FACULTAD DE MEDICINA
UNIVERSIDAD DE CHILE

MAURICIO CERDA + ASHISH MAHABAL

DEEP LEARNING

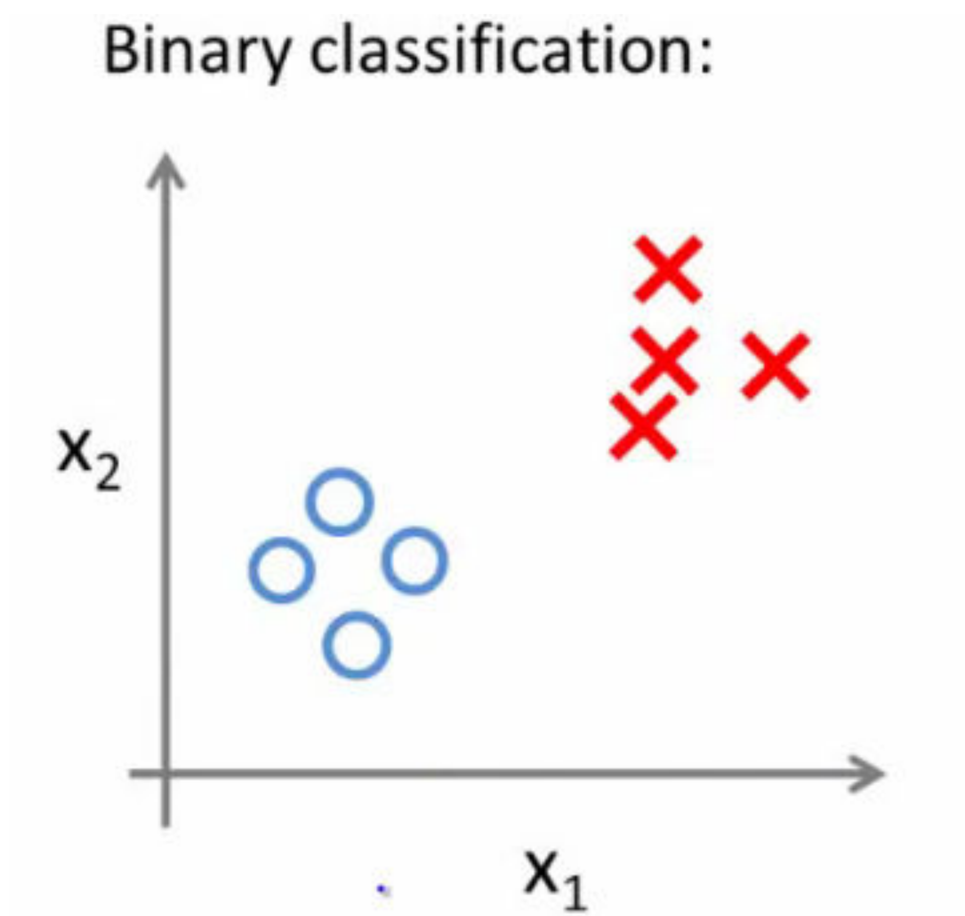
- La Serena, 8/22/2018 -

Outline

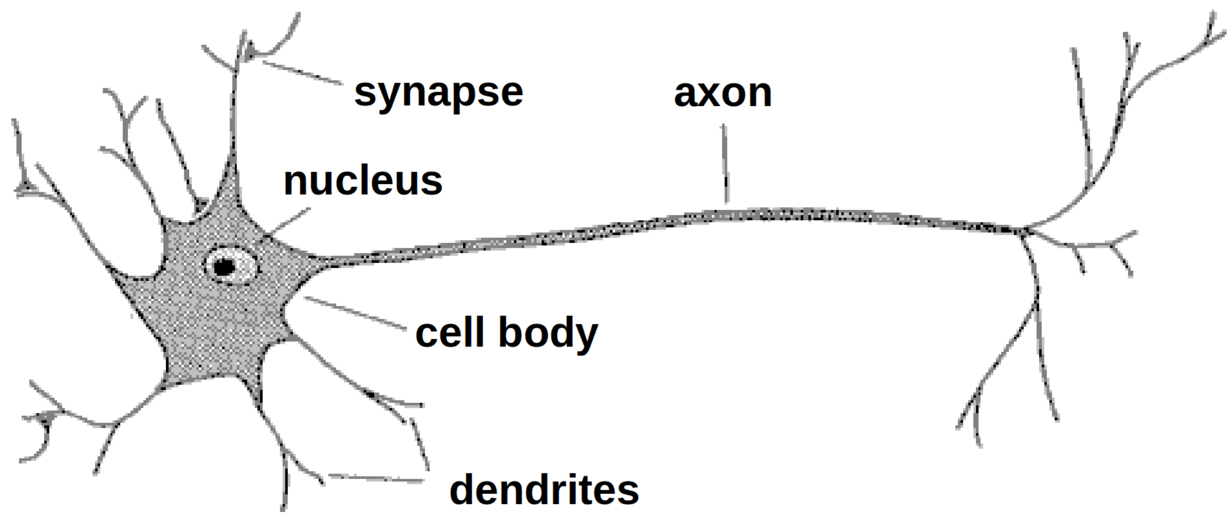
- Perceptron & Multilayer Perceptron
- Deep Learning
- Demo

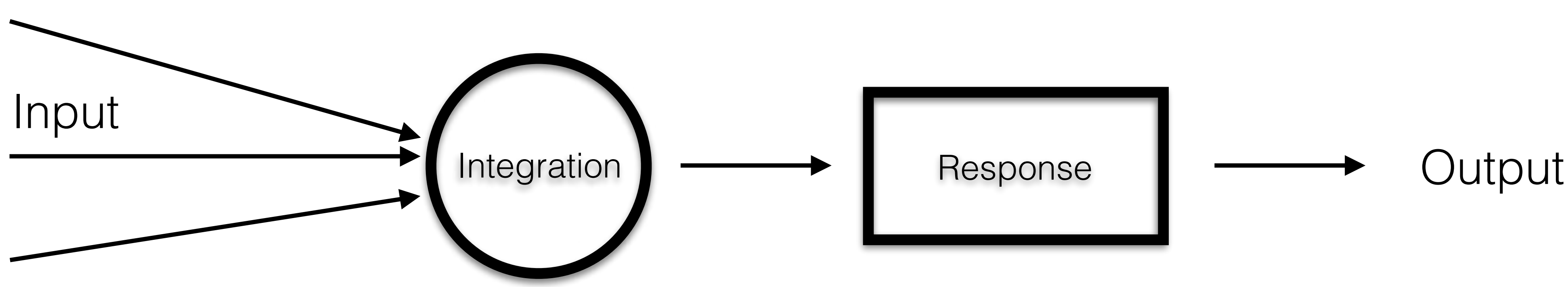
Supervised learning

- Objective: learn input/output association.



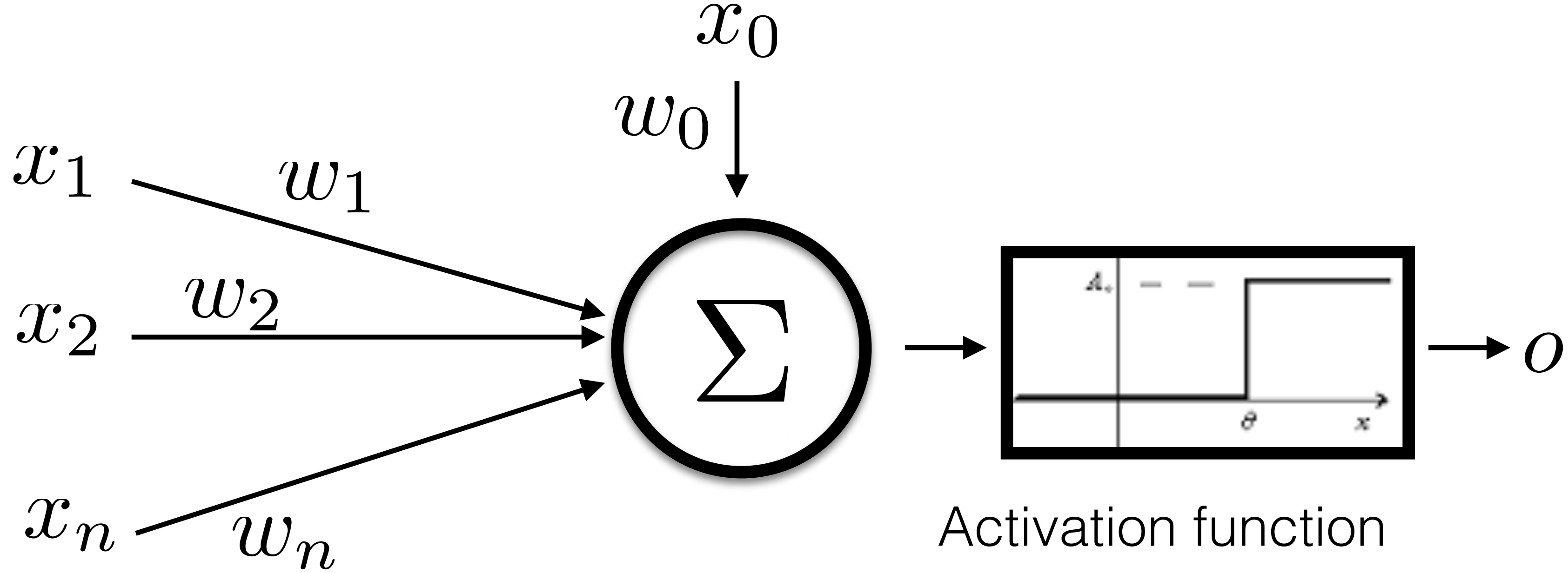
The human brain



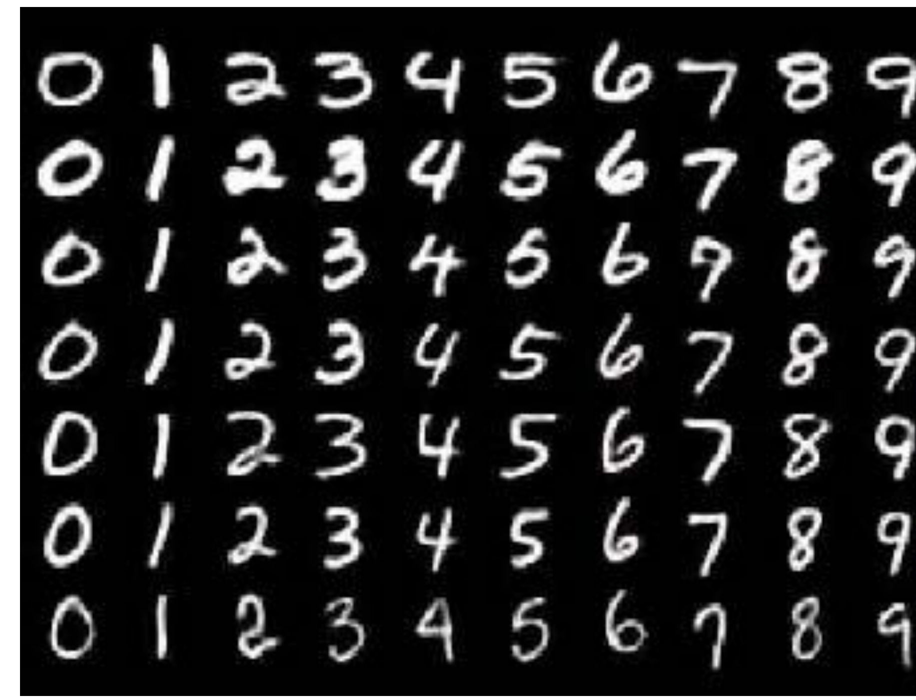


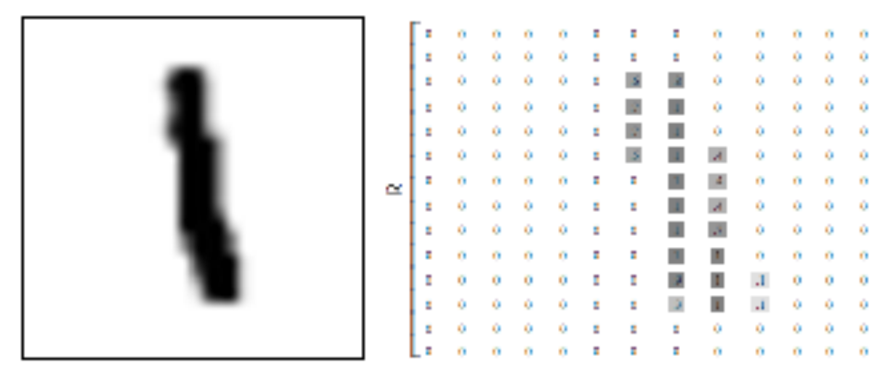
Perceptron

- A (functional) model of how neurons work.



Activation function





MNIST digits

0100000000

Real and artificial neurons

Learning rule

- How to learn with a perceptron?

$$w_i = w_i + \Delta w_i$$
$$\Delta w_i = \rho(t - o)x_i$$

where t is the objective, ρ is the learning rate, o is perceptron output.

- How does it work?

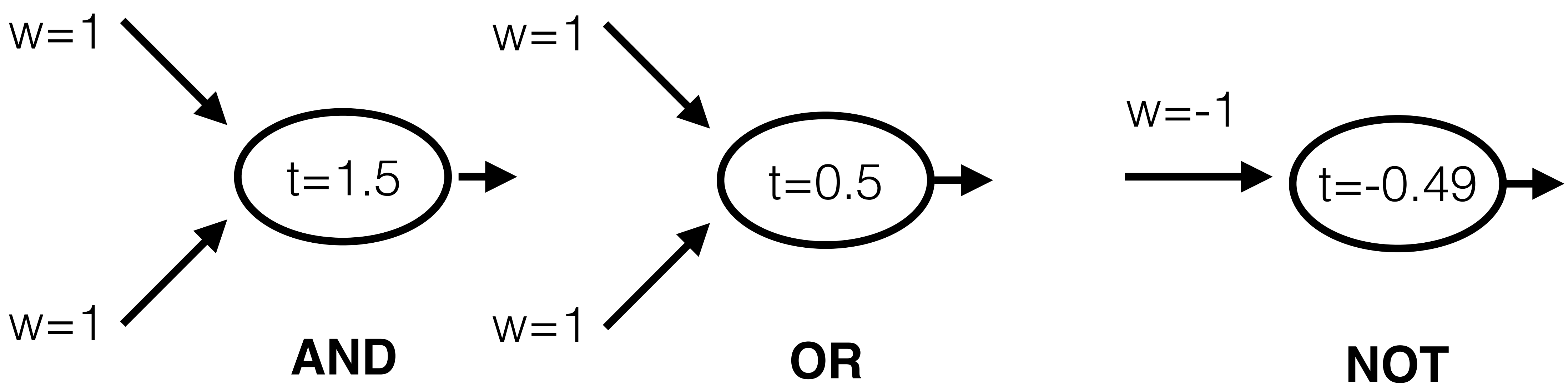
- If the output is correct (t= o), w does not change.

- If the output is incorrect (t != o), w will change to make the output as similar as possible to the objective.

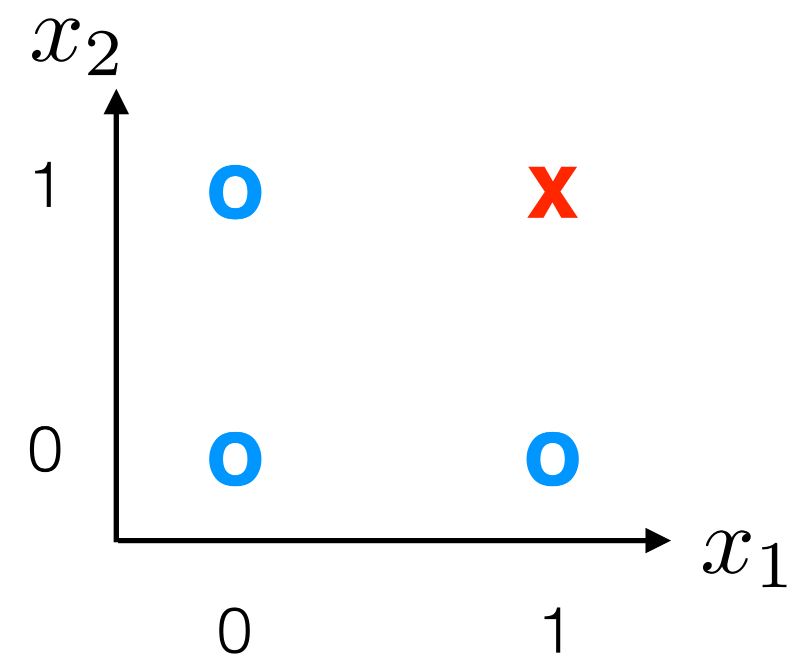
- The algorithm will converge if:
  - Data is linearly separable.
  - ρ is small enough

Example perceptron

- A few examples:



Perceptron



- Training the AND operation:

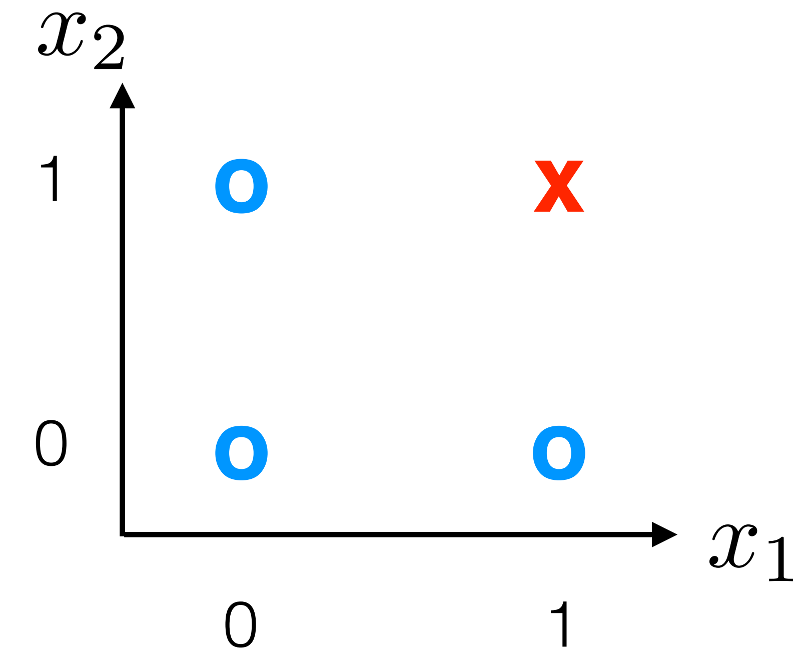
Iteration 1, f(x)=x>0.5, w=(0.1, 0.2, 0.3) $\rho = 0.1$

x_1	x_2	x_3	$w \cdot x$	o	t
-1	0	0		0	0
-1	0	1		0	0
-1	1	0		0	0
-1	1	1		**0**	**1**

Iteration 2, f(x)=x>0.5, w=(, ,) $\rho = 0.1$

x_1	x_2	x_3	$w \cdot x$	o	t
-1	0	0		0	0
-1	0	1		0	0
-1	1	0		0	0
-1	1	1		1	1

Perceptron



- Training the AND operation:

Iteration 1, f(x)=x>0.5, w=(0.1, 0.2, 0.3) $\rho = 0.1$

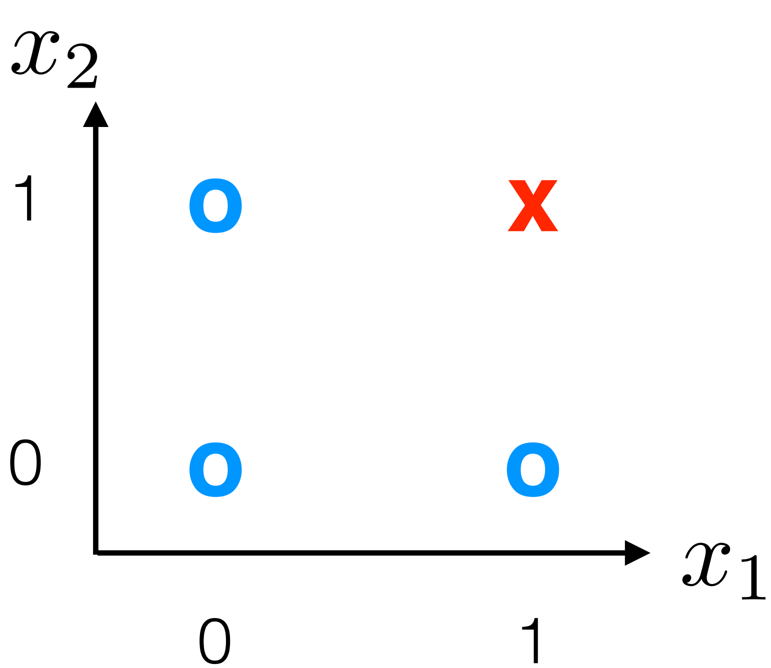
x_1	x_2	x_3	$w \cdot x$	o	t
-1	0	0	0.1*-1+0.2*0+0.3*0	0	0
-1	0	1	0.1*-1+0.2*0+0.3*1	0	0
-1	1	0	0.1*-1+0.2*1+0.3*0	0	0
-1	1	1	0.1*-1+0.2*1+0.3*1	**0**	**1**

Iteration 2, f(x)=x>0.5, w=(0, 0.3, 0.4) $\rho = 0.1$

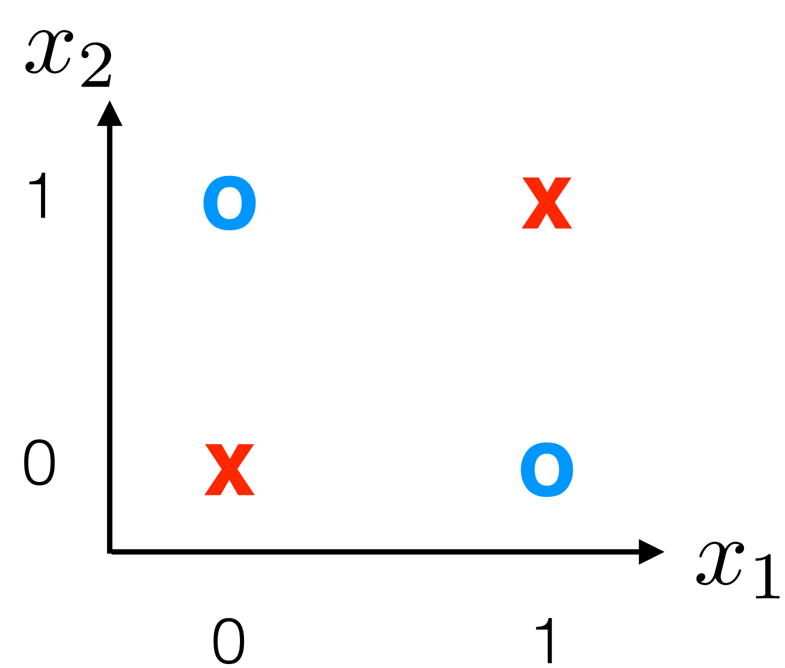
x_1	x_2	x_3	$w \cdot x$	o	t
-1	0	0	-1*0+0.3*0+0.4*0	0	0
-1	0	1	-1*0+0.3*0+0.4*1	0	0
-1	1	0	-1*0+0.3*1+0.4*0	0	0
-1	1	1	-1*0+0.3*1+0.4*1	1	1

Perceptron

- But perceptron can do only linear separations.
- In the 70-80 researchers hit this problem.



