Deep Learning for computer vision

2019-2020

David Filliat

Unité Informatique et Ingénierie des Systèmes

École Nationale Supérieure
de **Techniques Avancées**

ENSTA

IP PARIS

Objectives

Todays program

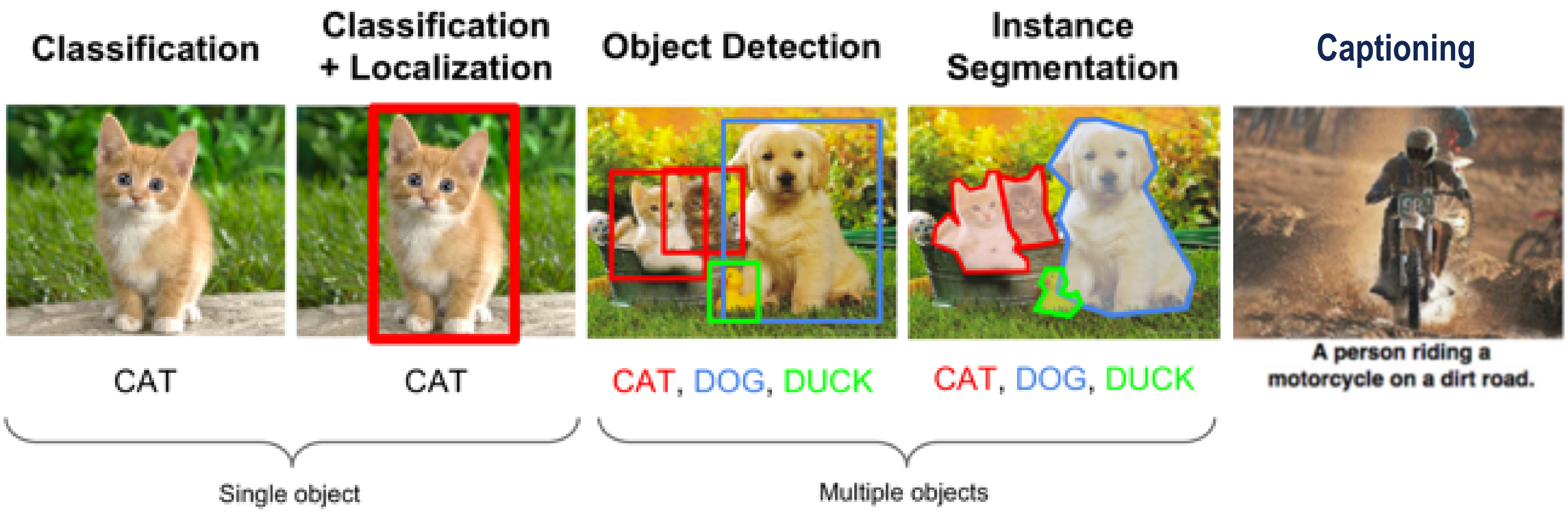
- Recall on Computer vision / Machine learning basics
- Introduction to deep learning
- Convolutional Neural Networks
- Applications of CNN in computer vision

