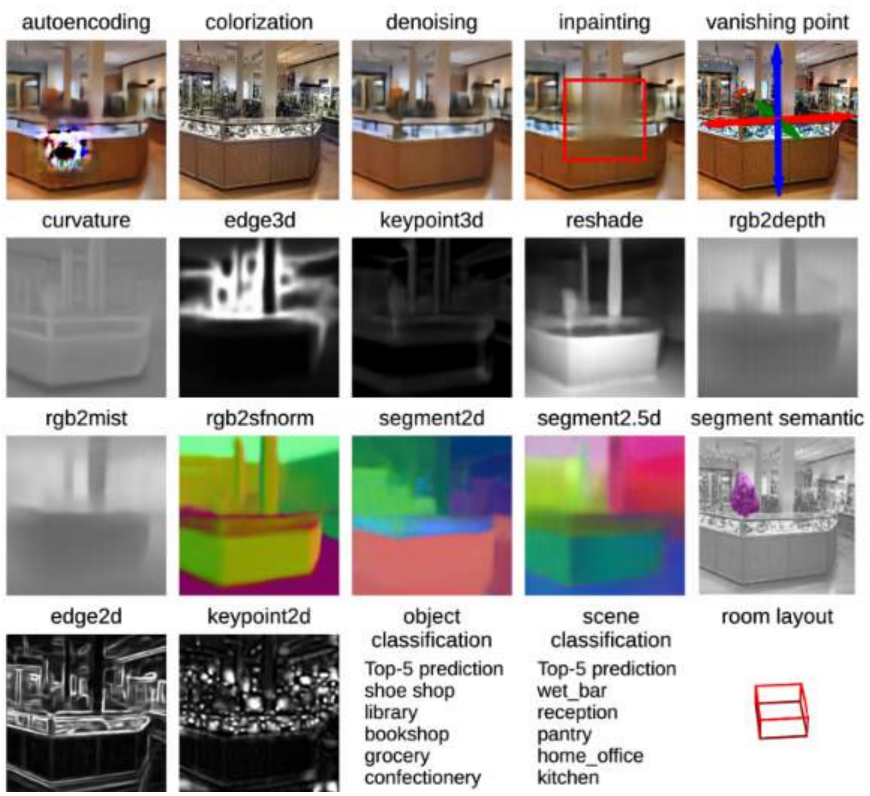
Depth Map Prediction

Depth Map Prediction from a Single Image
using a Multi-Scale Deep Network

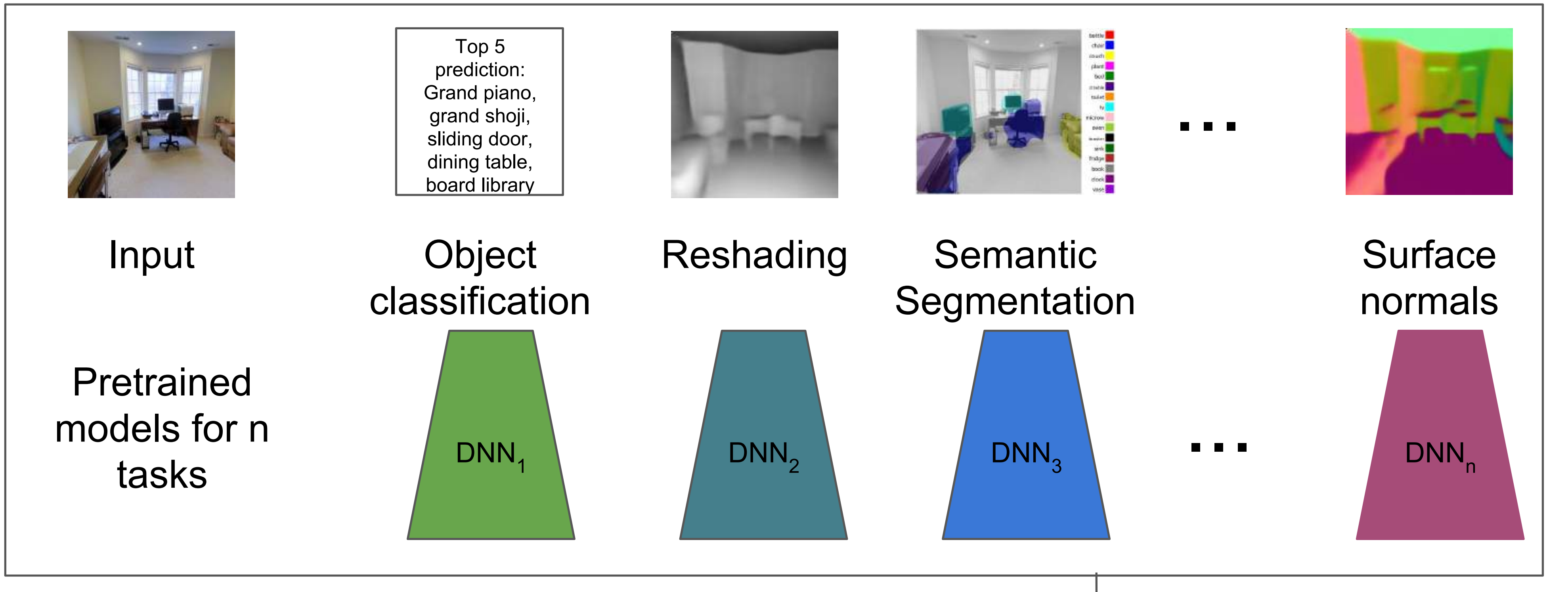


Eigen and Fergus 14

Multiple tasks predictions



taskonomy.stanford.edu

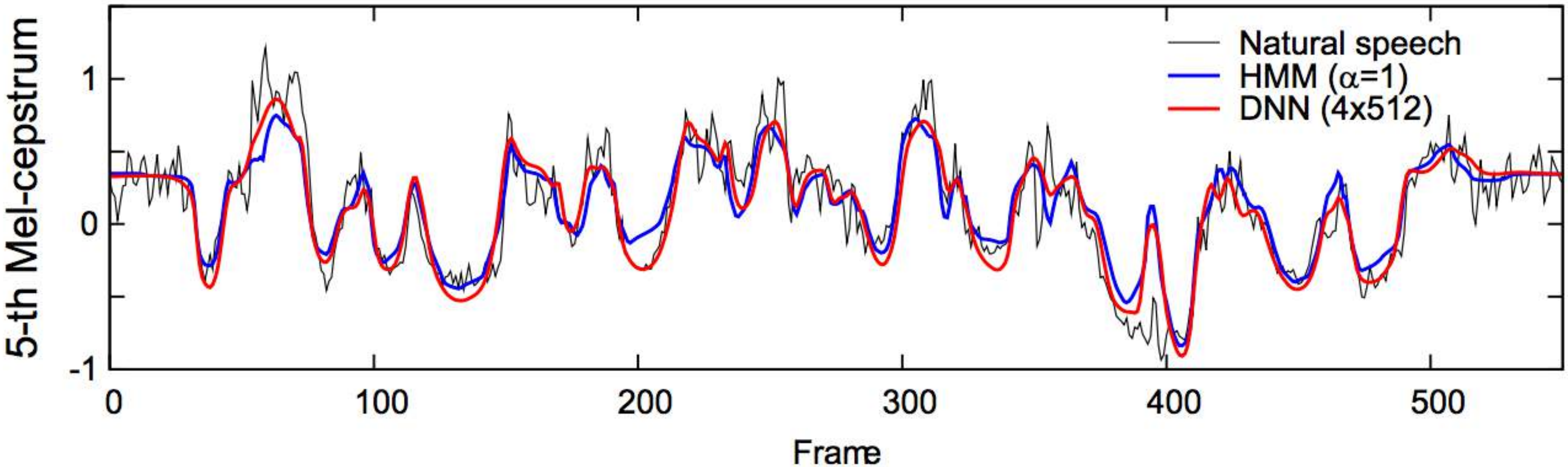