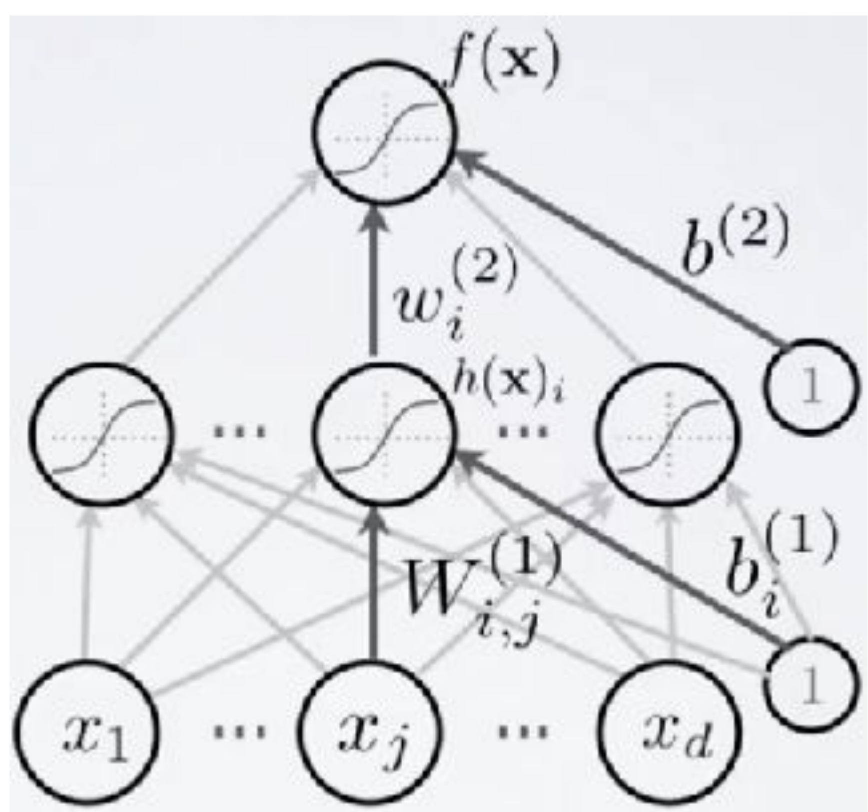
AND



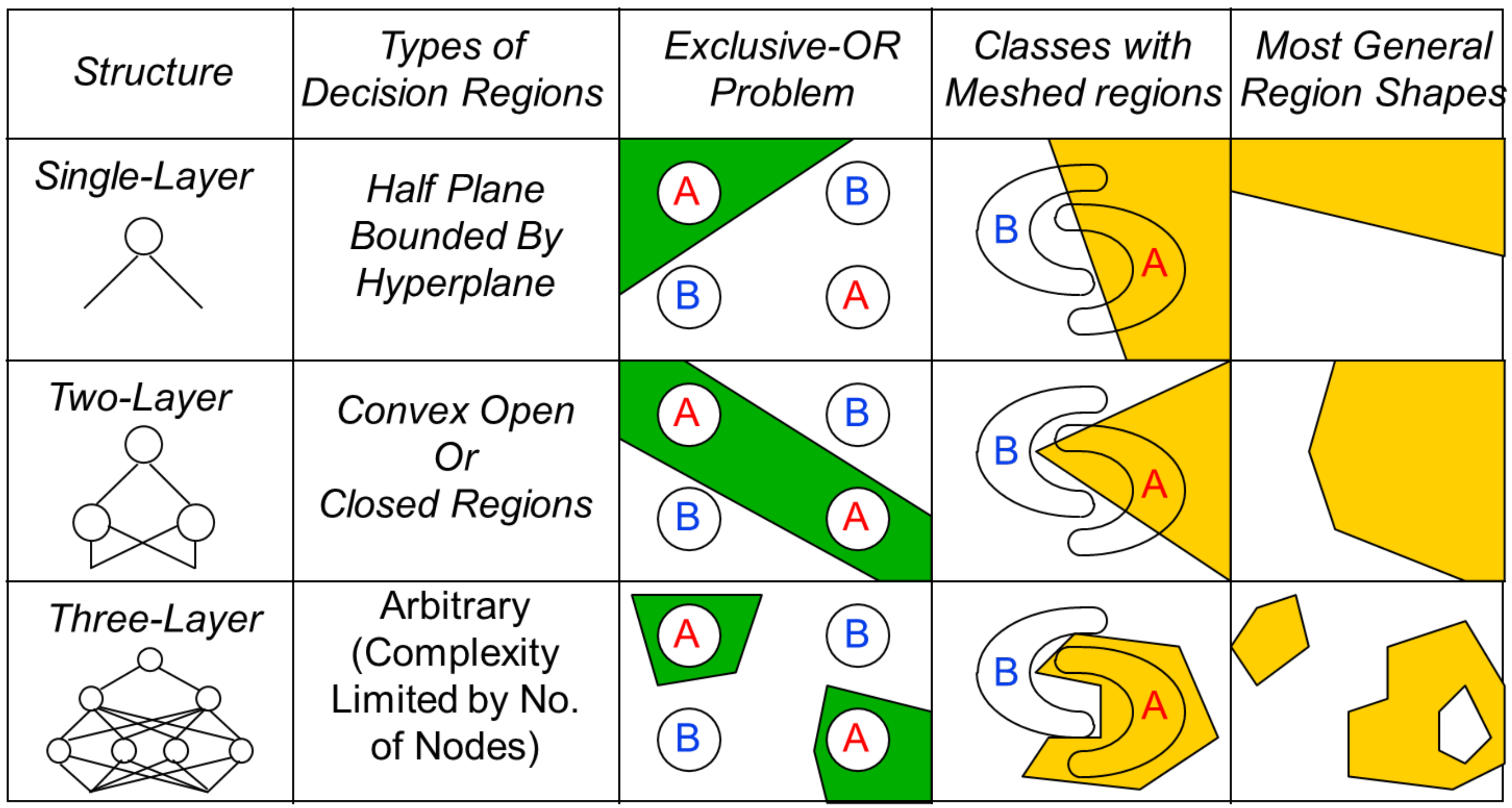
XOR

Multi-Layer Perceptron

- What about more layers? (*MultiLayer Perceptron o MLP*)
  - For more complex problems
  - Solve classification problems that are not linearly separable
  - Learning must be propagated between layers



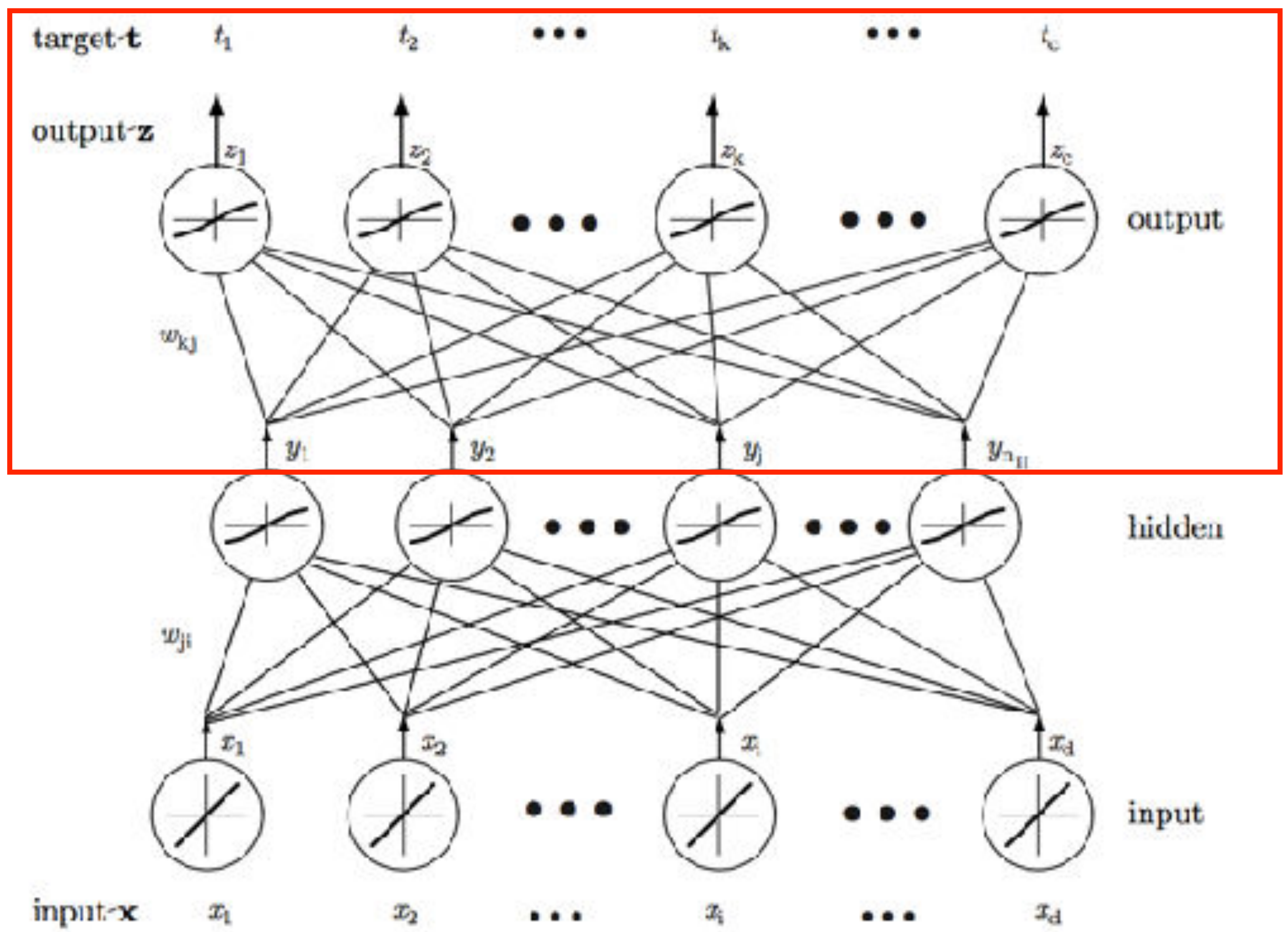
MLP



- Three layers are enough in theory, but more may be useful in practice…

MLP: Backpropagation

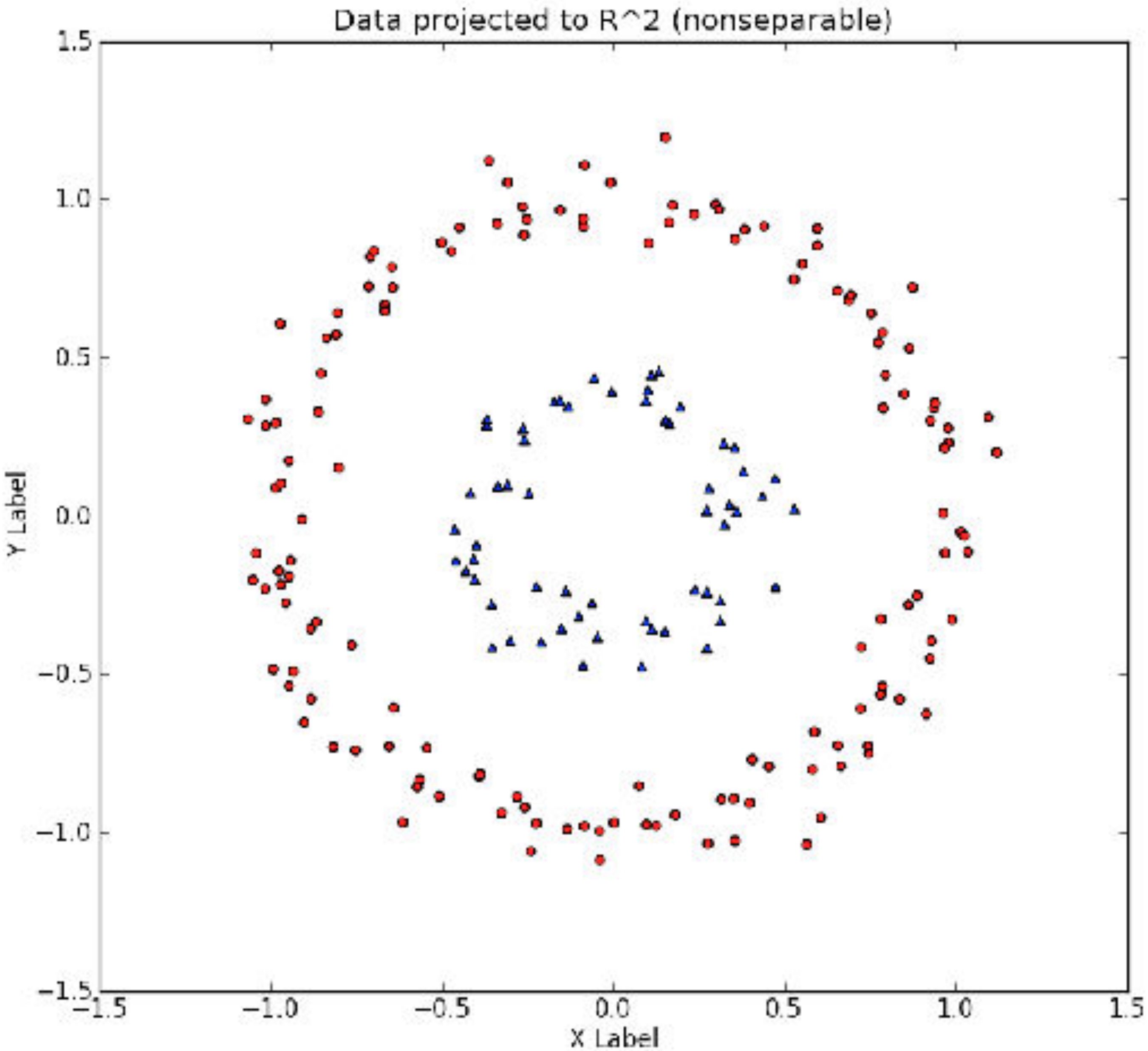
- In a multilayer network:

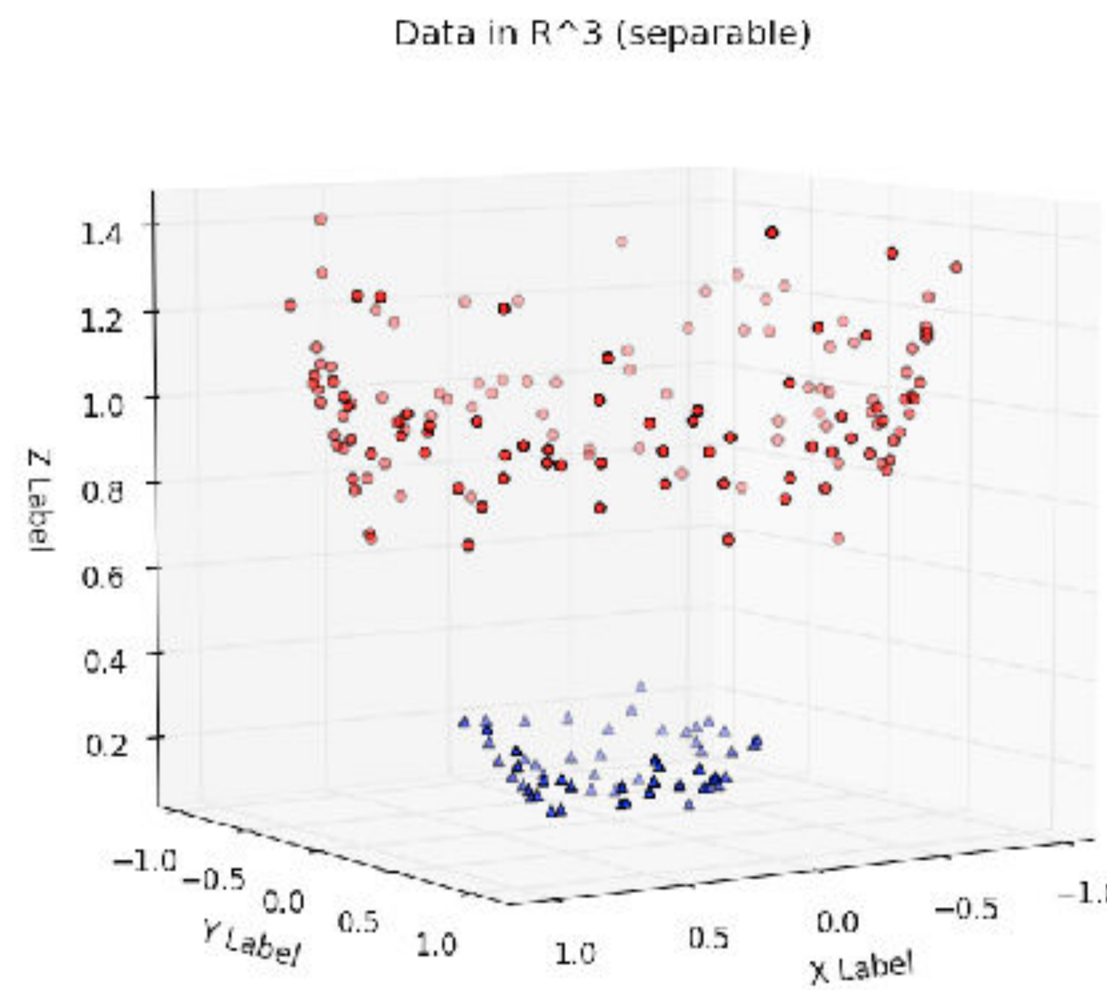


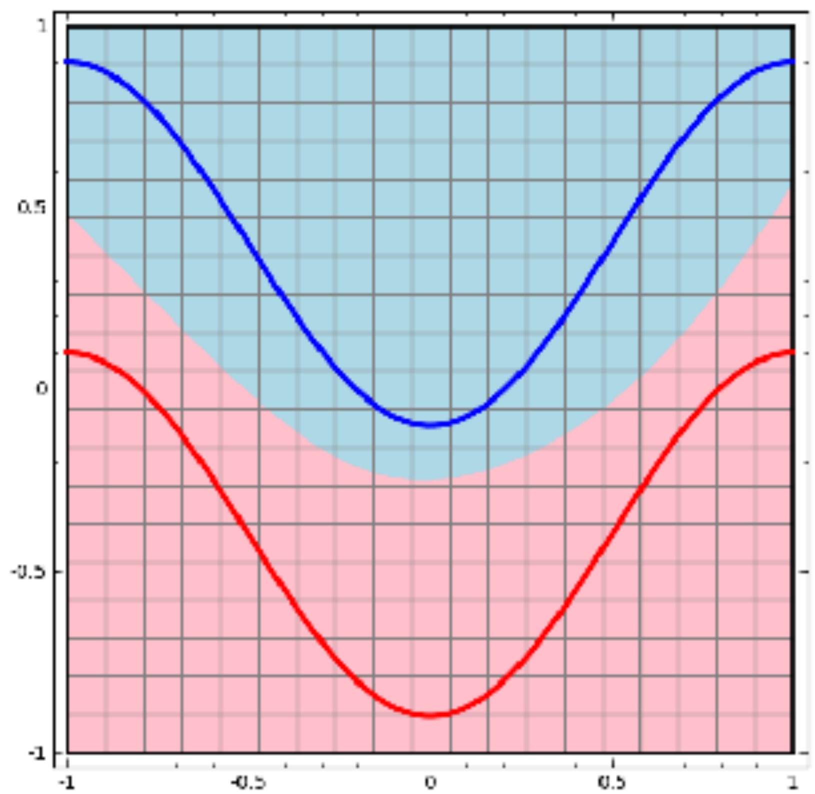
- Learning rule is:

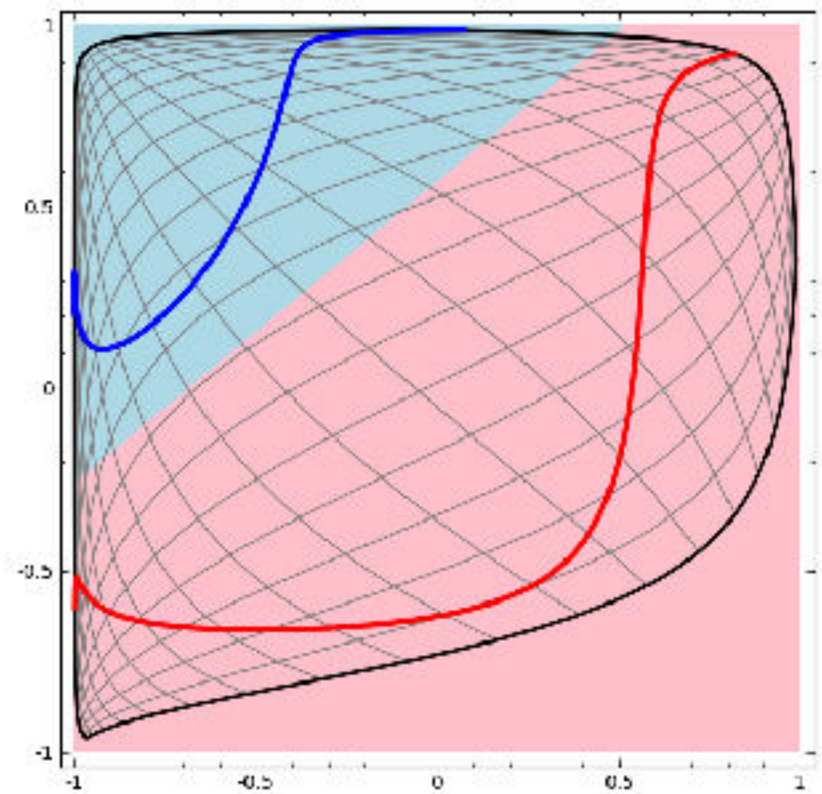
$$\Delta w_{kj} = -\frac{\partial J}{\partial net_k}\frac{\partial net_k}{\partial w_{kj}} = \rho(t_k - z_k)y_j f'(net_k)$$