Practical

- CNN for image classification

Machine learning / computer vision basics

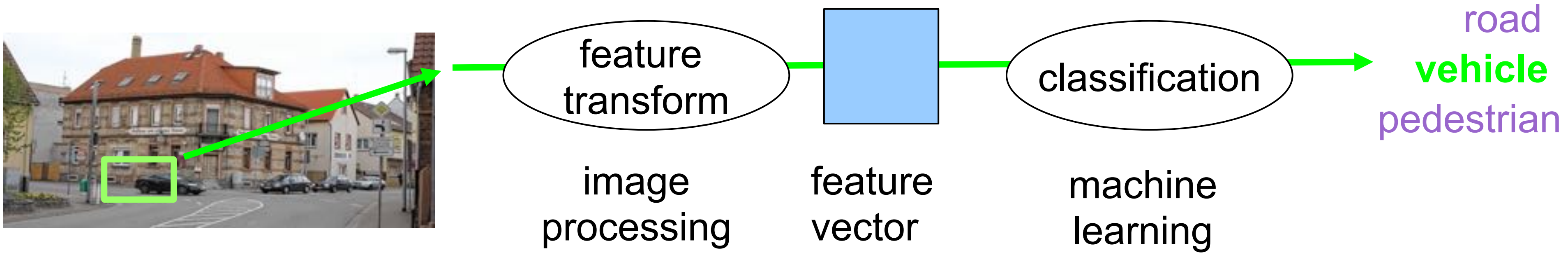
Computer Vision Tasks



Requires Classification

Recognition

Typical recognition architecture



Standard procedure

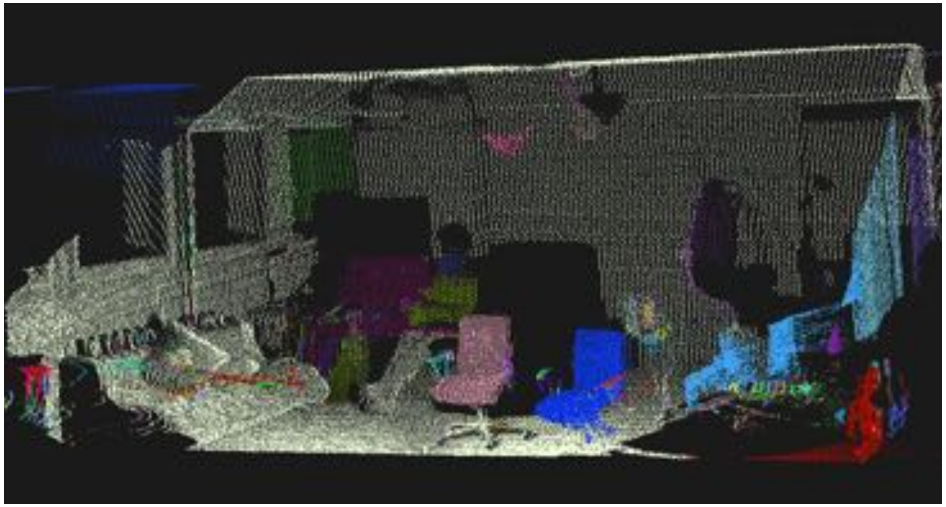
- **Feature transform**: problem-dependent, hand-crafted, transforms image into a form useful for classification
- **Classification**: generic, trained, takes feature vector and produces decision

Detection

Detection = localization + recognition

How to choose elements to recognize ?

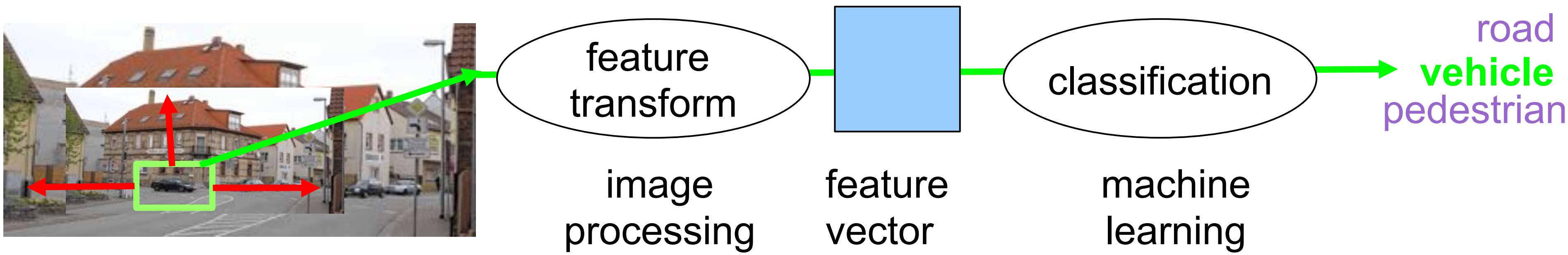
- In robotics, often use 3D data which simplifies object segmentation




- **Segmentation** : localization of objects irrespective of identity
- Based on environment hypothesis. e.g., objects on planes
- As difficult as recognition in general

Detection by recognition

Efficient recognition makes localization possible



Sliding Window approach

- Slide window over whole image
- detections: positive binary classification results
- for larger objects: repeat after shrinking image
- Can be very efficient when exploiting windows overlap
- **Warning** : need very good recognition:

  if 10 000 windows/image; 0.1 % error -> 10 errors/image !