Zamir et al., CVPR 2018

Dwivedi & Roig, CVPR 19

Applications

not only for vision…

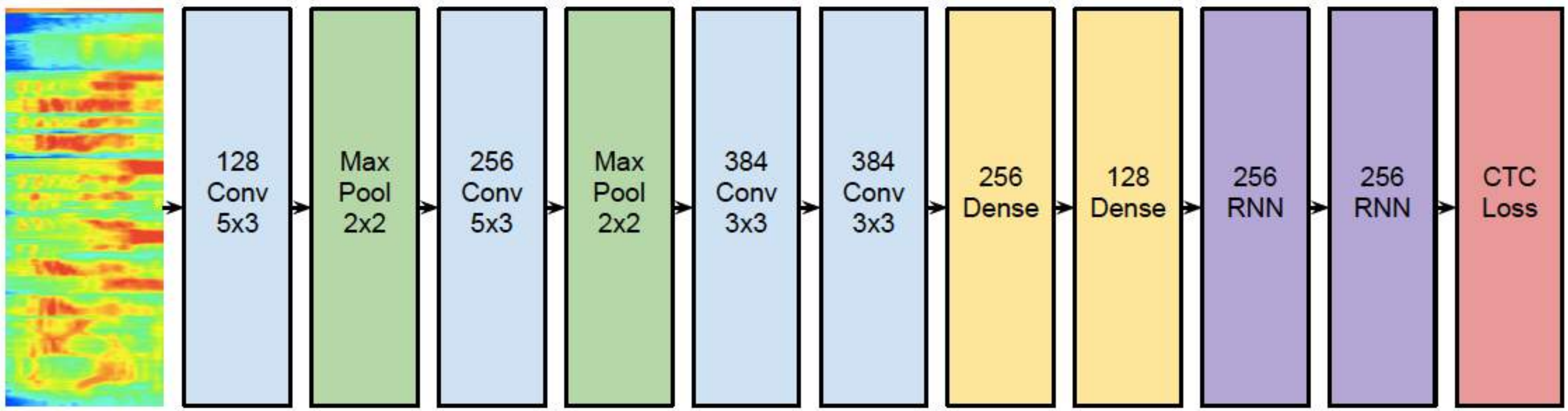


Statistical parametric speech synthesis
using deep neural networks

Zen et. al 13

Applications

End-to-End Deep Neural Network for Automatic Speech Recognition