Non-linear SVM





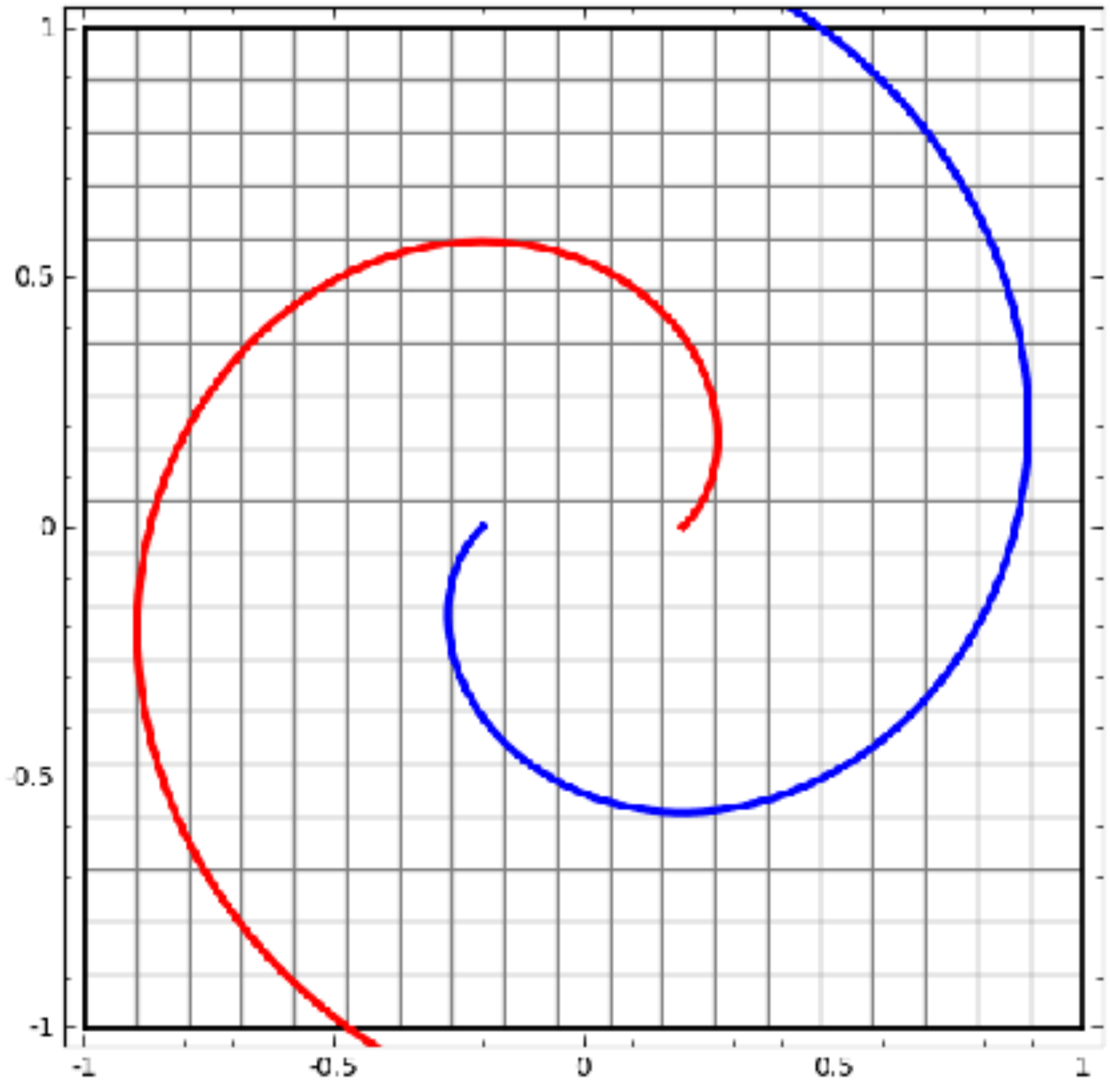


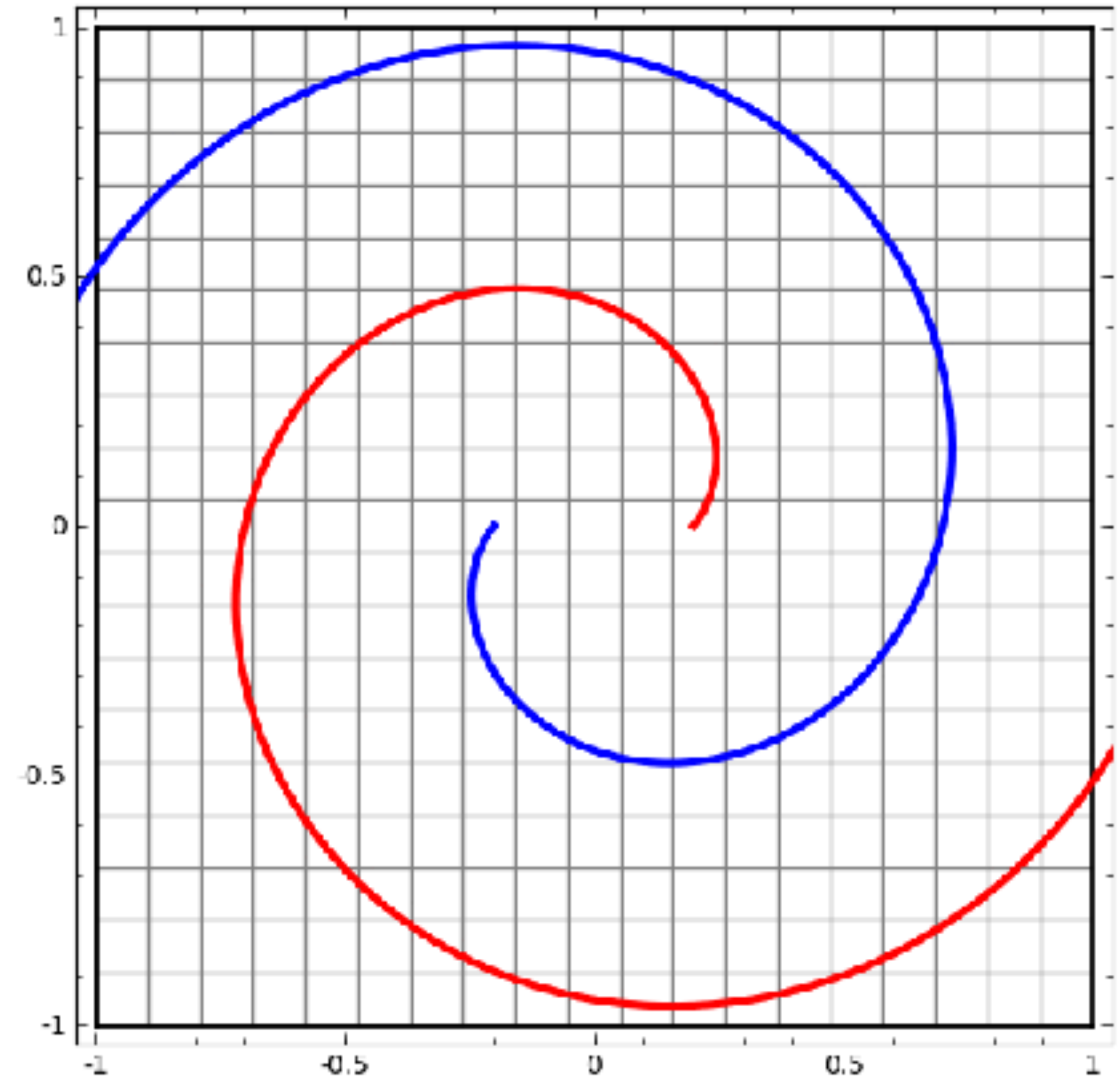


Mapping in order to linearly separate clusters

http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

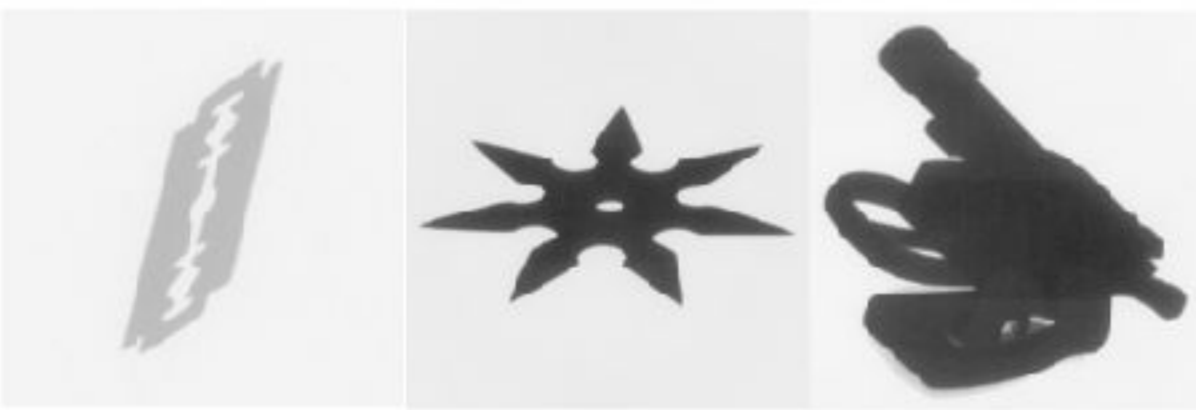
Disentangling with multiple layers





http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

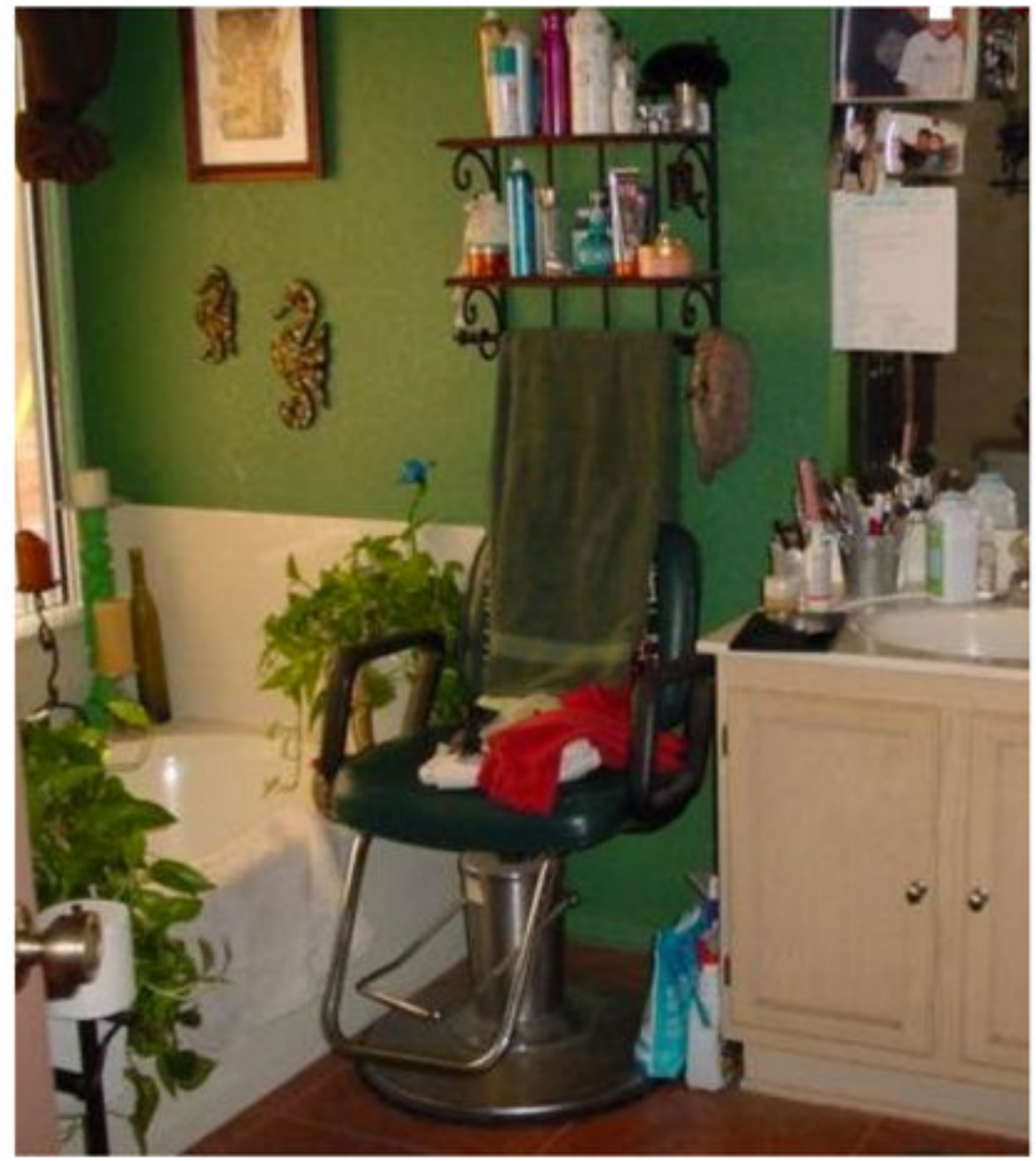
Easy



Difficult

Easy

Difficult



Difficult


Aeroplanes


Bicycles


Birds


Boats


Bottles


Buses


Cars


Cats


Chairs


Cows


Dining tables


Dogs


Horses


Motorbikes


People


Potted plants


Sheep


Sofas

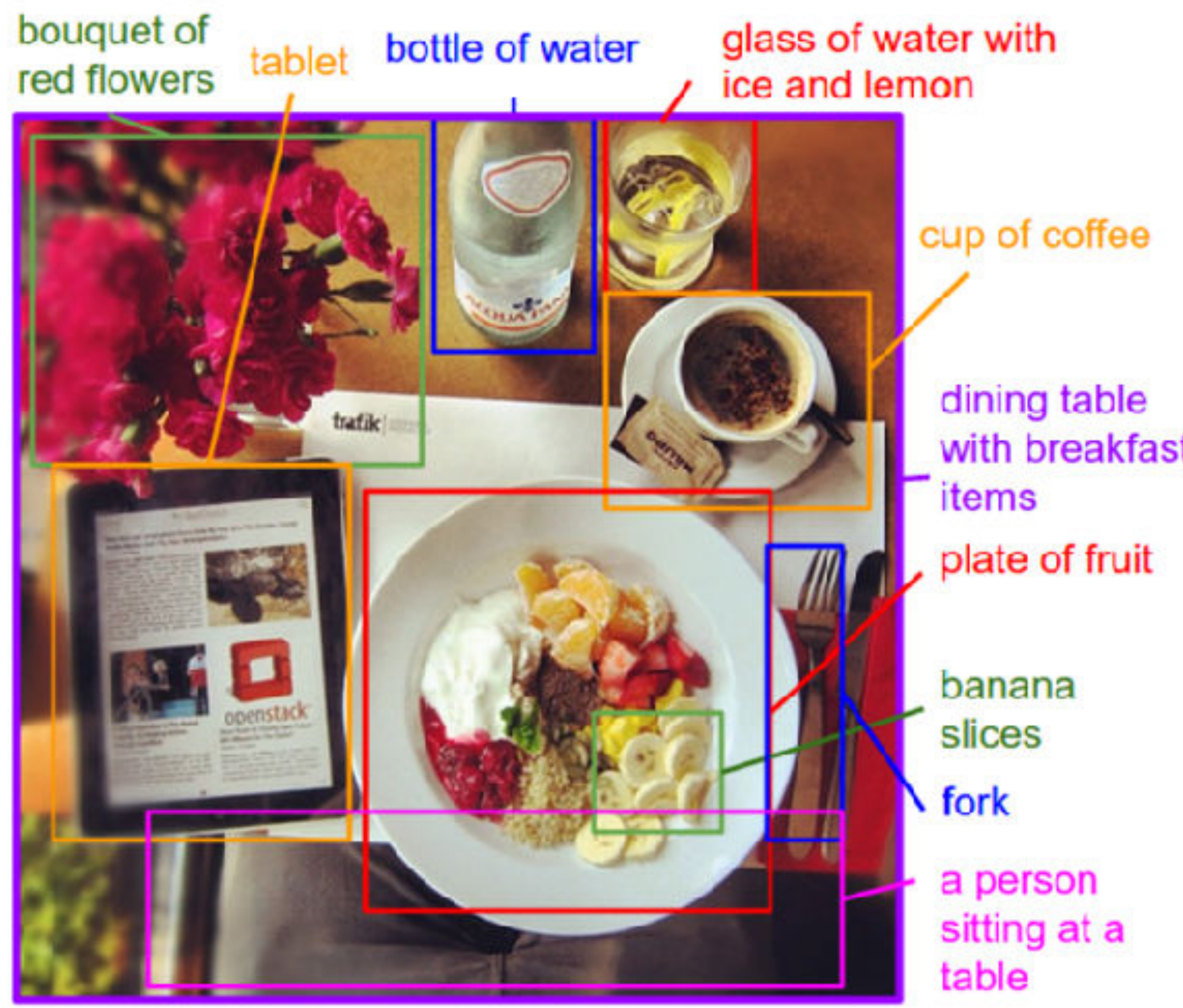

Trains


TV/Monitors

More Difficult

Laundry list for image archives

- Large sets
- Labelled data
- Metadata (**CDEs!**)
- Peripheral data
- Balanced datasets