Detection by recognition

Region proposal approach

- Generate likely object bounding boxes

E.g. : selective search

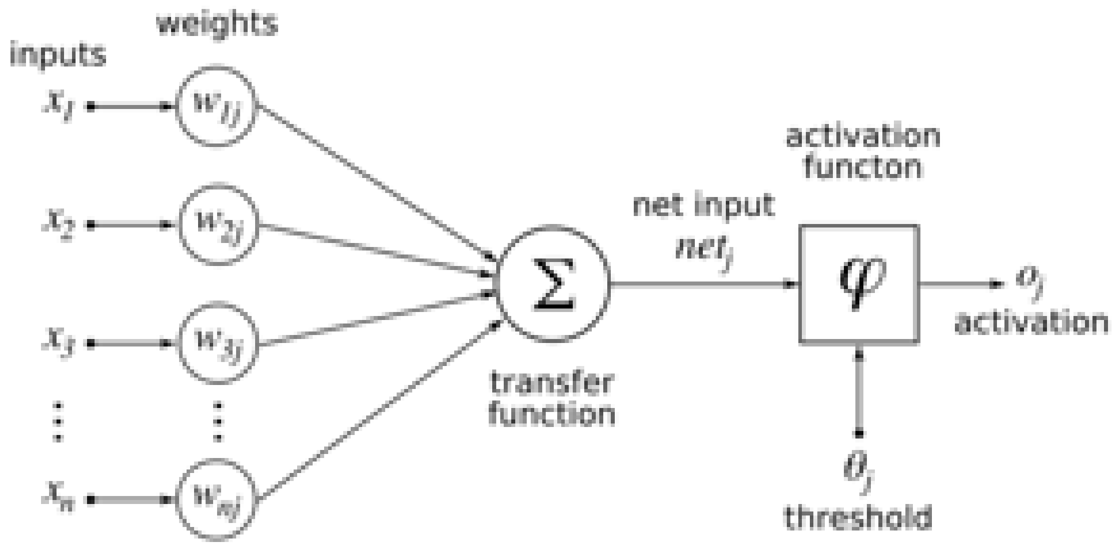
- Segment images using multiple color spaces
- Hierarchically group regions and process bounding boxes

Bottom-up segmentation, merging regions at multiple scales

Neural Networks

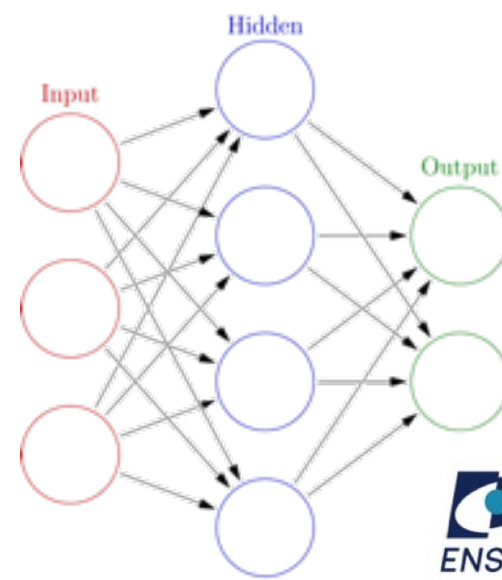
Artificial neuron ~1950

- Element performing sum of weighted input + non linear fct



Neural network (Perceptron, (Rosenblatt, 58))

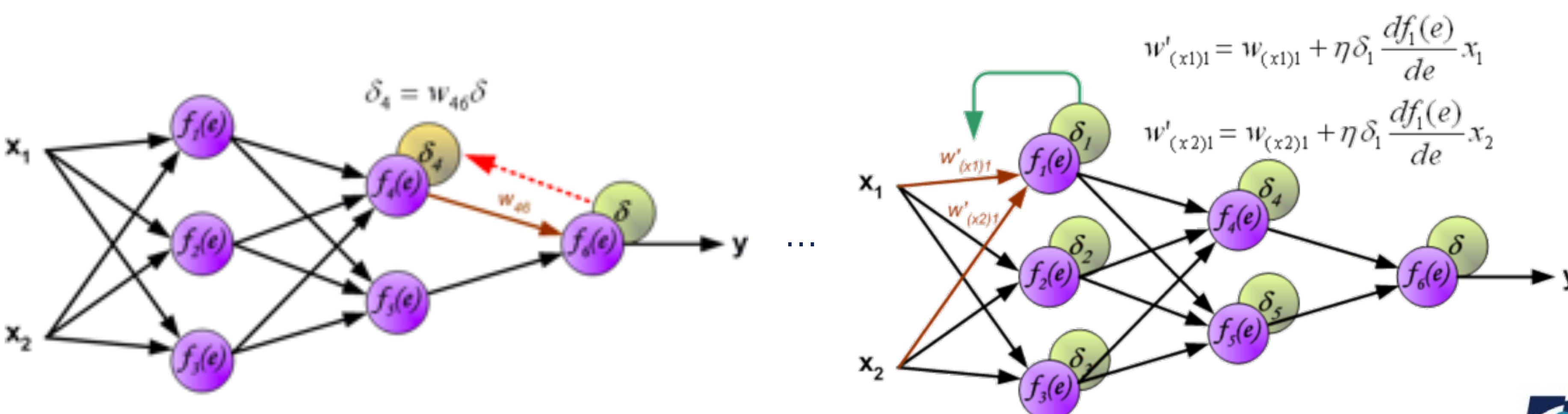
- Assembly of neurons, often organized in layers
- Parameterized by all connection weights $\mathbf{w}_{ij}$