phonemes recognition

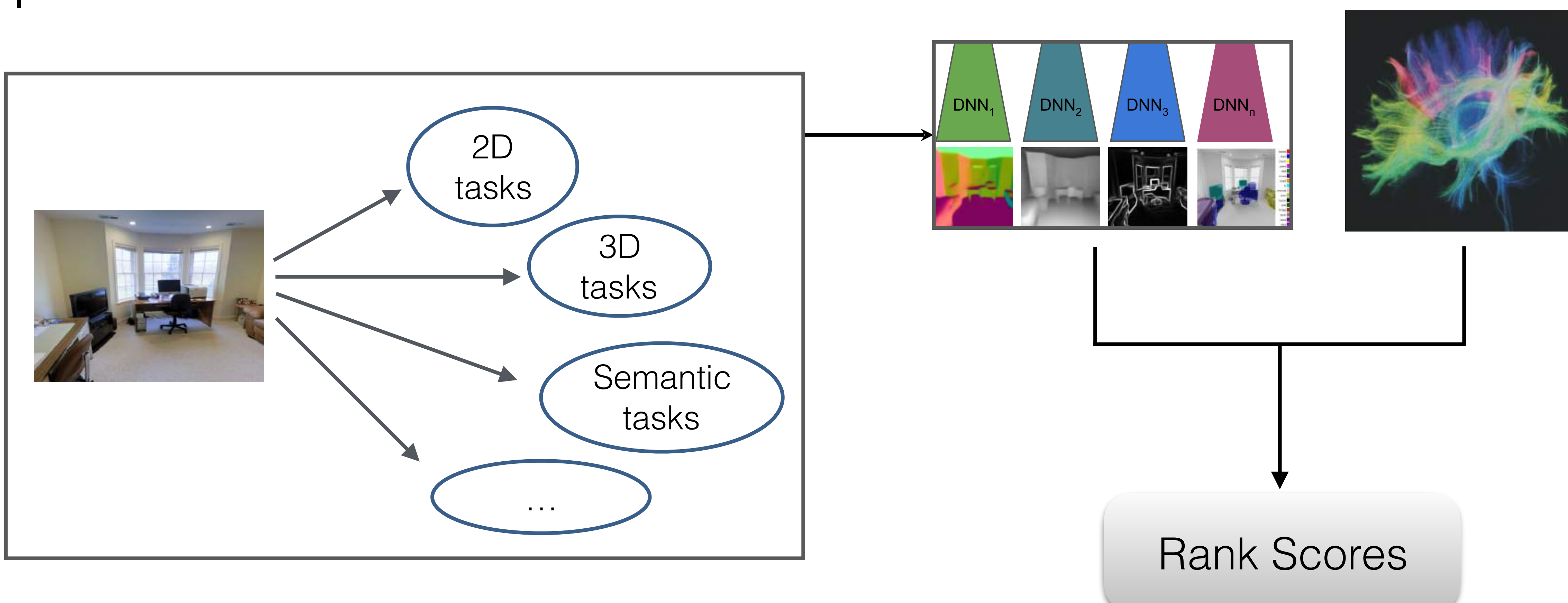
Song and Cai 15

Exploring vision tasks representation in the brain

* Can we assess functions of a brain area by comparing the correlation of its responses with a large set of diverse models trained on different computer vision tasks?