Example output of the model

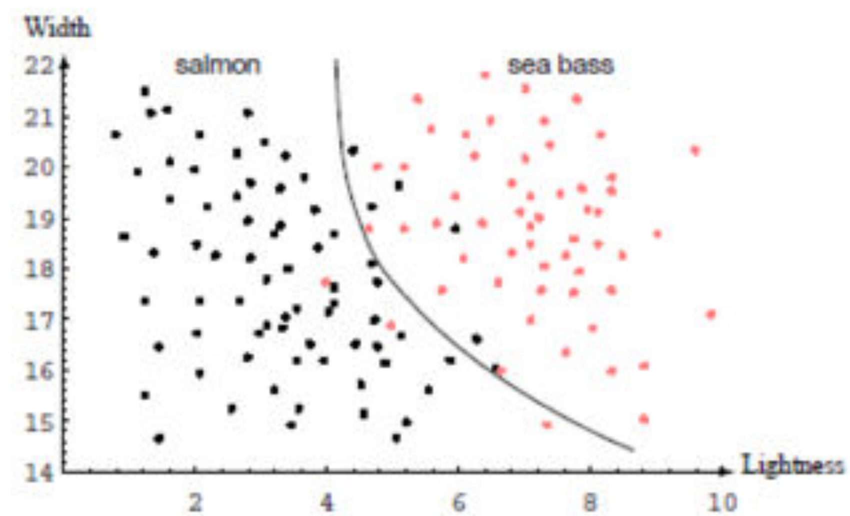
https://arxiv.org/pdf/1412.2306v2.pdf

Standard workflow

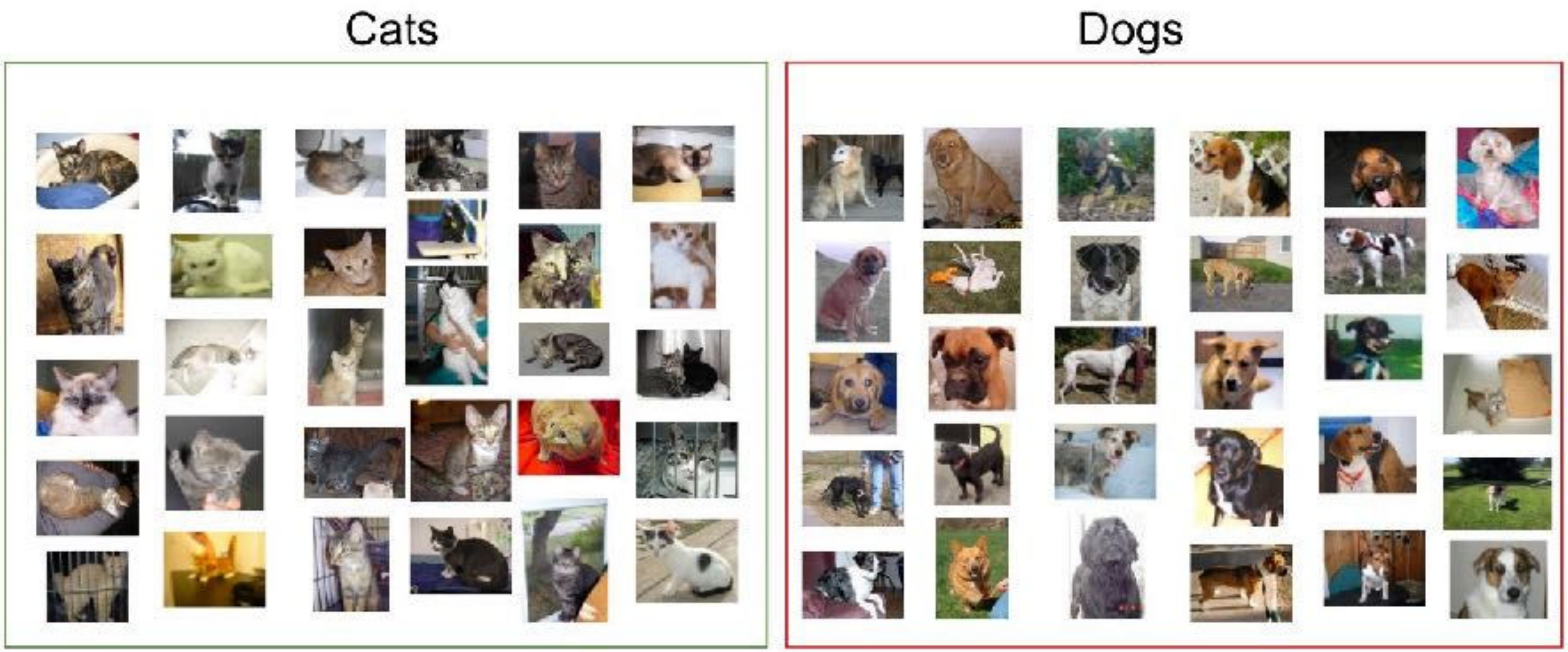
Data

$$\rightarrow [x_1, x_2, ... x_d]^T \rightarrow$$

Features

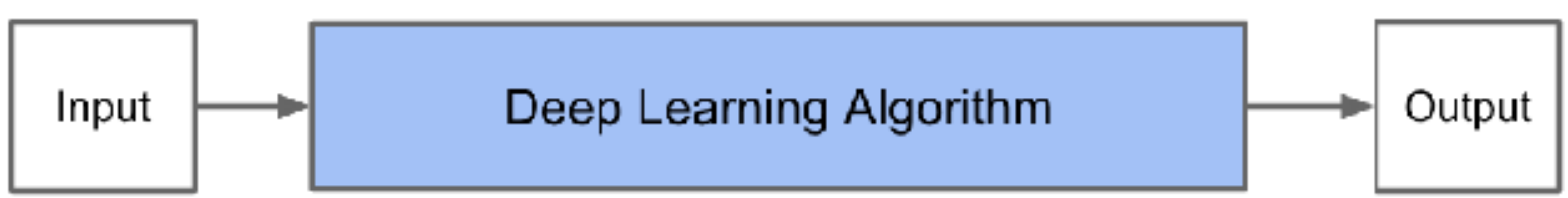


Classification



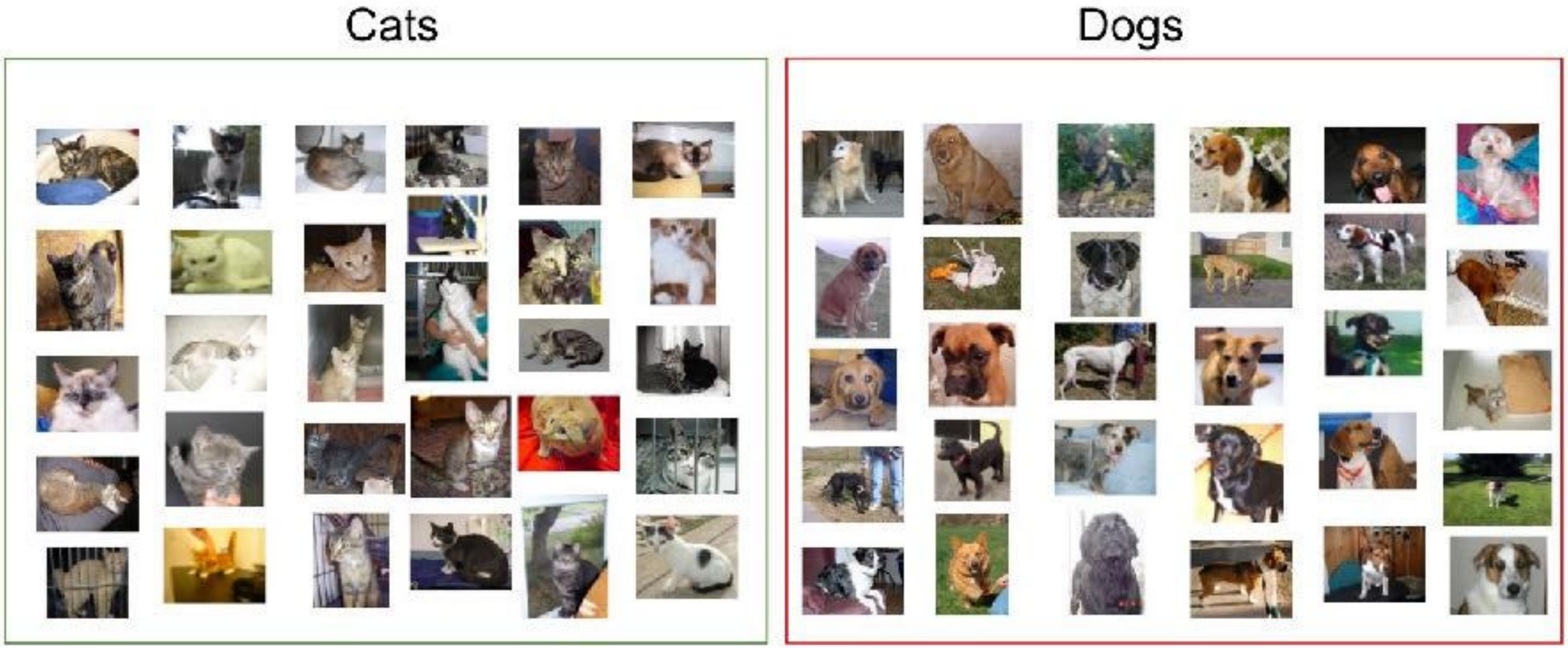
Sample of cats & dogs images from Kaggle Dataset

Traditional Machine Learning Flow



Deep Learning Flow

Adil Moujahid

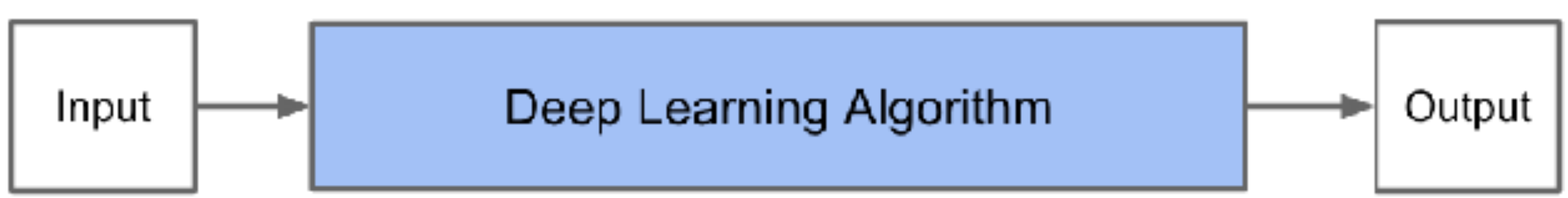


Sample of cats & dogs images from Kaggle Dataset

Promise:
Works better

Traditional Machine Learning Flow

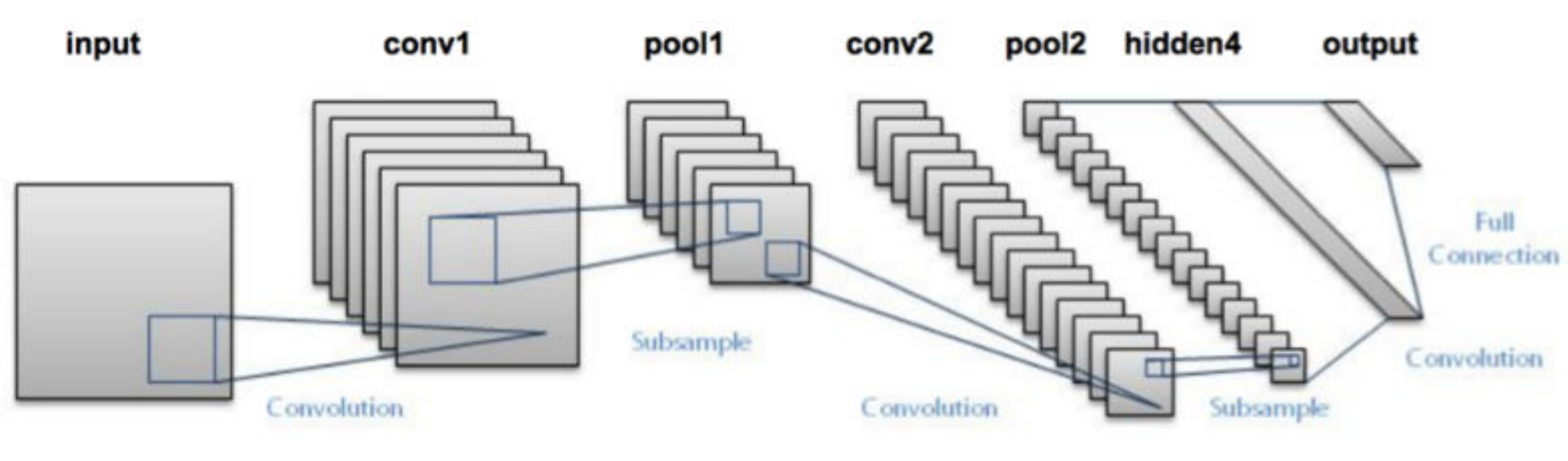
Pitfall:
Blacker box



Deep Learning Flow

Adil Moujahid

Convolutional network (single slide) primer



analyticsvidhya.com

INPUT IMAGE

18	54	51	239	244	188
55	121	75	78	95	88
35	24	204	113	109	221
3	154	104	235	25	130
15	253	225	159	78	233
68	85	180	214	245	0

WEIGHT

1	0	1
0	1	0
1	0	1

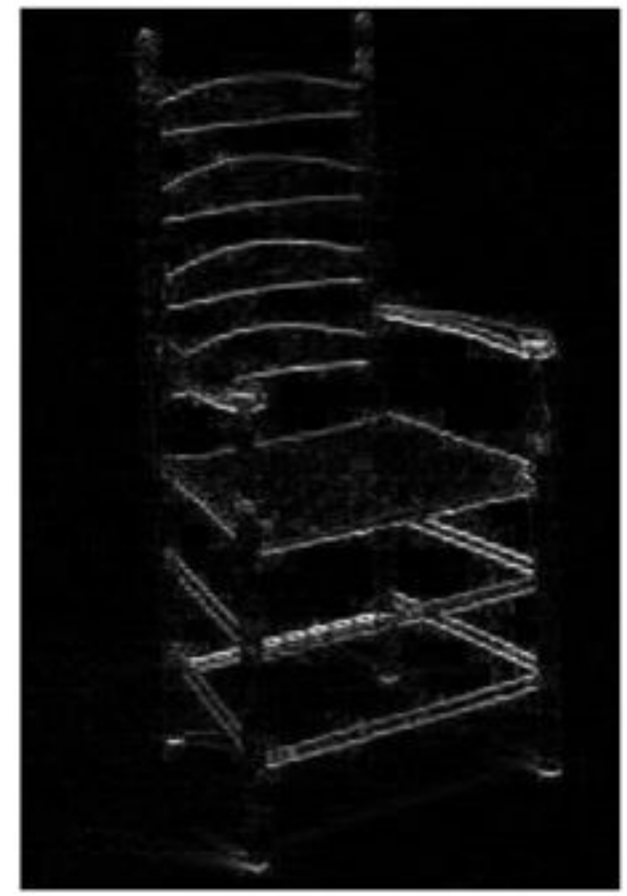
429

429	505	686	856
261	792	412	640
633	653	851	751
608	913	713	657

792	856
913	851

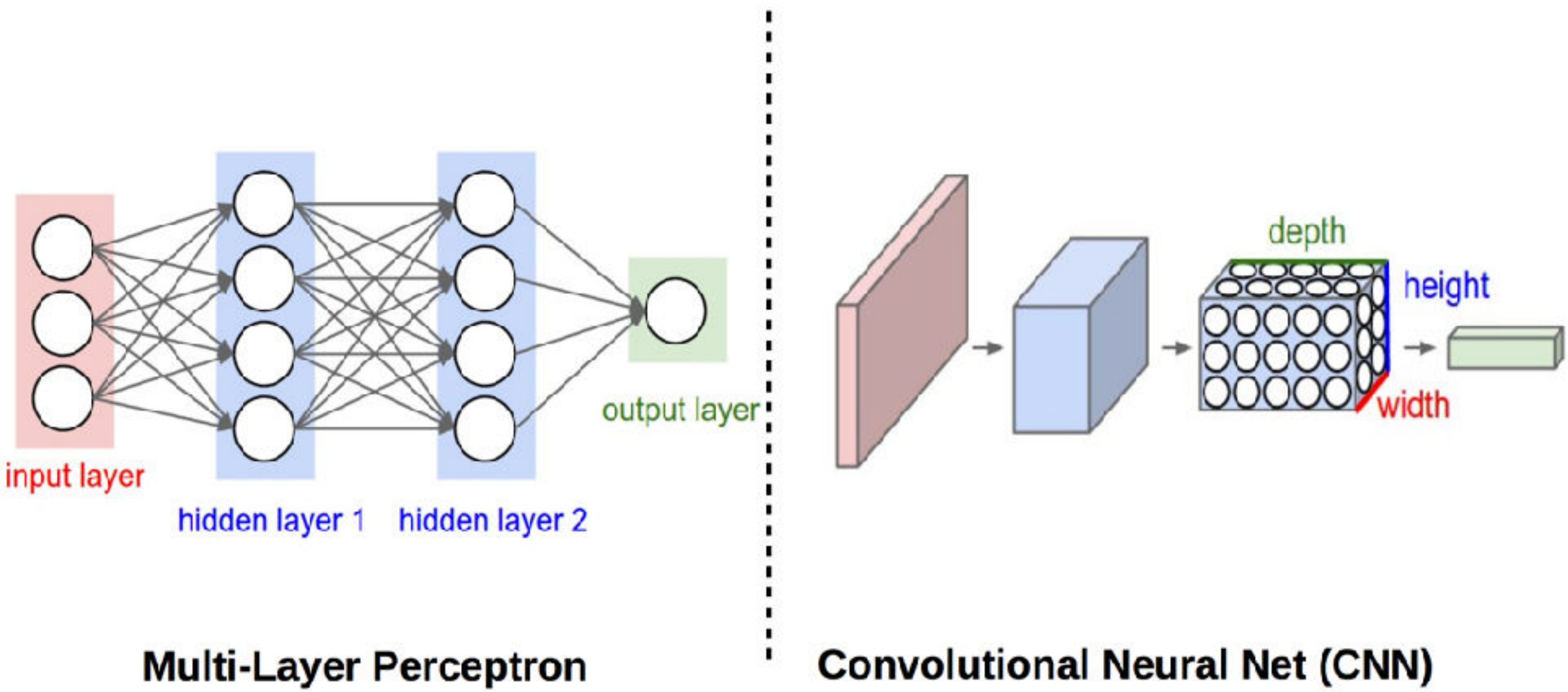
Convolution filter

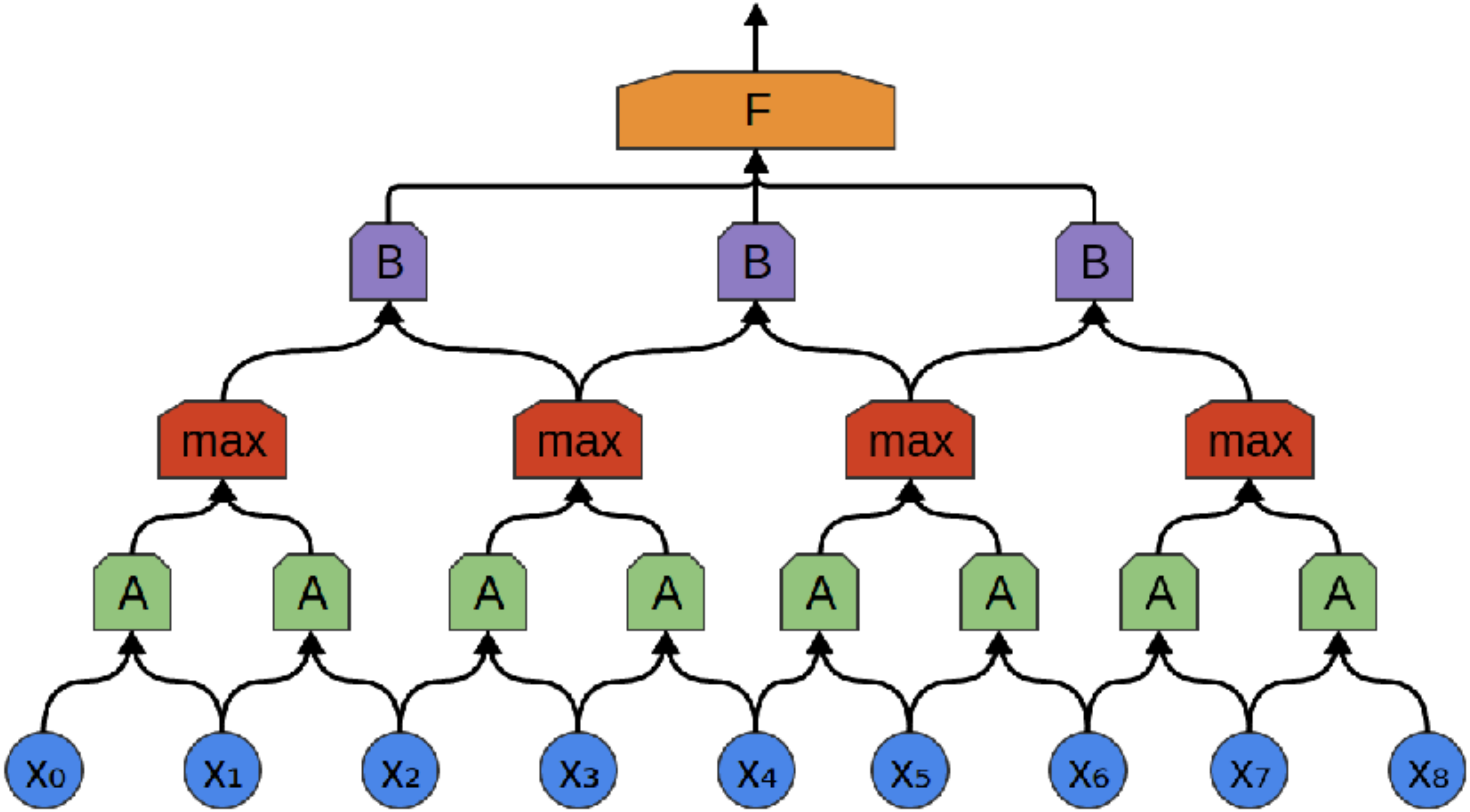
-1
0
1



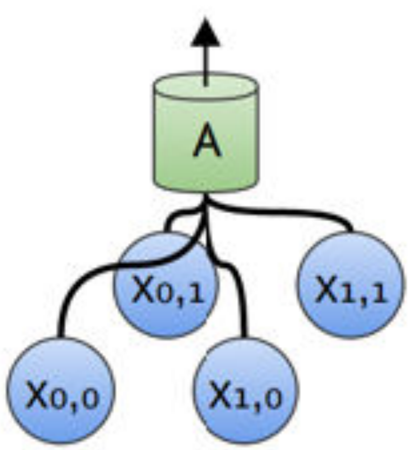
CONVOLUTION AS FEATURE EXTRACTOR

MLP & CNN


input layer
hidden layer 1
hidden layer 2
output layer
Multi-Layer Perceptron
depth
height
width
Convolutional Neural Net (CNN)