Neural Networks

Learning in neural networks

- Find weights w_{ij} that minimize prediction error
- Backpropagation of error with gradient descent (Werbos, 75)
- Compute: error of output, gradient wrt. weights; update weight following gradient
- Do the same thing for previous layers using ‘chain rule’



Mariusz Bernacki - http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html

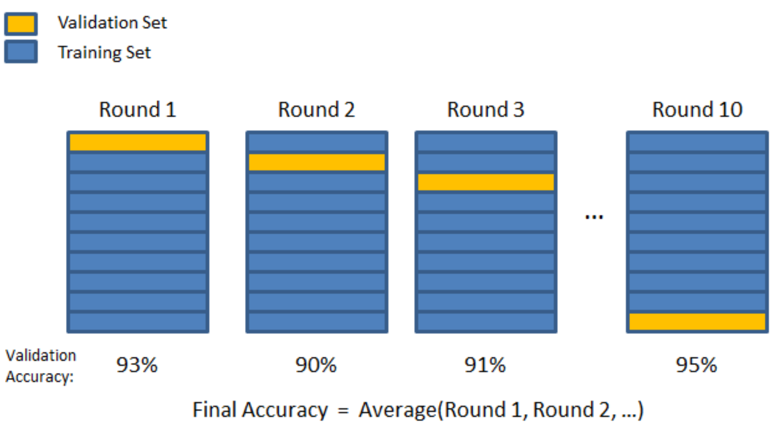
Training procedure

Data sets

- If possible, make 3 sets : training, validation, test
- Use Training for training …
- Use Validation to check training quality, tune algorithm params
- Use test only to report final performance (hidden in ML competitions)

K-fold Cross validation

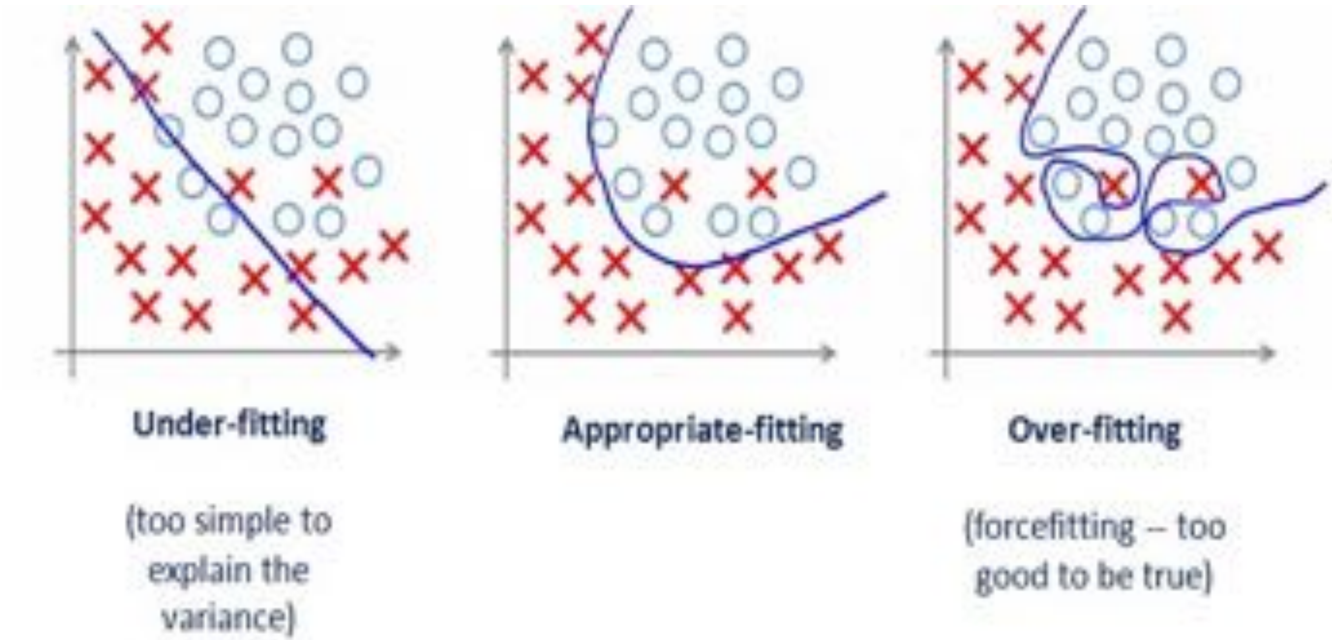
- When little data : split dataset in k sets
- Train on k-1, validate on remaning one
- Repeat k times
- Report mean performances