Kshitij Dwivedi

Mick Bonner



Navigational Affordance Cortical Responses Explained by Semantic Segmentation model, *ECCVW 2018.*
Explaining Scene-selective Visual Areas Using Task-specific Deep Neural Network Representations, submitted. *Ext. in prep.*

Applications - Frameworks

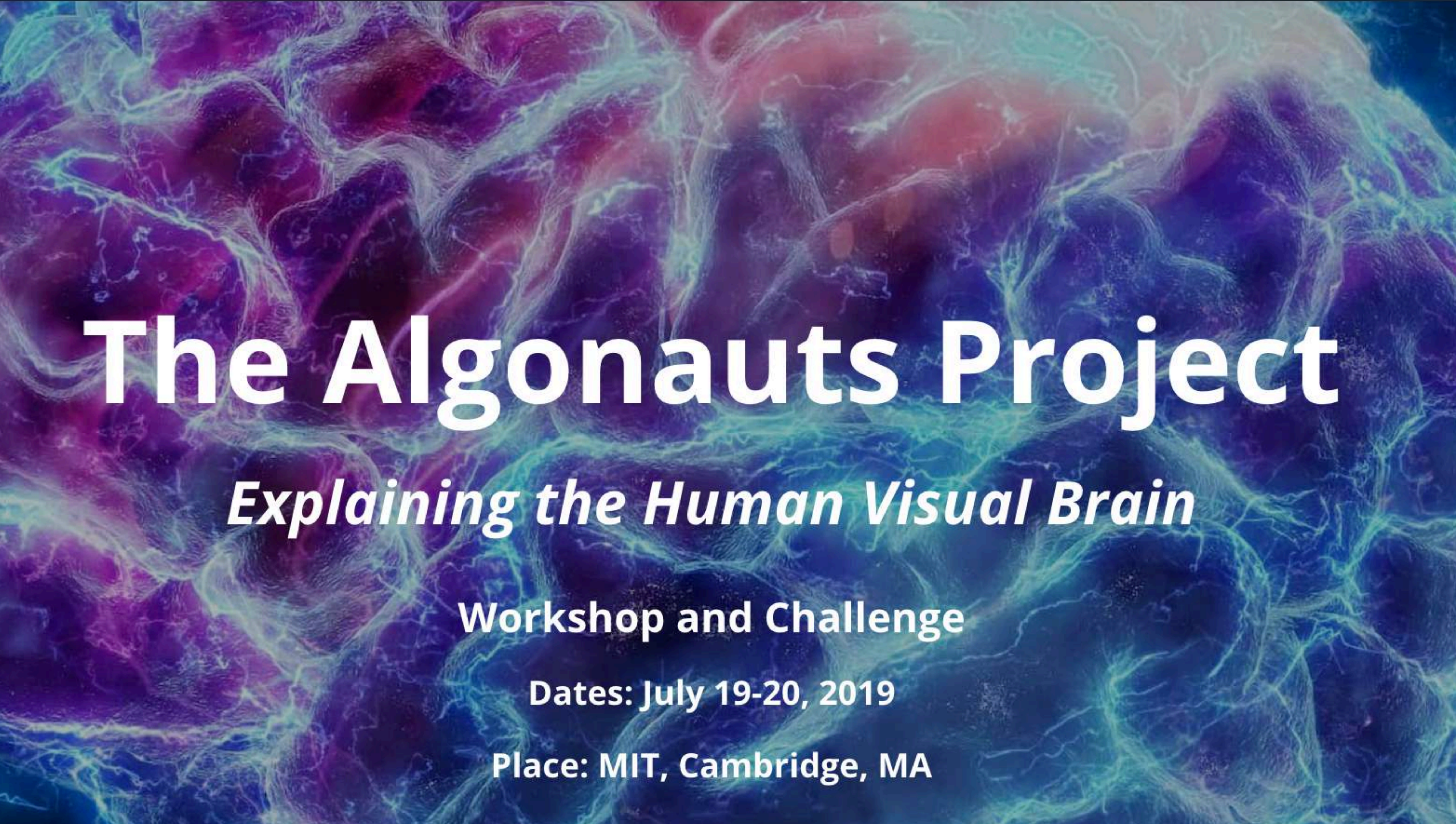
- pyTorch
  - Python
  - http://pytorch.org
- TensorFlow
  - Python, JavaScript
  - https://www.tensorflow.org
- Keras
  - Python, high level API on top of TensorFlow
  - https://keras.io
- Caffe
  - C++ with Matlab and Python interfaces
  - http://caffe.berkeleyvision.org