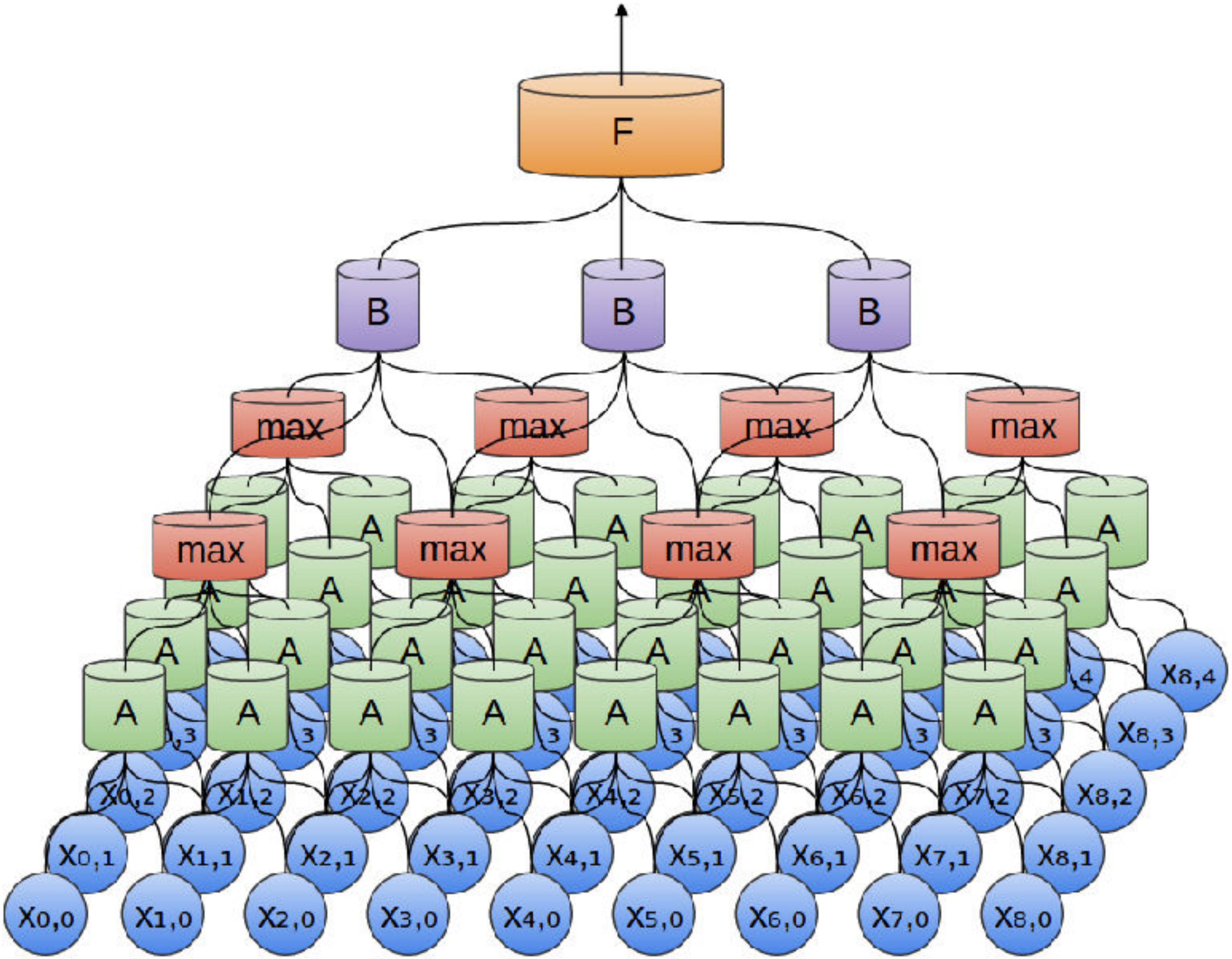


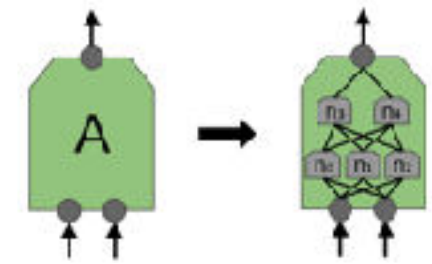
conv + pool + conv + connected



http://colah.github.io/posts/2014-07-Conv-Nets-Modular/

2D version of convolution





http://colah.github.io/posts/2014-07-Conv-Nets-Modular/

Final layers are fully connected

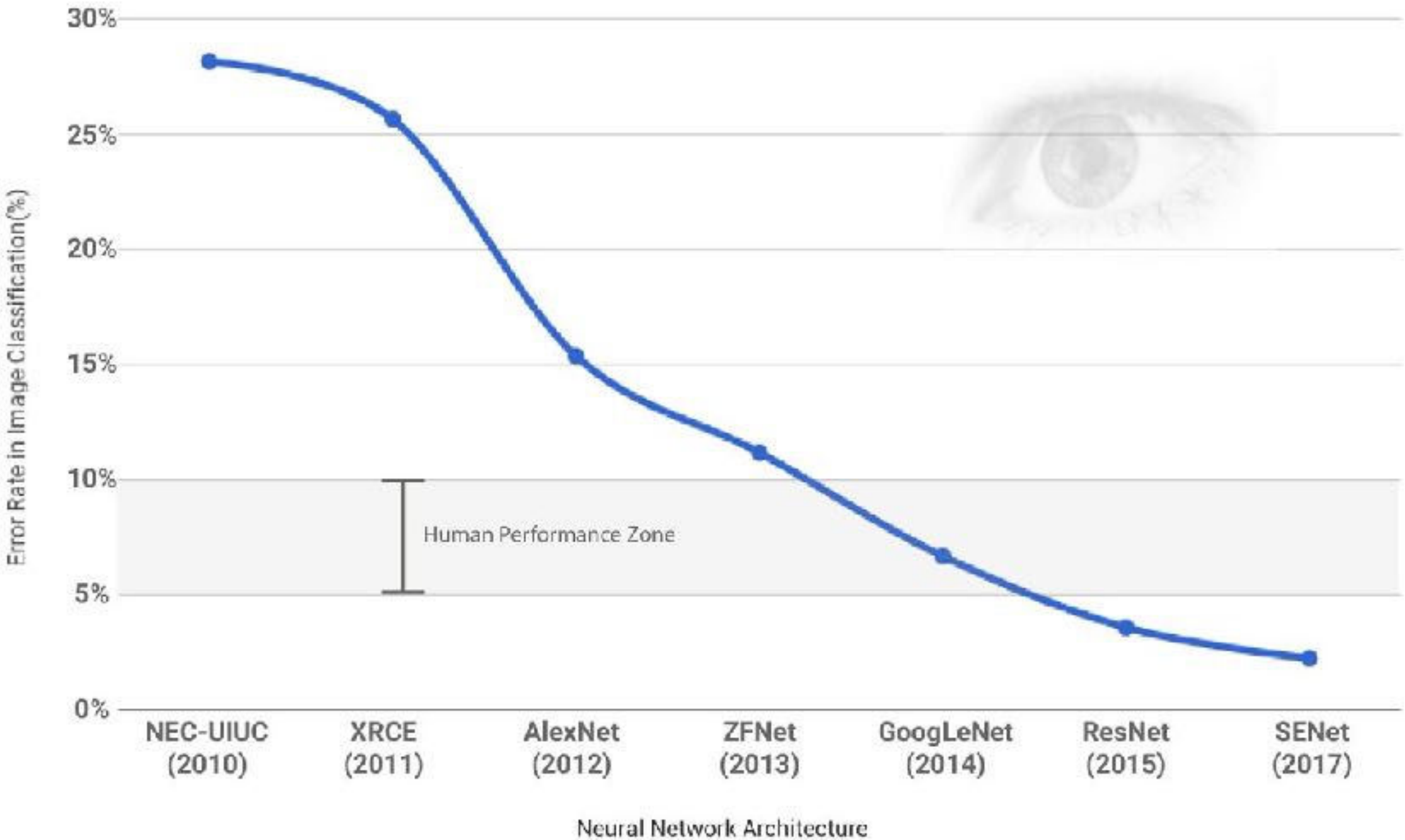
Several libraries available

- Theano: http://deeplearning.net/software/theano/
- Caffe: http://caffe.berkeleyvision.org/
- Tensorflow: https://www.tensorflow.org/
- MXnet: https://mxnet.apache.org/ (Gluon)
- PyTorch: https://pytorch.org/
- Keras: https://en.wikipedia.org/wiki/Keras

https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software

abstractions, instant gratification

Evolution of CNNs



ImageNet Competition

2015 ILSVRC leaderboard

Team name	Entry description	**Number of object categories won**	mean AP
MSRA	An ensemble for detection.	194	0.620741
Qualcomm Research	NeoNet ensemble with bounding box regression. Validation mAP is 54.6	4	0.535745
CUImage	Combined multiple models with the region proposals of cascaded RPN, 57.3% mAP on Val2.	2	0.527113
The University of Adelaide	9 models	0	0.514434
MCG-ICT-	2 models on 2 proposals without category information: {[SS+EB]+		

Classification error: 0.03567

Yellow: Winner in category
Yellow/White: Reveal code
Gray: Won't reveal code

2016 ILSVRC leaderboard

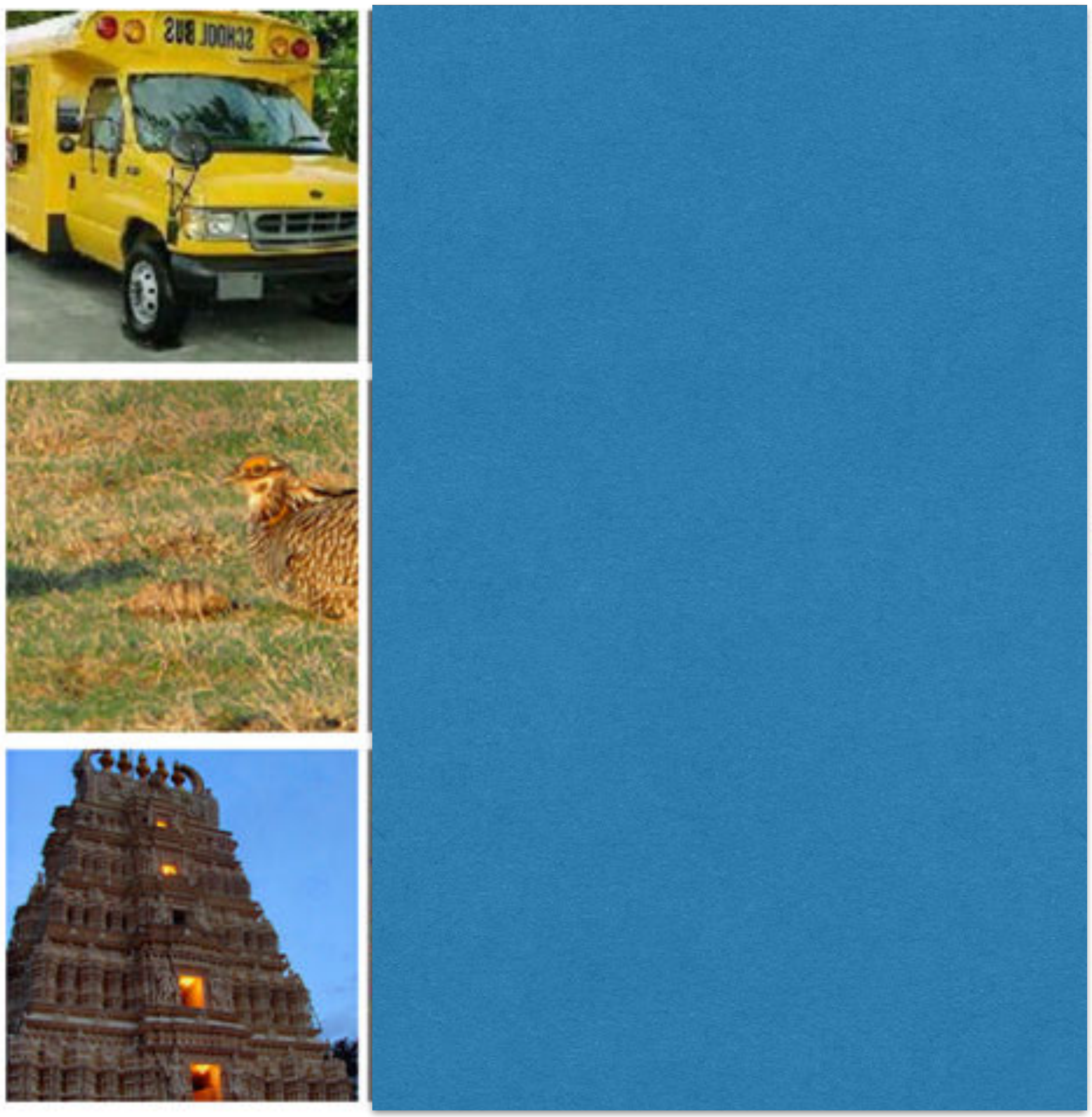
Team name	Entry description	Number of object categories won	mean AP
CUImage	Ensemble of 6 models using provided data	109	0.662751
Hikvision	Ensemble A of 3 RPN and 6 FRCN models, mAP is 67 on val2	30	0.652704
Hikvision	Ensemble B of 3 RPN and 5 FRCN models, mean AP is 66.9, median AP is 69.3 on val2	18	0.652003
NUIST	submission_1	15	0.608752
NUIST	submission_2	9	0.607124
Trimps-Soushen	Ensemble 2	8	0.61816
360+MCG-ICT-CAS_DET	9 models ensemble with validation and 2 iterations	4	0.615561
360+MCG-ICT-CAS_DET	Baseline: Faster R-CNN with Res200	4	0.590596
Hikvision	Best single model, mAP is 65.1 on val2	2	0.634003
CIL	Ensemble of 2 Models	1	0.553542
360+MCG-ICT-CAS_DET	9 models ensemble	0	0.613045

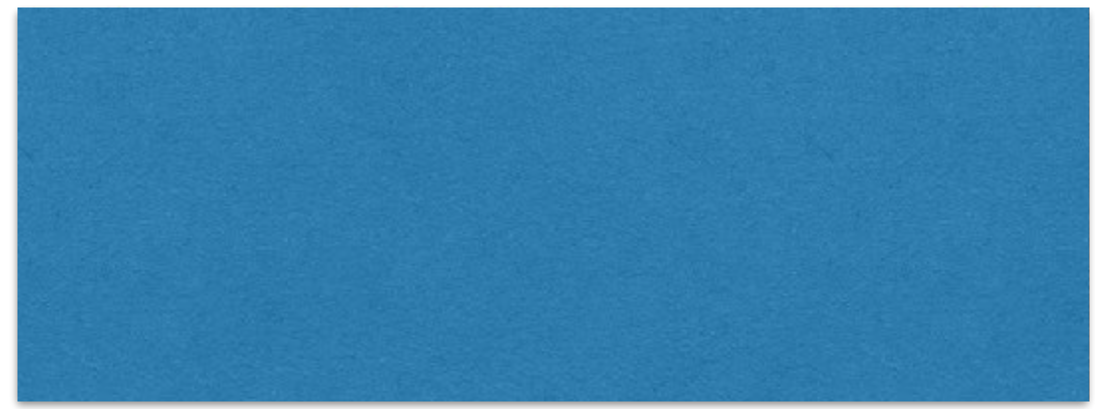
Classification error:
0.02991

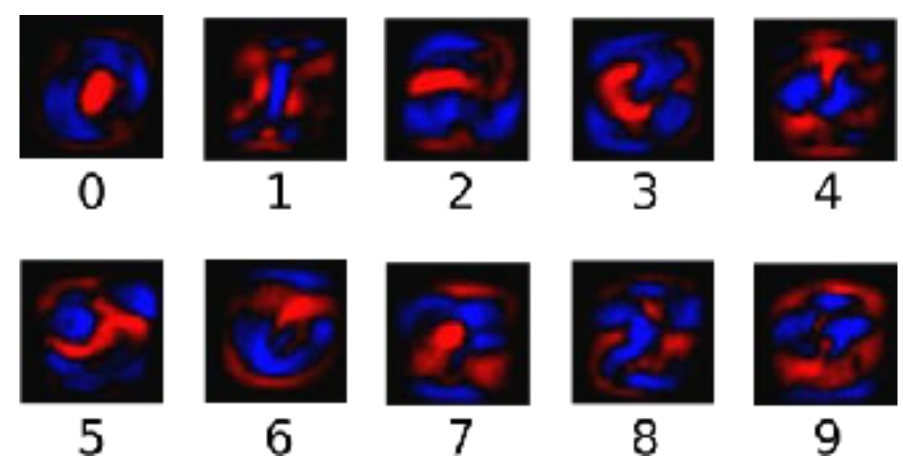
What bird is that?

or: what features is my deep network using?

Including adversarial examples during training



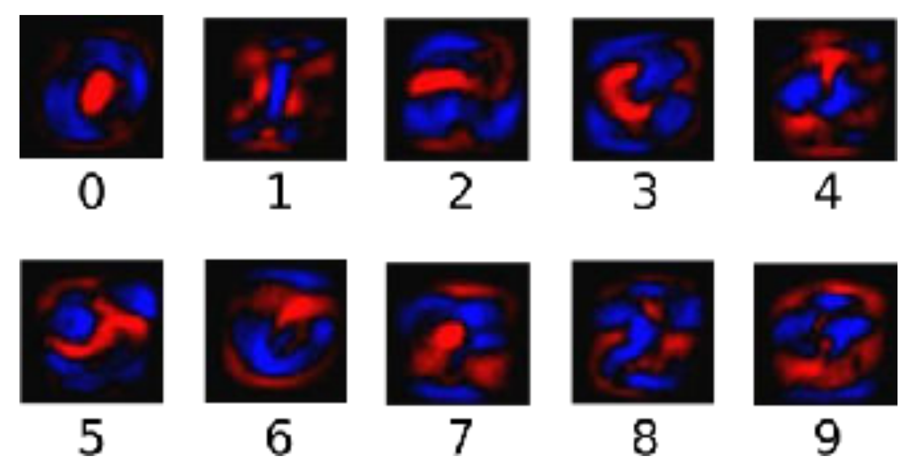




MNIST digits
significance map

https://arxiv.org/pdf/1312.6199v4.pdf

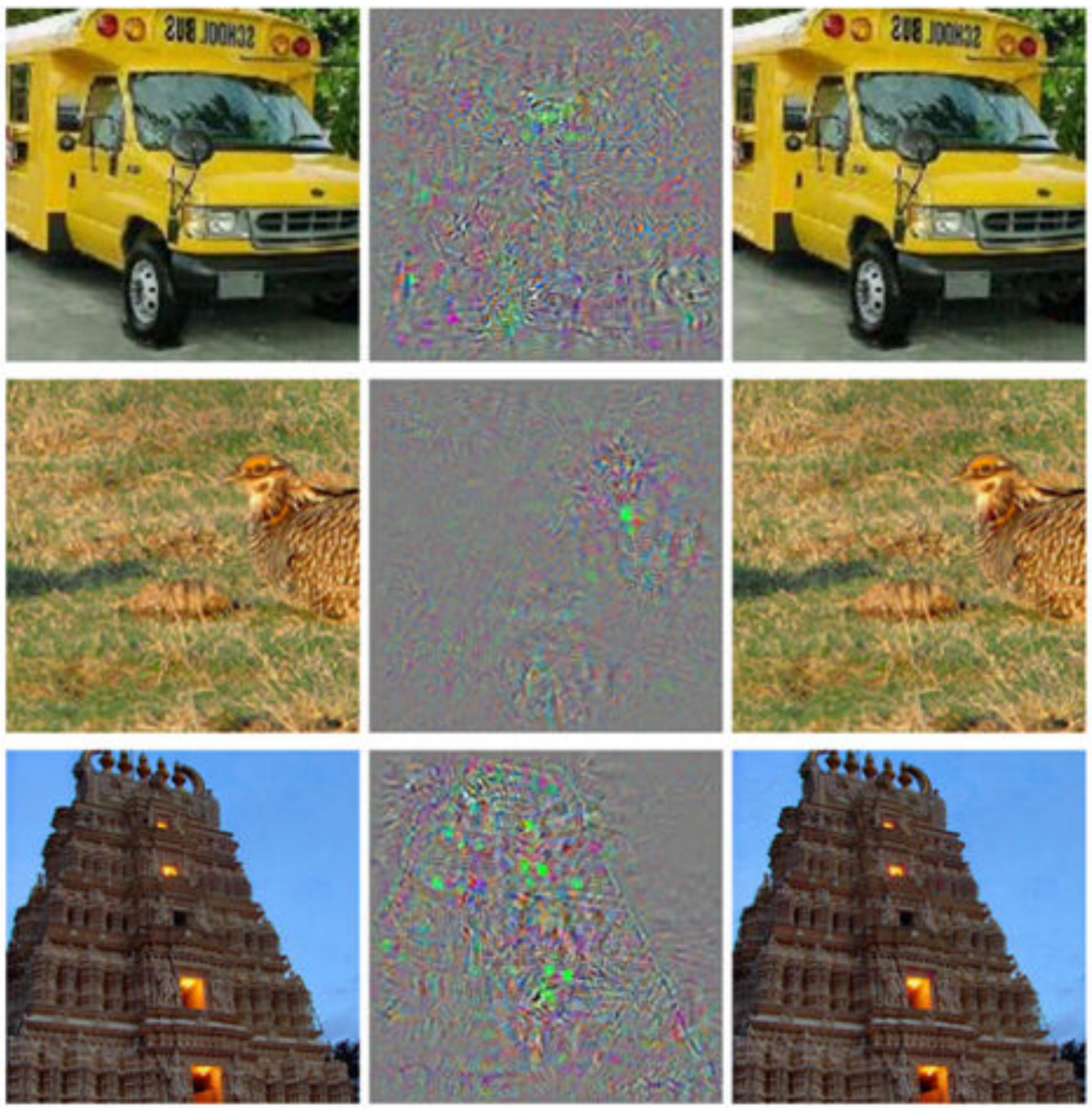
Including adversarial examples during training



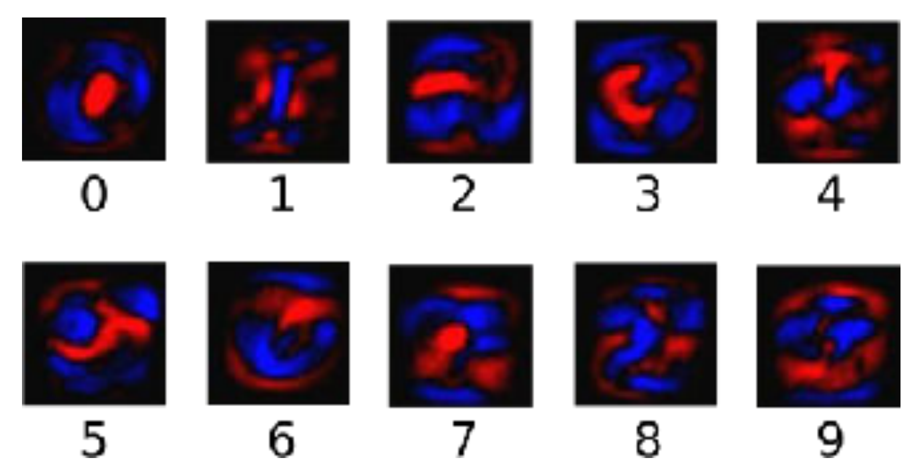
MNIST digits significance map

https://arxiv.org/pdf/1312.6199v4.pdf

Including adversarial examples during training



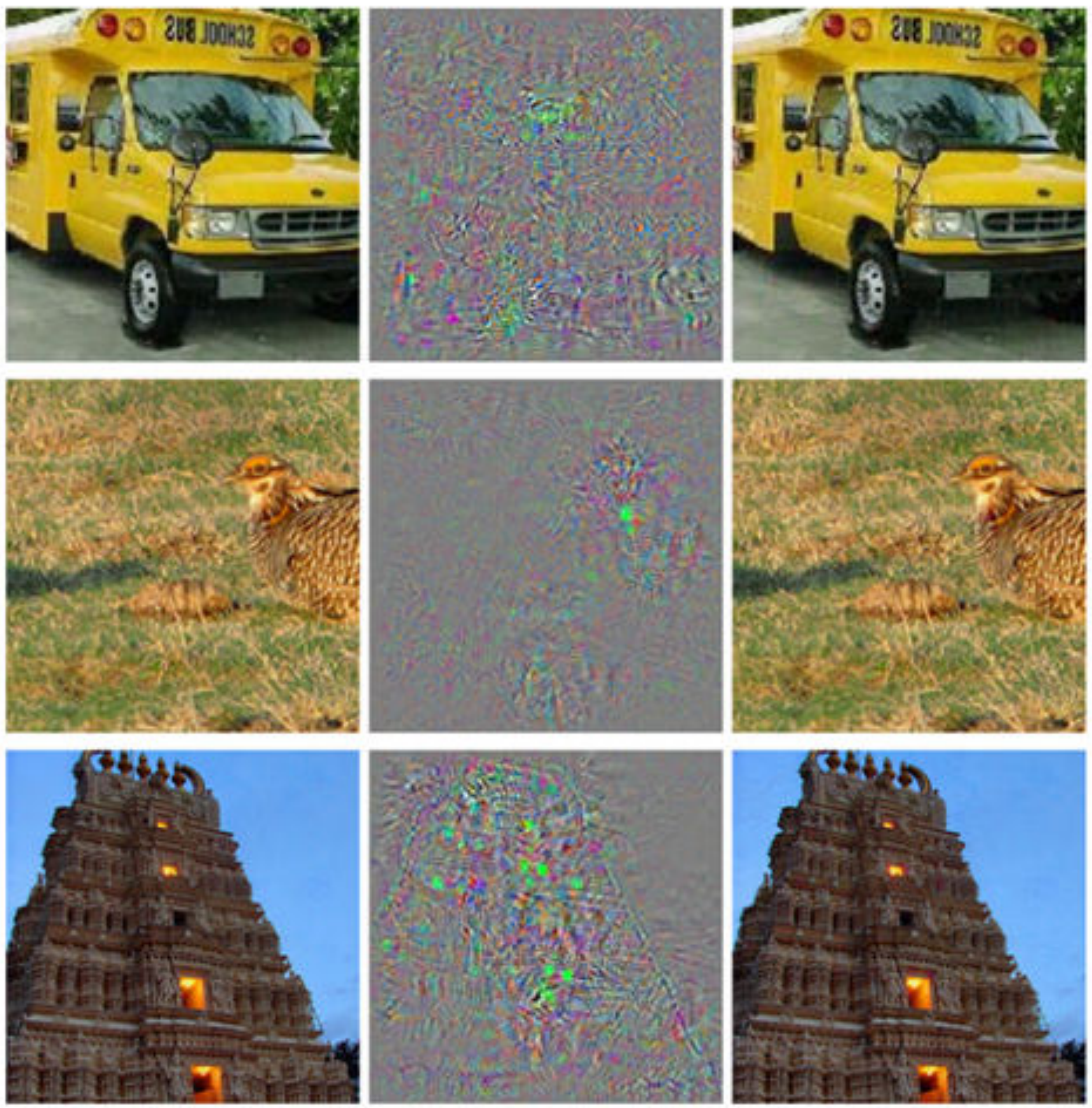
The images in the left most column are correctly classified examples. The middle column represents the distortion between the left and right images. The images in the right most column are predicted to be of the class ostrich! Even though the difference between the images on the left and right is imperceptible to humans, the ConvNet makes drastic errors in classification.