Training procedure

Overfitting

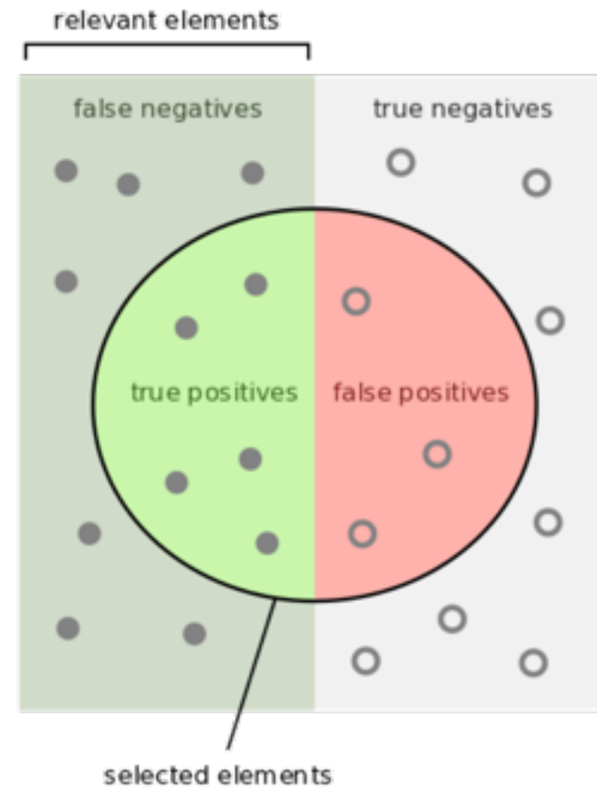
- Training too much on training set limits generalization
- Important to keep an eye on validation error
- Trick (early stopping) : Stop learning if validation error increase