Q&A

The Algonauts Project

Explaining the Human Visual Brain

Time	Event
12:30 pm - 1:00 pm	Registration / Welcome
1:00 pm - 2:00 pm	**Introduction to Neural Networks** ***Gemma Roig***
2:00 pm - 2:15 pm	BREAK
2:15 pm - 3:15 pm	**Introduction to Brain Imaging: fMRI and MEG/EEG** ***Yalda Mohsenzadeh***
3:15 pm - 3:30 pm	BREAK
3:30 pm - 4:30 pm	**Comparing Brains and DNNs: Methods and Findings** ***Martin Hebart***
4:30 pm - 4:45 pm	BREAK
4:45 pm - 5:45 pm	**Comparing Brains and DNNs: Theory of Science** ***Radoslaw Cichy***
5:45 pm - 6:00pm	Summary

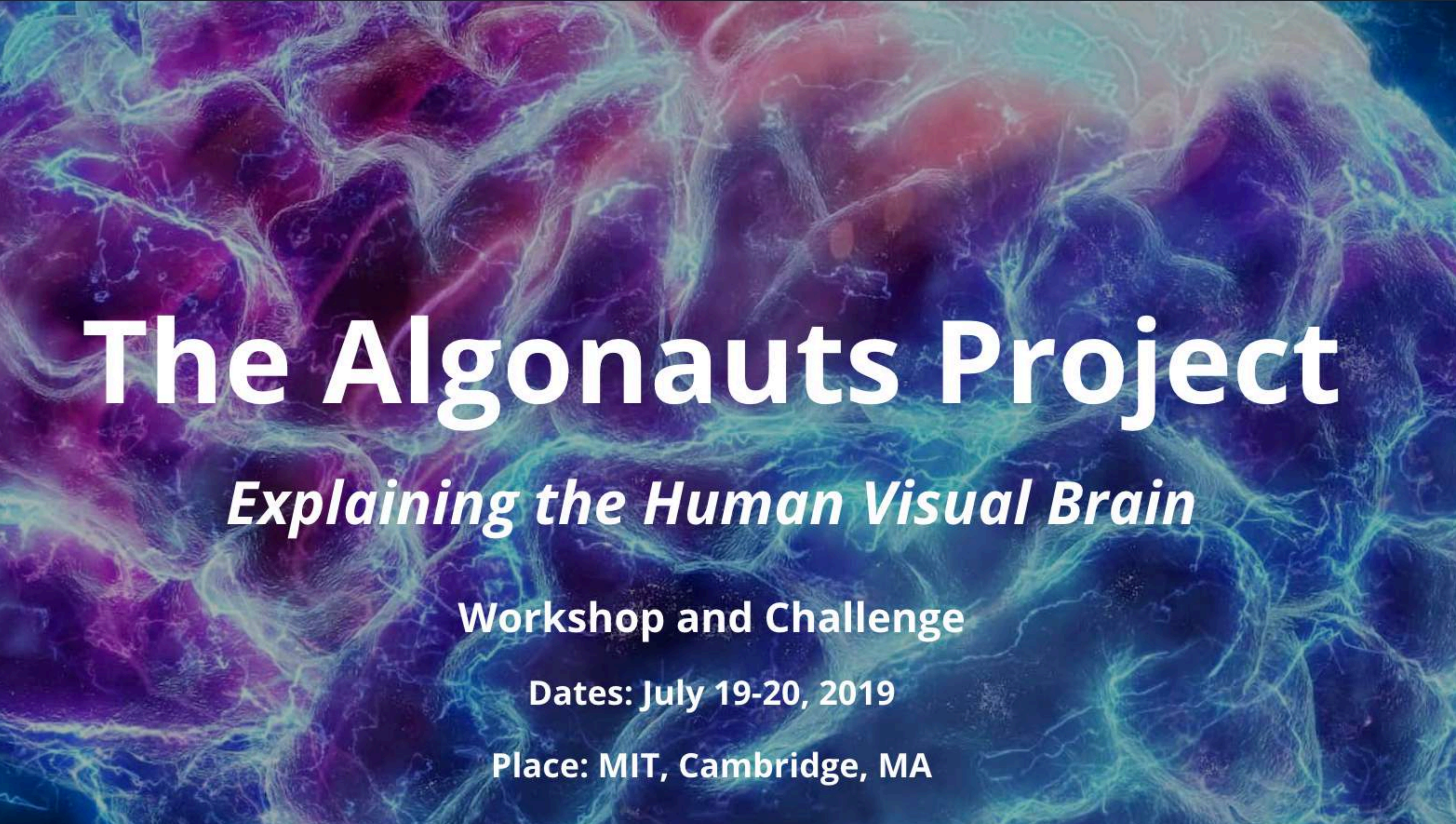


algonauts.csail.mit.edu

The Algonauts Project

Explaining the Human Visual Brain

20 July Schedule	Event
8:30 am – 9:00 am	Breakfast
9:00 am – 9:15 am	Introduction by Radoslaw Cichy
9:15 am – 9:35 am	Matt Botvinick
9:35 am – 9:55 am	Aude Oliva
9:55 am – 10:15 am	Thomas Naselaris
10:15 am – 11:00 am	Posters and Coffee
11:00 am – 11:20 am	David Cox
11:20 am – 11:40 am	James DiCarlo
11:40 am – 12:00 pm	Kendrick Kay
12:00 pm – 1:30 pm	Lunch on Your Own
1:30 pm – 1:50 pm	Introduction to the Algonauts Challenge by Radoslaw
1:50 pm – 2:50 pm	Invited Talks: Challenge Winners
2:50 pm – 3:30 pm	Posters and Coffee
3:30 pm – 3:50 pm	Talia Konkle
3:50 pm – 4:10 pm	Nikolaus Kriegeskorte
4:10 pm – 4:30 pm	Jack Gallant
4:30 pm – 5:00 pm	Panel Discussion with Speakers
5:00 pm – 6:30 pm	Reception



algonauts.csail.mit.edu