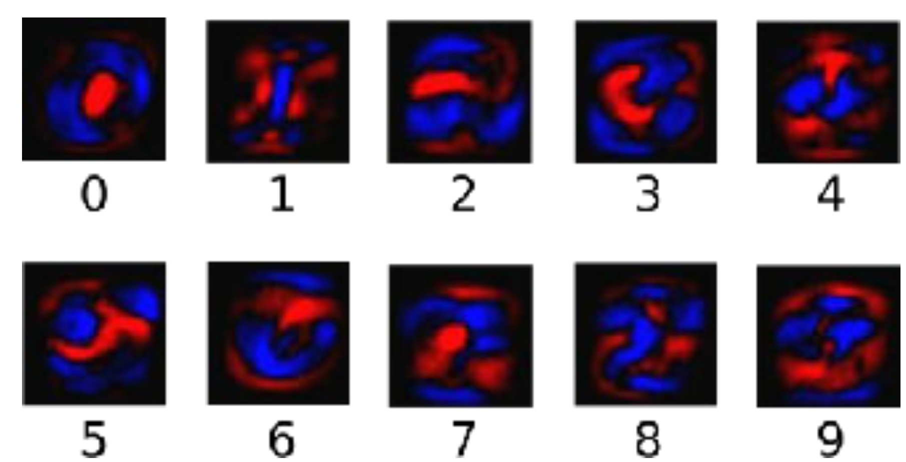
MNIST digits significance map

https://arxiv.org/pdf/1312.6199v4.pdf

Including adversarial examples during training



The images in the left most column are correctly classified examples. The middle column represents the distortion between the left and right images. The images in the right most column are predicted to be of the class ostrich! Even though the difference between the images on the left and right is imperceptible to humans, the ConvNet makes drastic errors in classification.

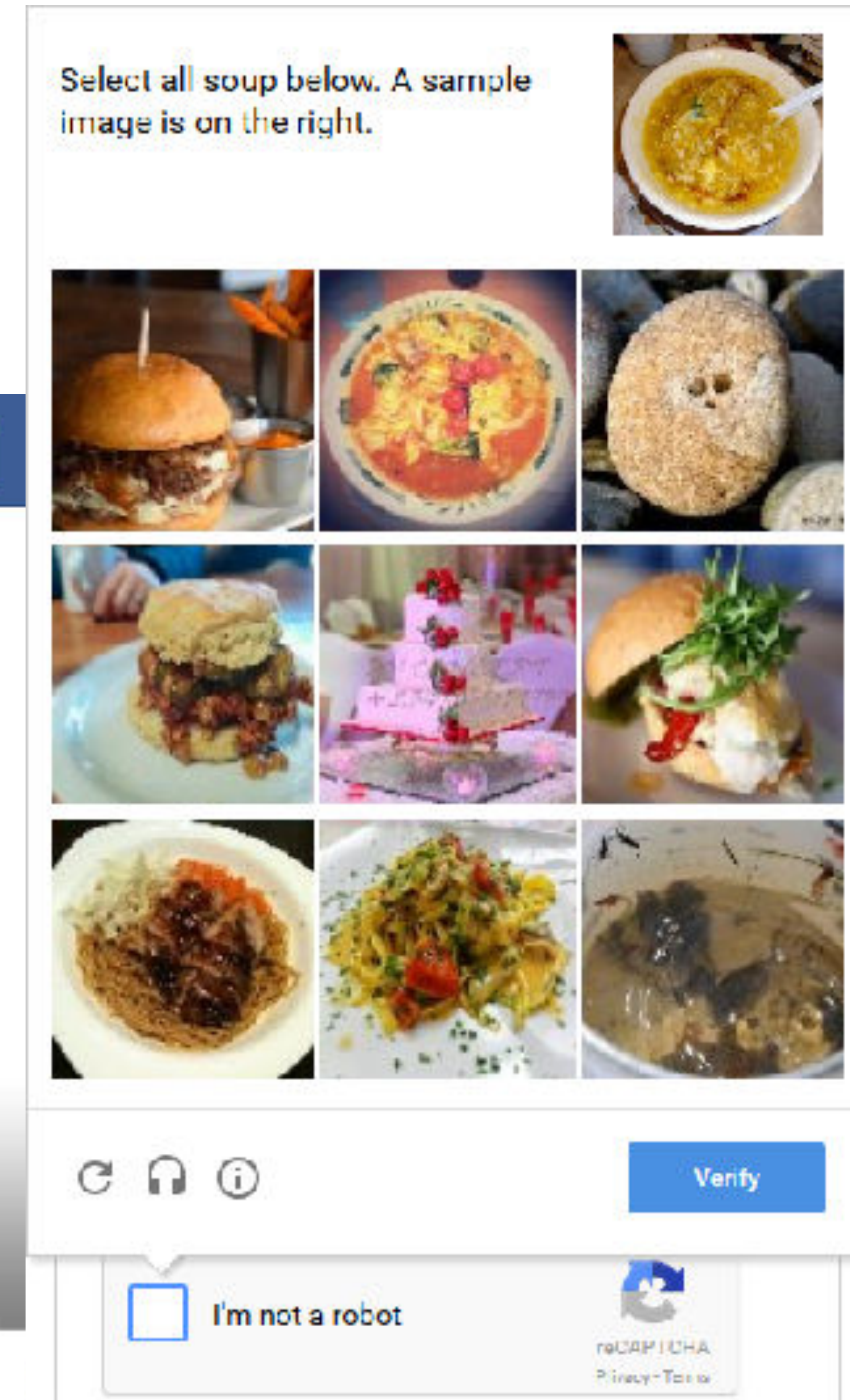


MNIST digits significance map

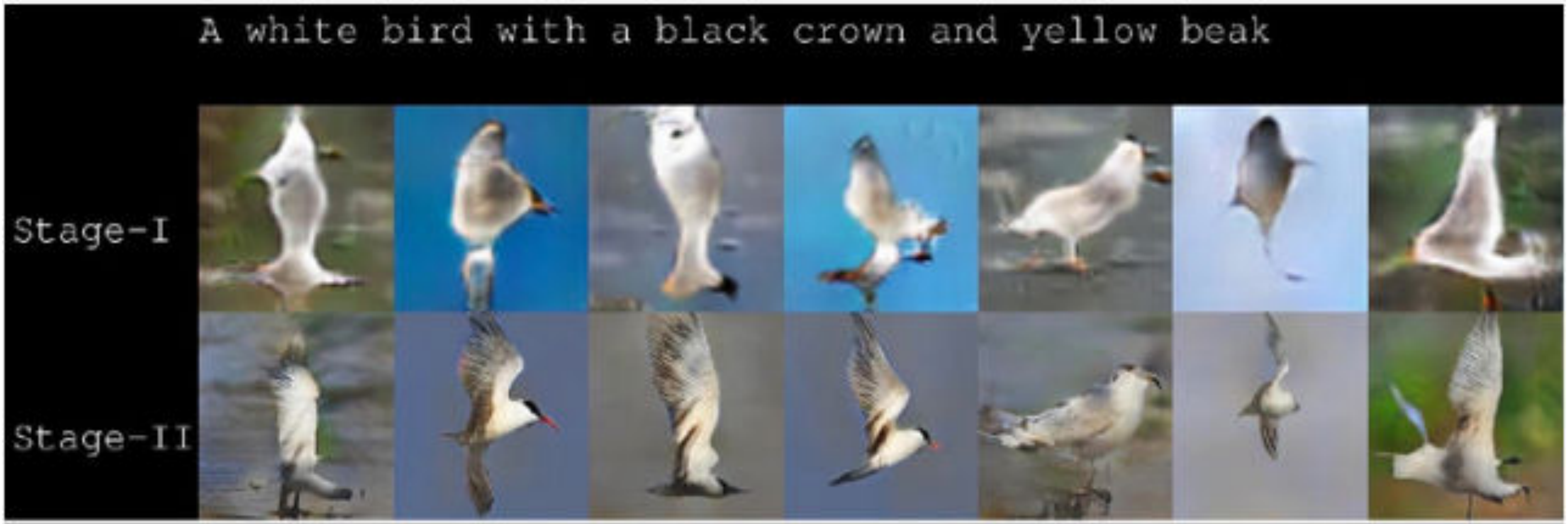
https://arxiv.org/pdf/1312.6199v4.pdf

Pitfall: Overlearning

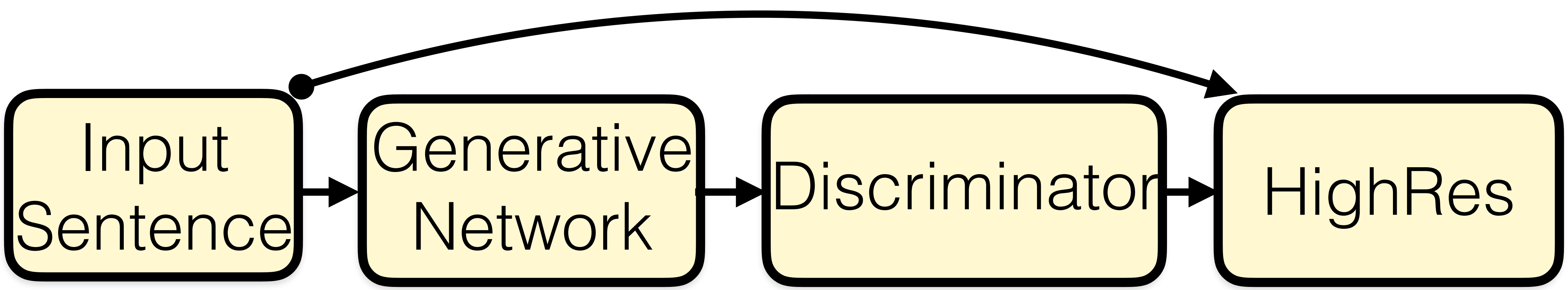
Labels are everywhere



Generative Adversarial Networks

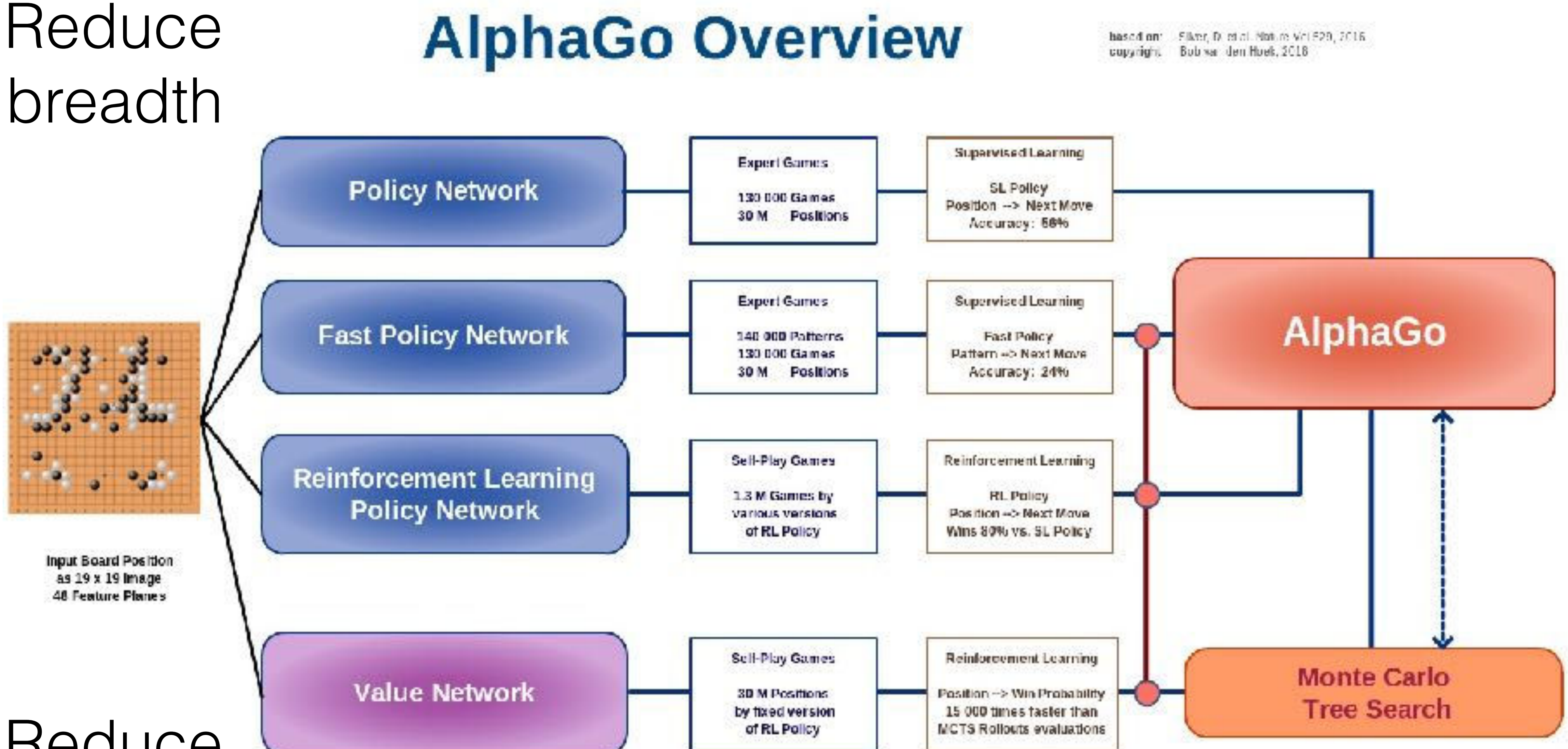


Zhang et al. 2016



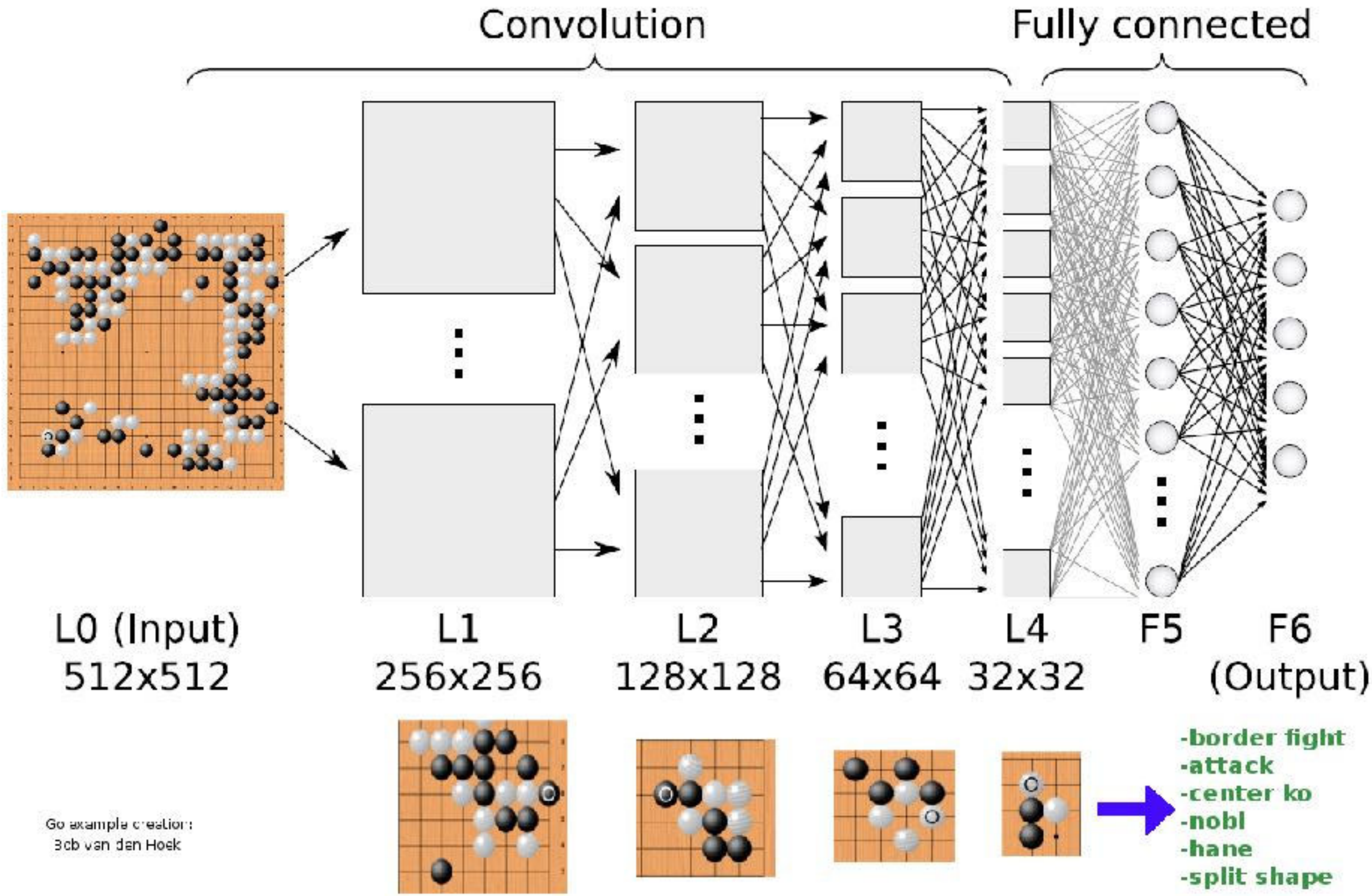
Reality check

Reduce breadth



Reduce depth

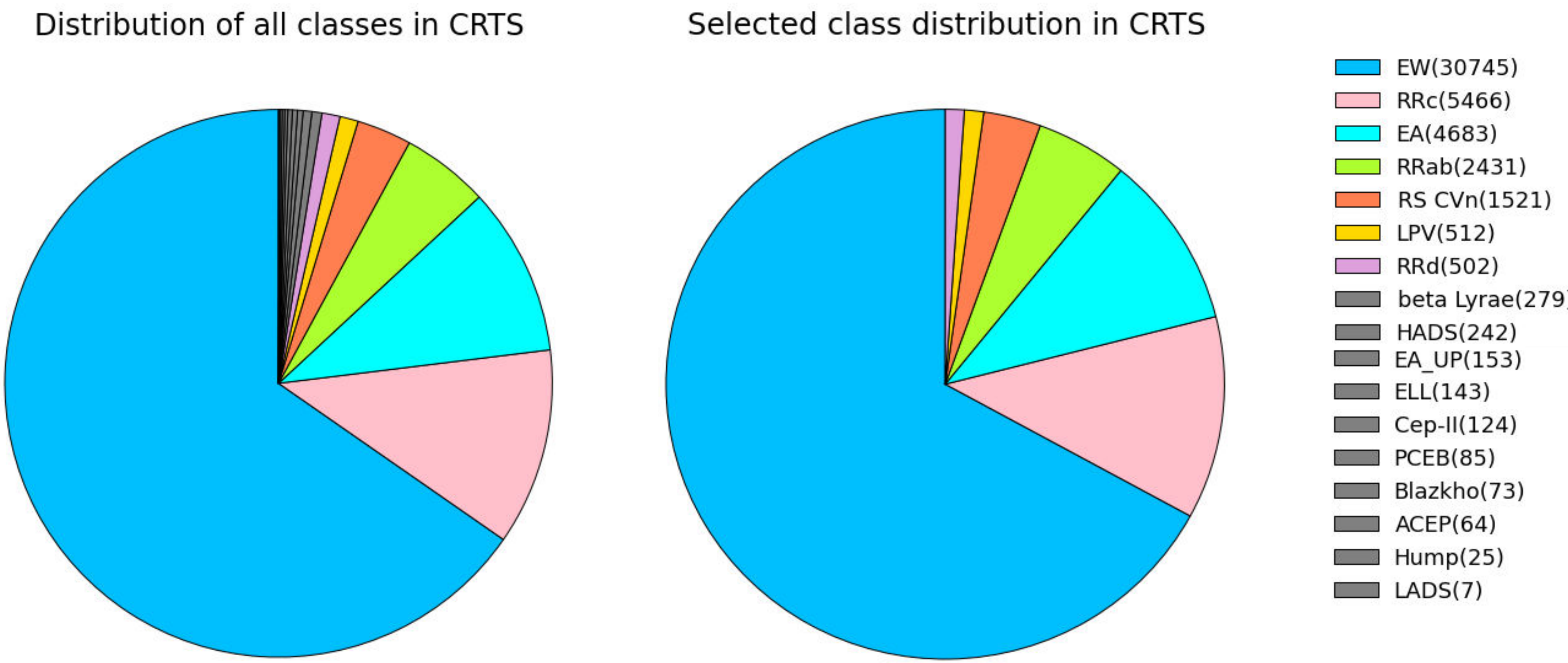
http://deeplearningskysthelimit.blogspot.com/2016/04/part-2-alphago-under-magnifying-glass.html



http://gobase.org/online/intergo/?query=%22hane%20nobi%22

http://deeplearningskysthelimit.blogspot.com/2016/04/part-2-alphago-under-magnifying-glass.html