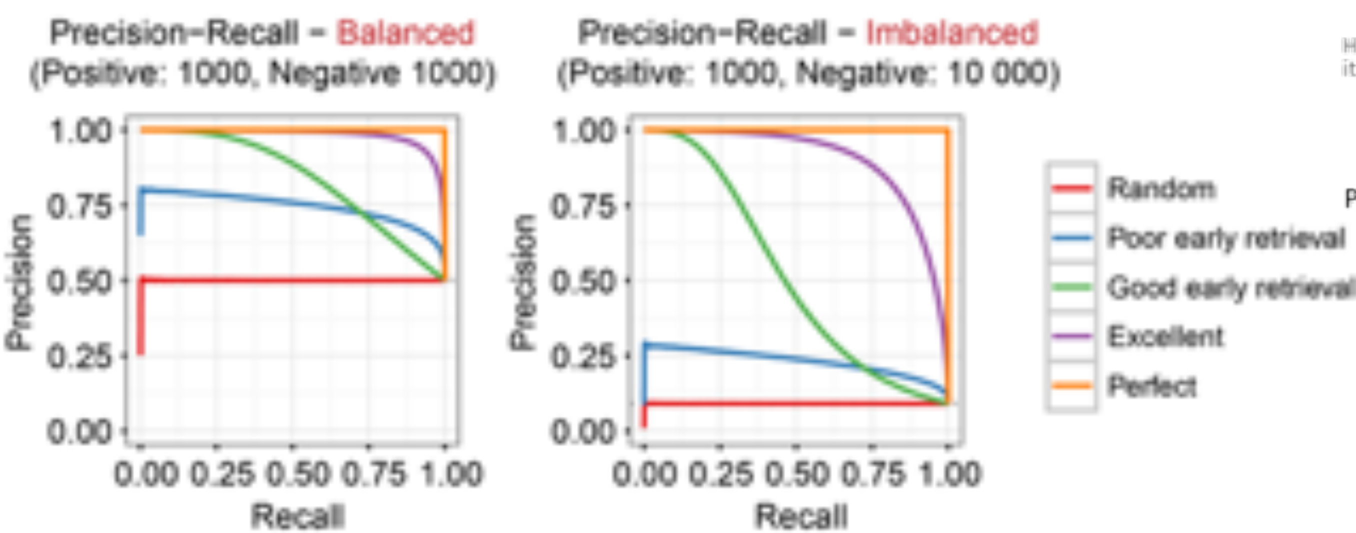


Reporting performances

Detection performance

- A number is not sufficient
- Report curve showing performance tradeoff
- Ex : precision/recall curves
  - Precision : correct detection / nb of detection
  - Recall : correct detection / nb of true elements