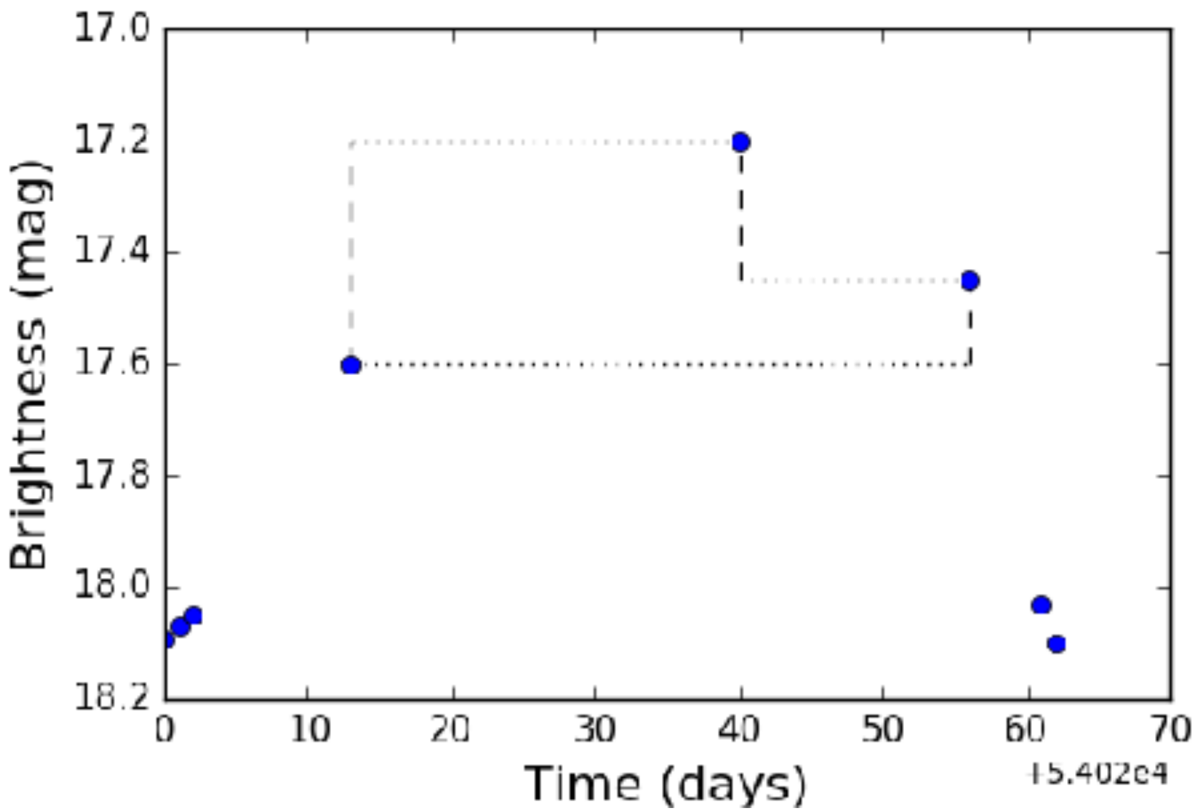
50K Periodic Variables from CRTS



Drake et al. 2014

(dmdt) Image representation

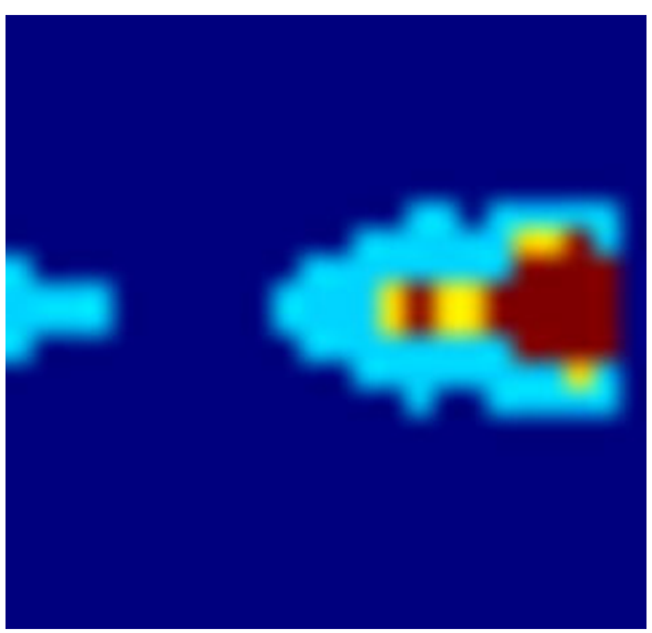
Mahabal et al., 2017

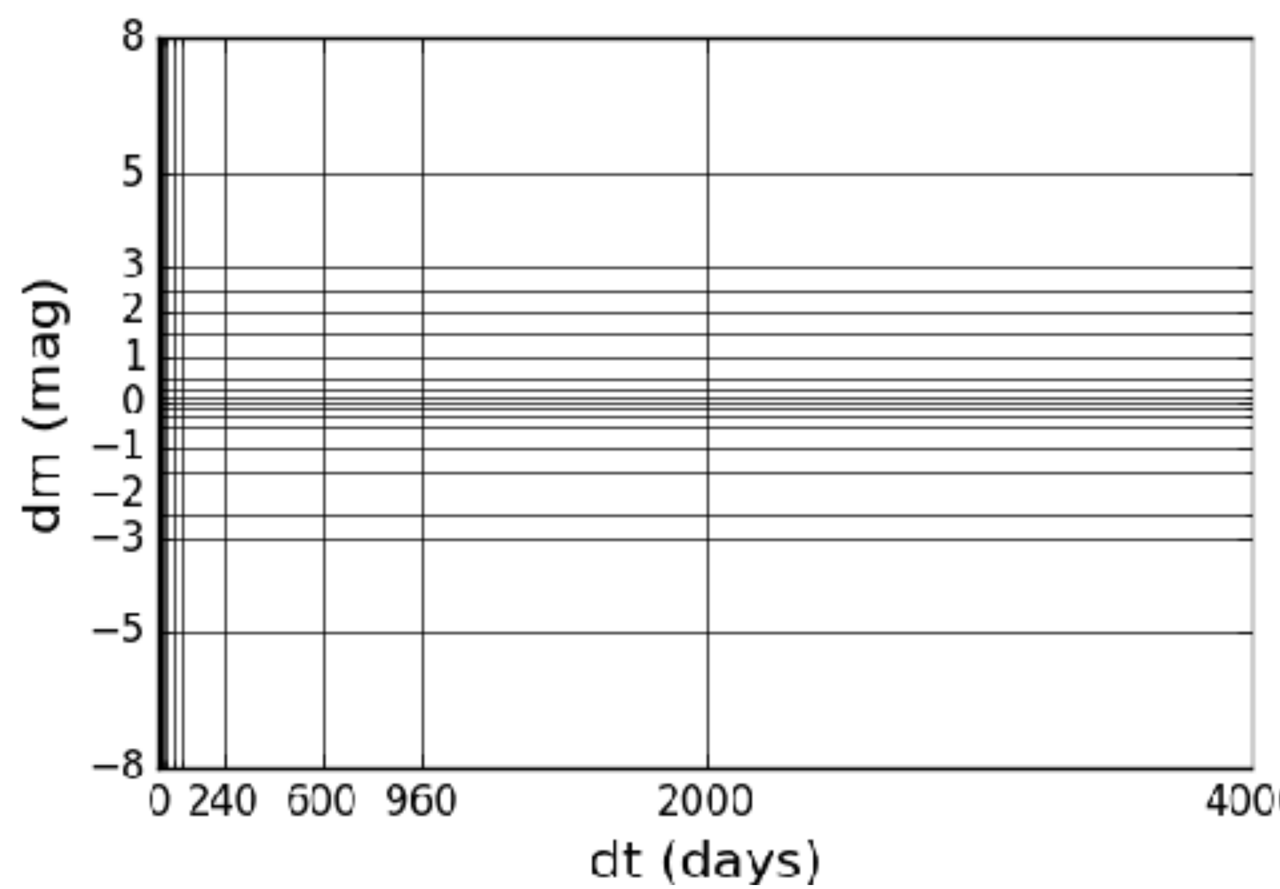


light curve with n points

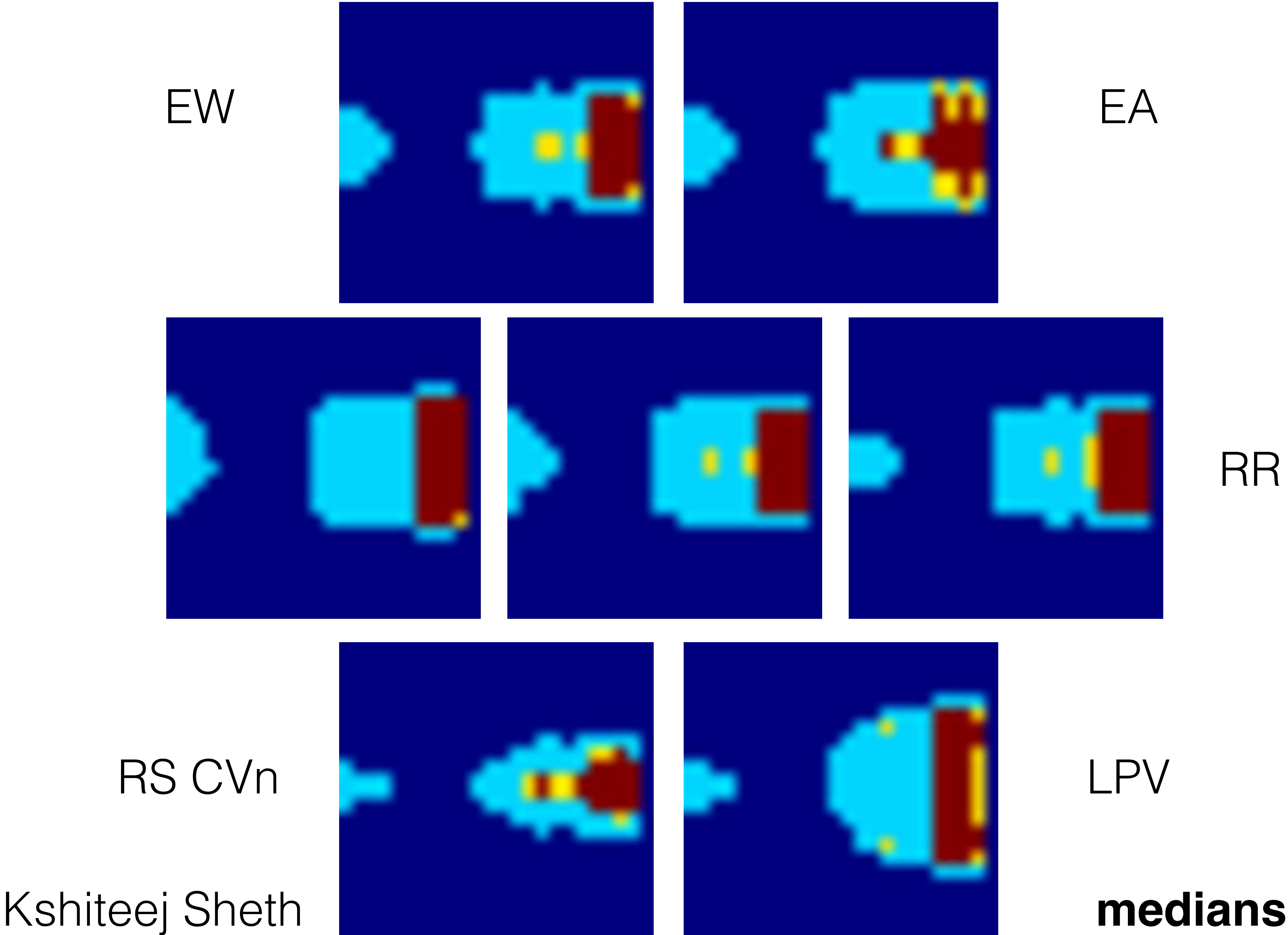
23 x 24 output grid

n * (n-1)/2 points

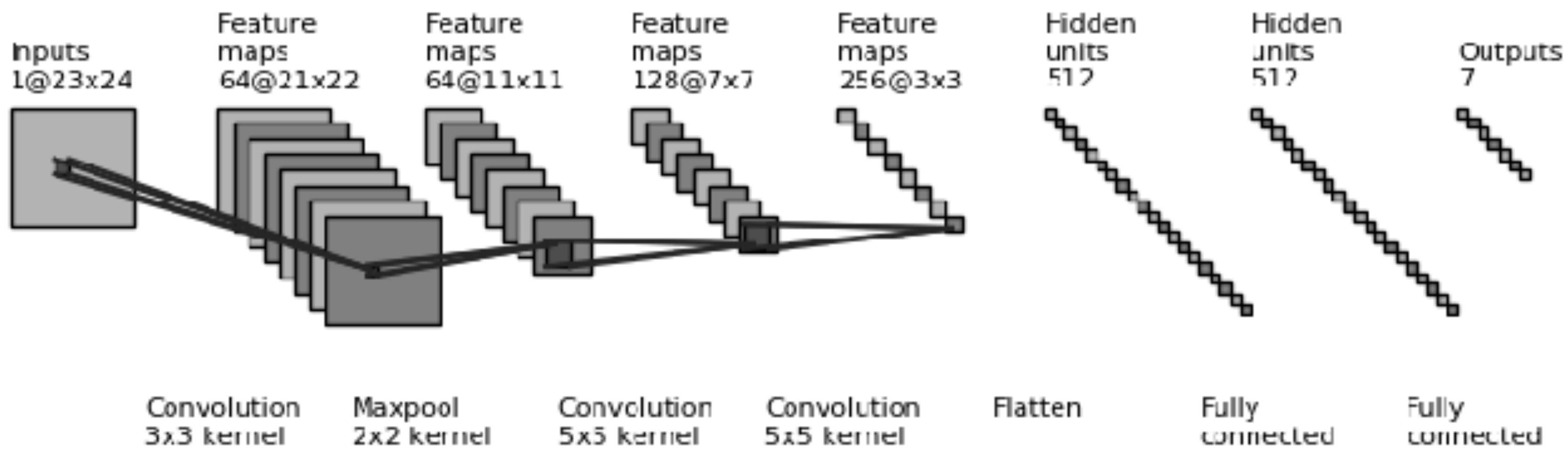




Area equalized pixels


EW
EA
RR
RS CVn
LPV
Kshiteej Sheth
medians

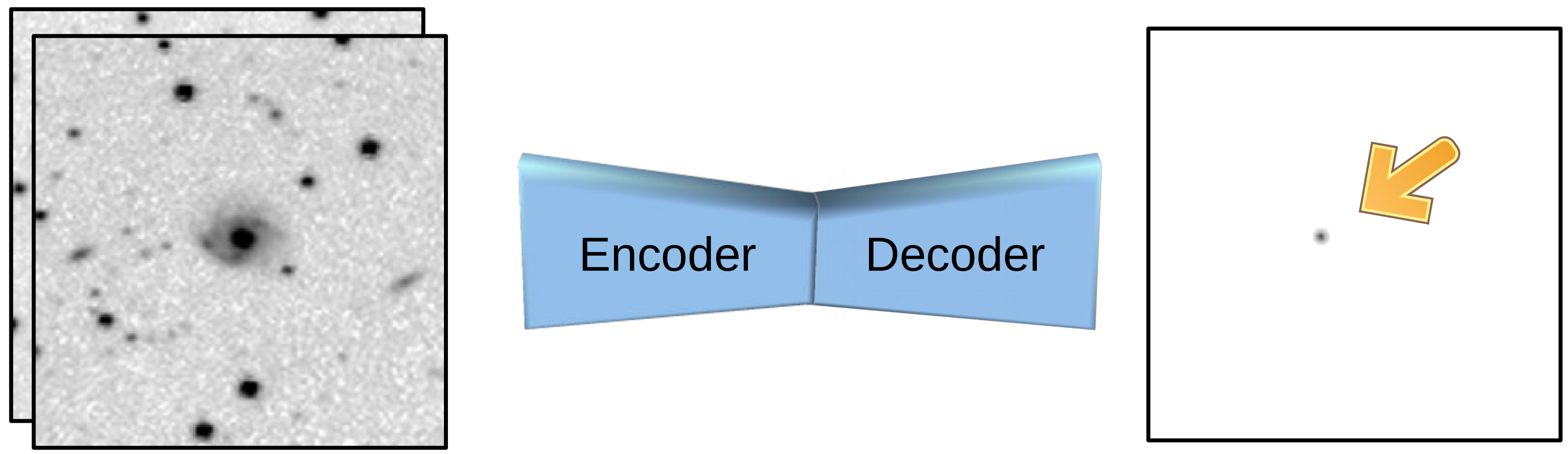
Network architecture

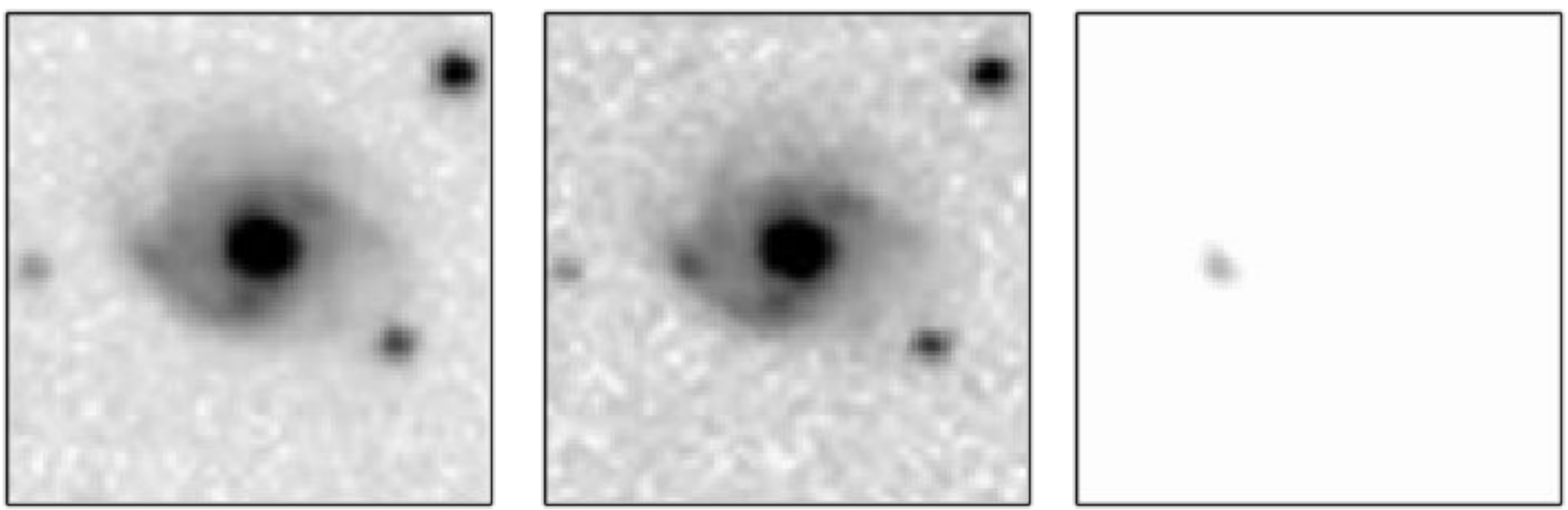


In this case shallow works well too

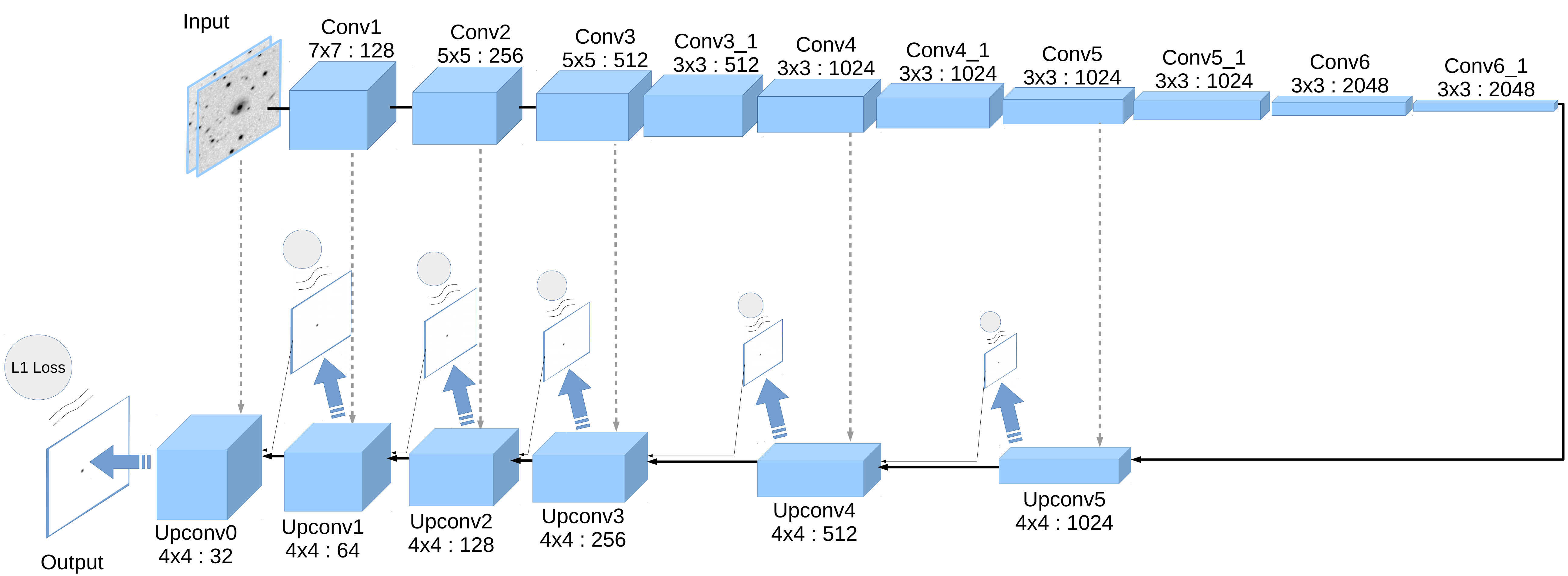
```
layers = [
    InputLayer,
    Conv2DLayer(32, size:3x3, rectify),
    DropoutLayer(0.1),
    DenseLayer(128),
    DropoutLayer(0.25),
    DenseLayer(128),
    DenseLayer(all, softmax),
```

Image subtraction for hunting transients without subtraction



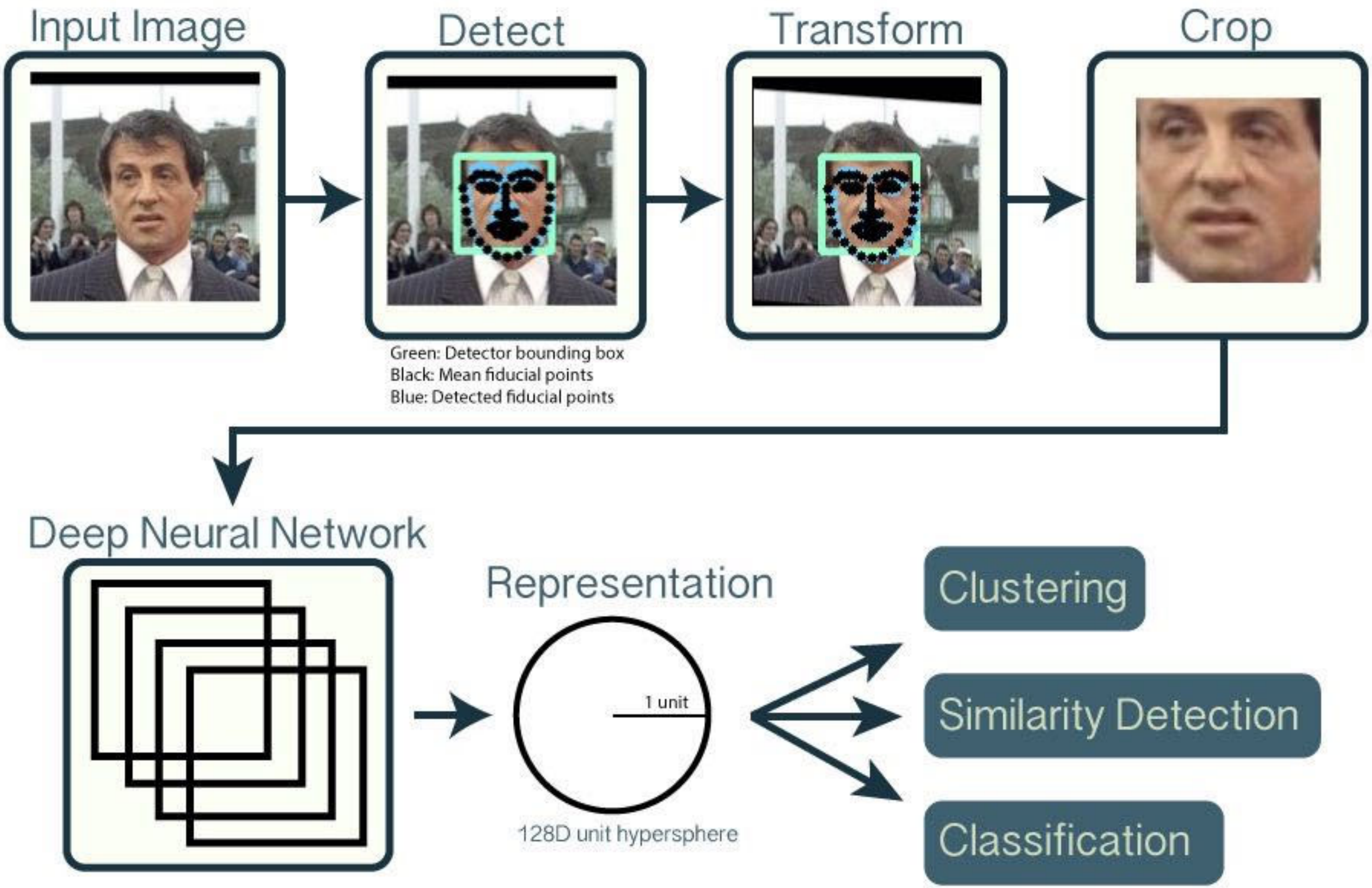


Sedaghat and Mahabal, 2017



Input
Conv1
7x7 : 128
Conv2
5x5 : 256
Conv3
5x5 : 512
Conv3_1
3x3 : 512
Conv4
3x3 : 1024
Conv4_1
3x3 : 1024
Conv5
3x3 : 1024
Conv5_1
3x3 : 1024
Conv6
3x3 : 2048
Conv6_1
3x3 : 2048
L1 Loss
Output
Upconv0
4x4 : 32
Upconv1
4x4 : 64
Upconv2
4x4 : 128
Upconv3
4x4 : 256
Upconv4
4x4 : 512
Upconv5
4x4 : 1024

OpenFace

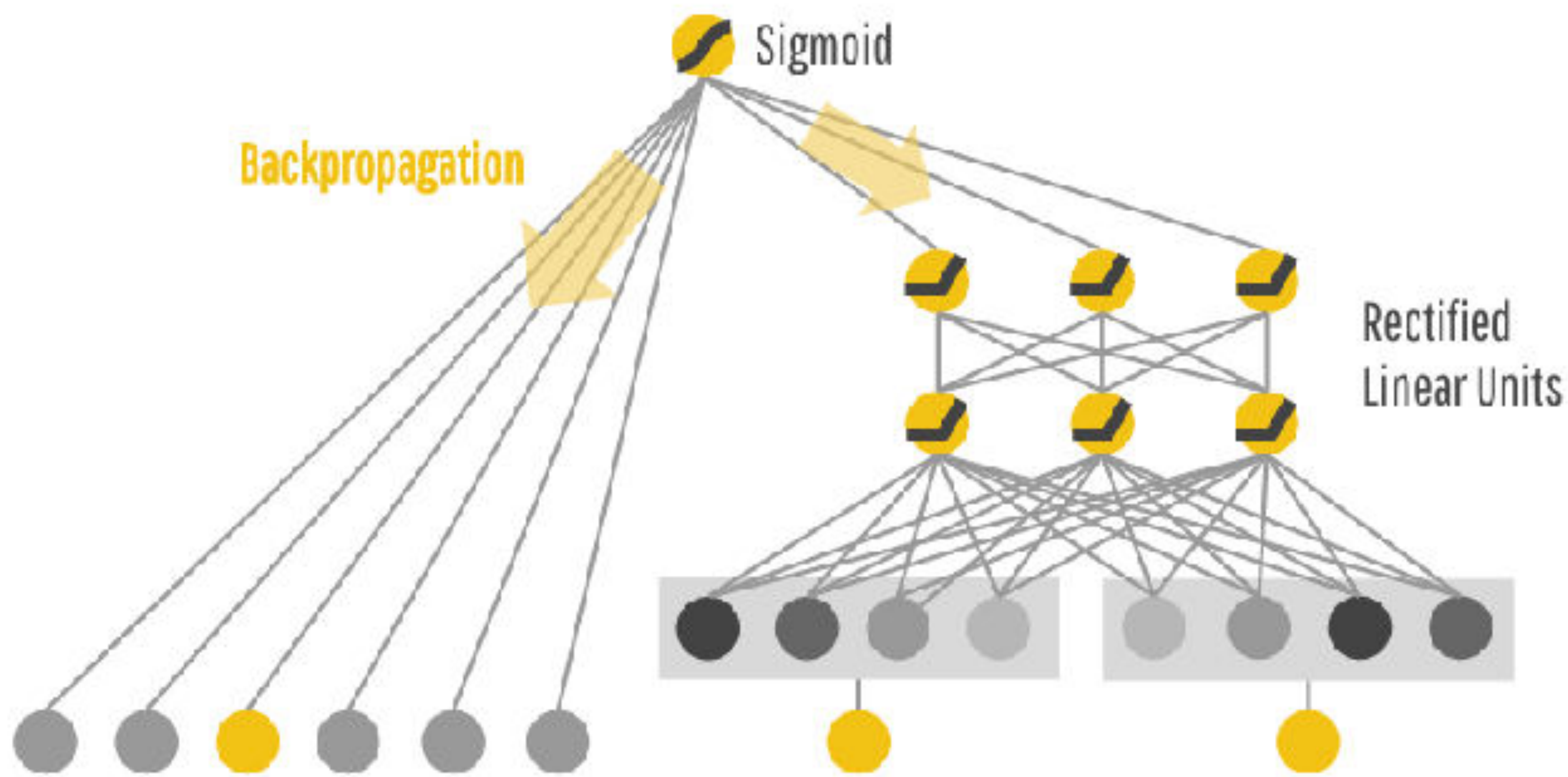


https://cmusatyalab.github.io/openface/

Another use for the face API

Faces masked in Streetview

Combining with unstructured data



Feature values and ancillary data

Image data (multiple wavelengths)

The “comments” or metadata become additional features (GoogLeNet)

https://research.googleblog.com/2016/06/wide-deep-learning-better-together-with.html

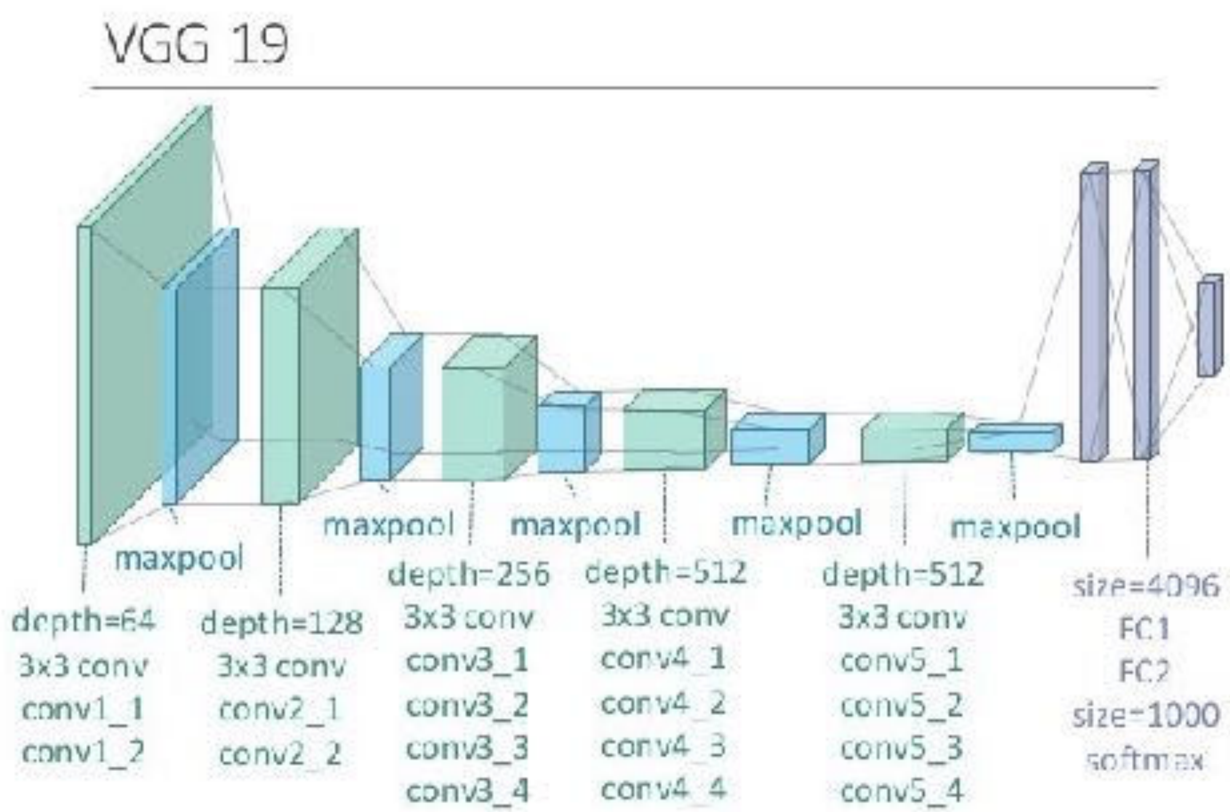
In <your area of interest> what can you apply deep learning to?

- One speculative example
- One more directly related to your work

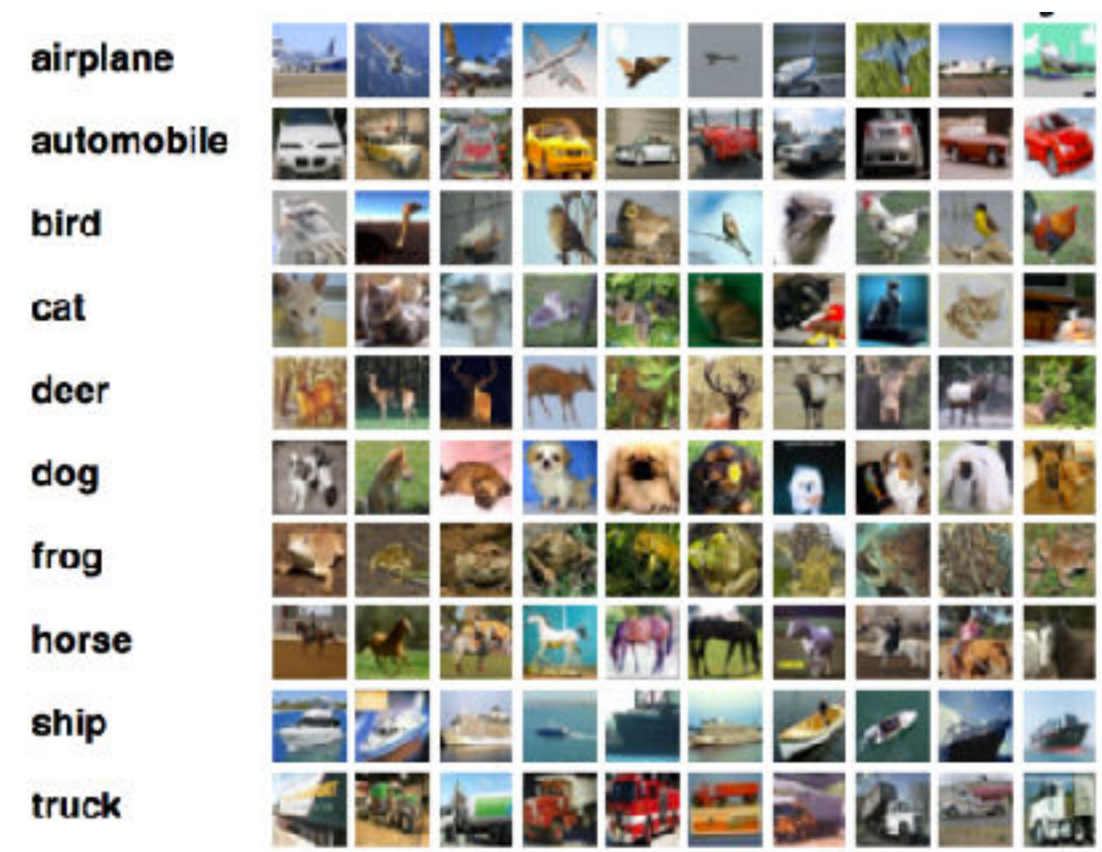
Hands on (TF+Keras)

- Open the available notebook (simple_cnn.ipynb)
- Create a CNN with the architecture:
    - CONV Layer (1x3x3)
    - MAX Pooling
    - RELU

Tutorial 1 (TF+Keras)



Network architecture



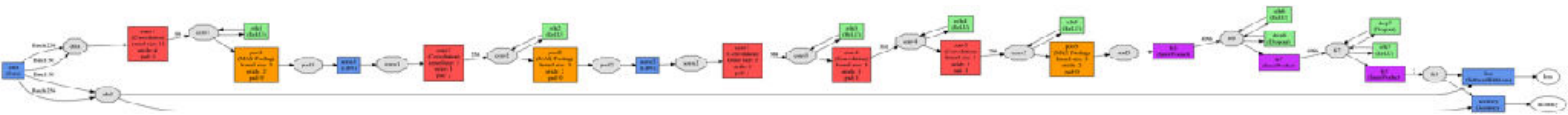
CIFAR-10 (6 10^4)

- Use VGG19 to classify CIFAR-10 database.

Useful links:

https://github.com/deep-diver/CIFAR10-VGG19-Tensorflow

Tutorial 2 (Caffe)



- Use AlexNet trained for ImageNet
- To classify Cats Vs Dogs

Kaggle Cats Vs dogs, 25.000

Useful links:

https://prateekvjoshi.com/2016/01/05/how-to-install-caffe-on-ubuntu/

https://prateekvjoshi.com/2016/02/02/deep-learning-with-caffe-in-python-part-i-defining-a-layer/

http://adilmoujahid.com/posts/2016/06/introduction-deep-learning-python-caffe/

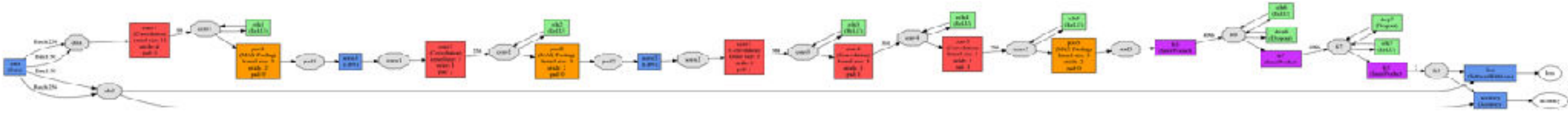
Tutorial 2 (Caffe)

25.000 examples

- Designing layers [demo]
- Pre-processing :

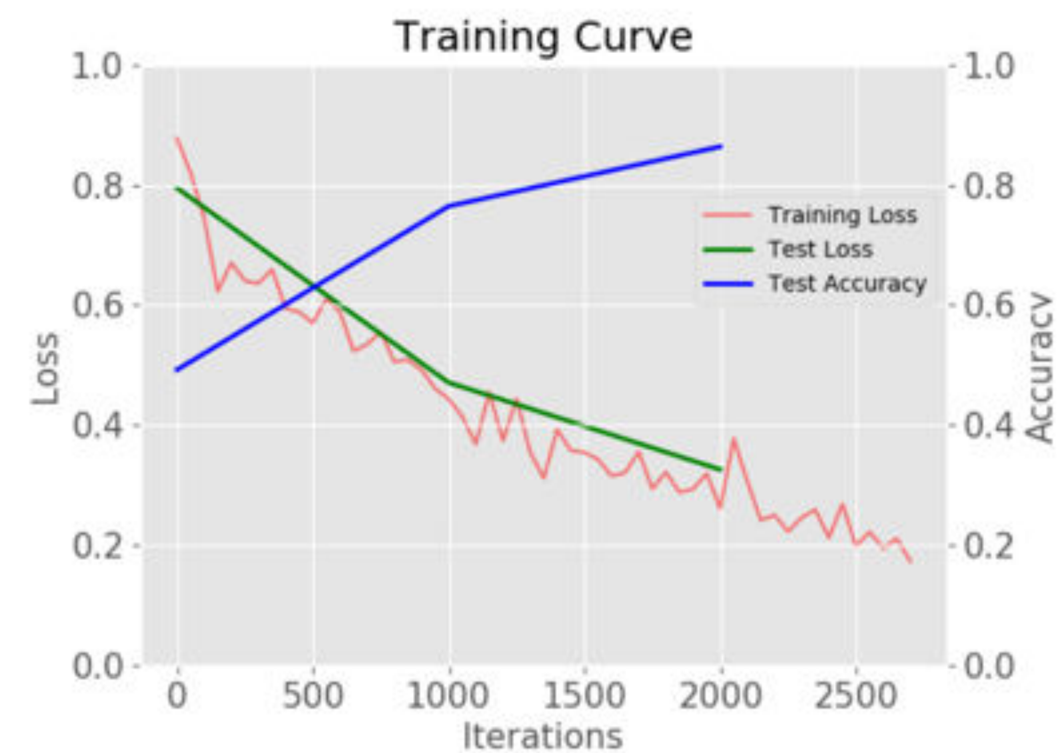
Example of image transformations applied to one training image

- Convolutional network architecture (AlexNet):

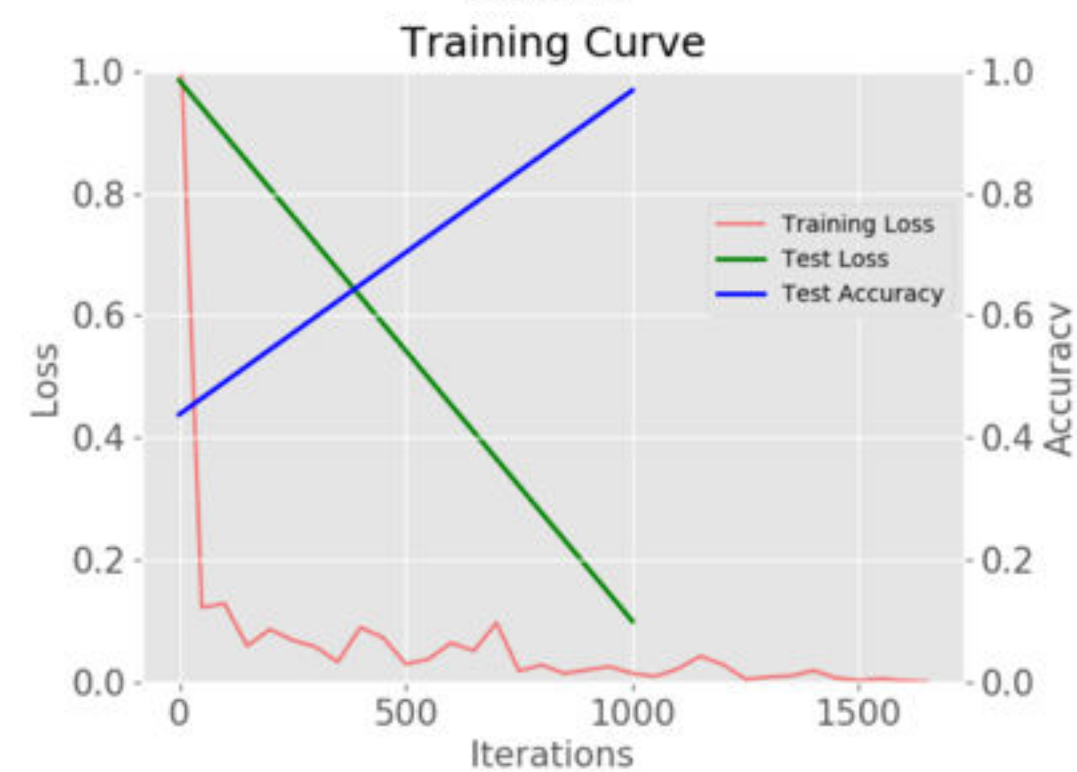


Tutorial 2 (Caffe)

- Training and test results (traditional, 2 days):



- Training and test results (transfer learning, hours):



Model zoo:
https://github.com/BVLC/caffe/wiki/Model-Zoo

Demo ConvNet

Online deep learning! [demo using AlexNet*]

http://demo.caffe.berkeleyvision.org/

ILSVRC 2012 (ImageNet, 10^7 examples, 10^4 categories)

Summary

- CNNs are taking over, especially the image domain
- Can come up with features not thought of before
- Abstracted libraries and visualizations available
- Over-learning can be a problem:
  - augmentation
  - adversarial examples/generative networks
- Should ensure they do not become convoluted
- Deep and wide networks may prove to be a boon