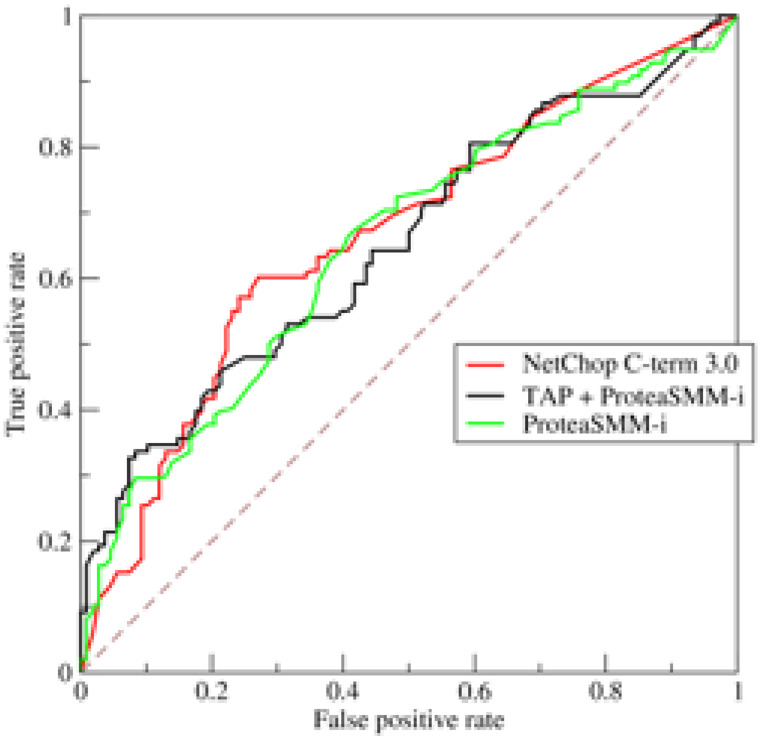


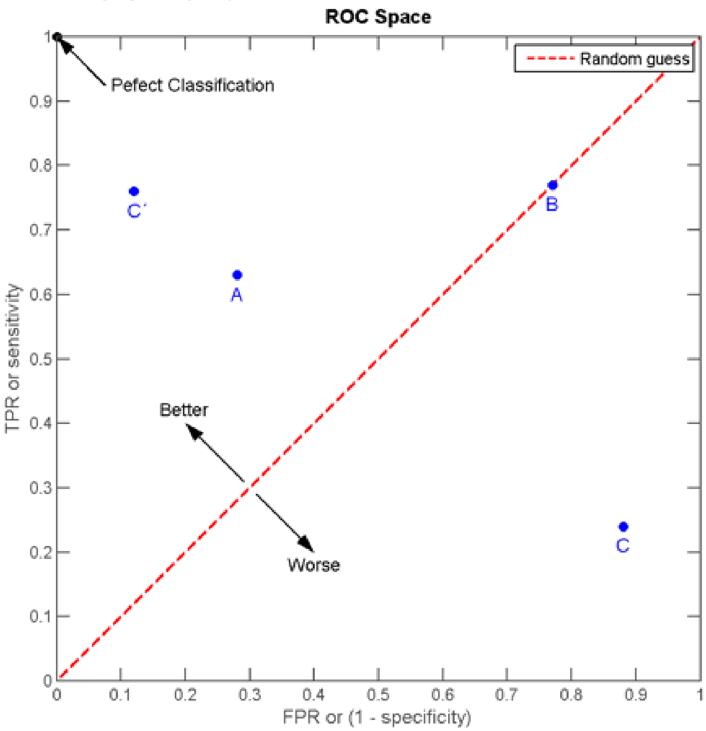


Reporting performances

ROC curves

- Receiver Operating Characteristic
- Plot true positive rate (= recall) / false positive rate (FP/real negative)
- Draw curve by varying detection threshold





Introduction to Deep Learning

Deep Learning

Return of the neural networks

- Around 2006 ?
- Neural networks with “many” layers
- Theory *very* similar to perceptrons (for most models)

Why “Deep” ?

- Approximate more complex functions
- Works well in practice (on many problems)

Why now ?

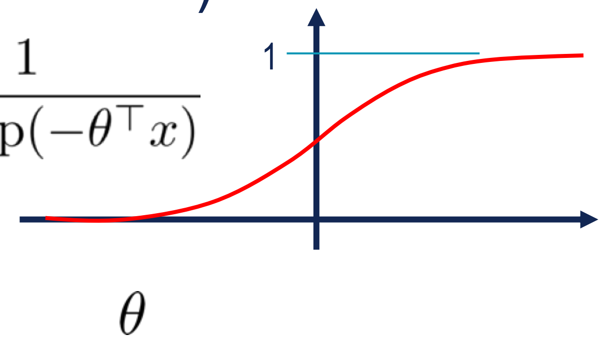
- More processing power
- Found solutions to some learning problems
- Availability of large datasets

Deep Learning

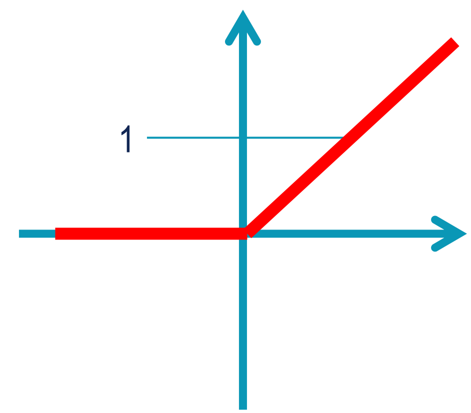
Training with back-propagation (supervised version)

- Dates back to Werbos (75)
- But did not work on “deep” networks
  - Many local minima in cost function
  - Vanishing/exploding gradient in the deep layers
  - Hard to debug/understand

$$\frac{1}{1+\exp(-\theta^\top x)}$$



ReLU(z) = max{0, z}



“Rectified Linear Unit”
→ Increasingly popular.

[Nair & Hinton, 2010]

What’s new ?

- Choice on activation function (instead of sigmoid)
  - Tanh, ReLU
- Initialization : unsupervised pre-training
  - Train each layer by reconstructing input
  - Provides good starting point toward global minima

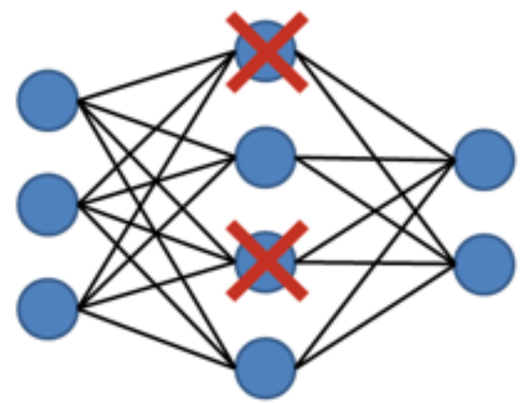
Deep Learning

What's new ?

- Choice of parameters : gradient step size, momentum
  - Avoids too strong modifications

$$v := \mu v + \epsilon_t \nabla_\theta \mathcal{L}(\theta)$$
$$\theta := \theta + v$$

- Dropout
  - Train while removing random connections
  - Force robustness to noise



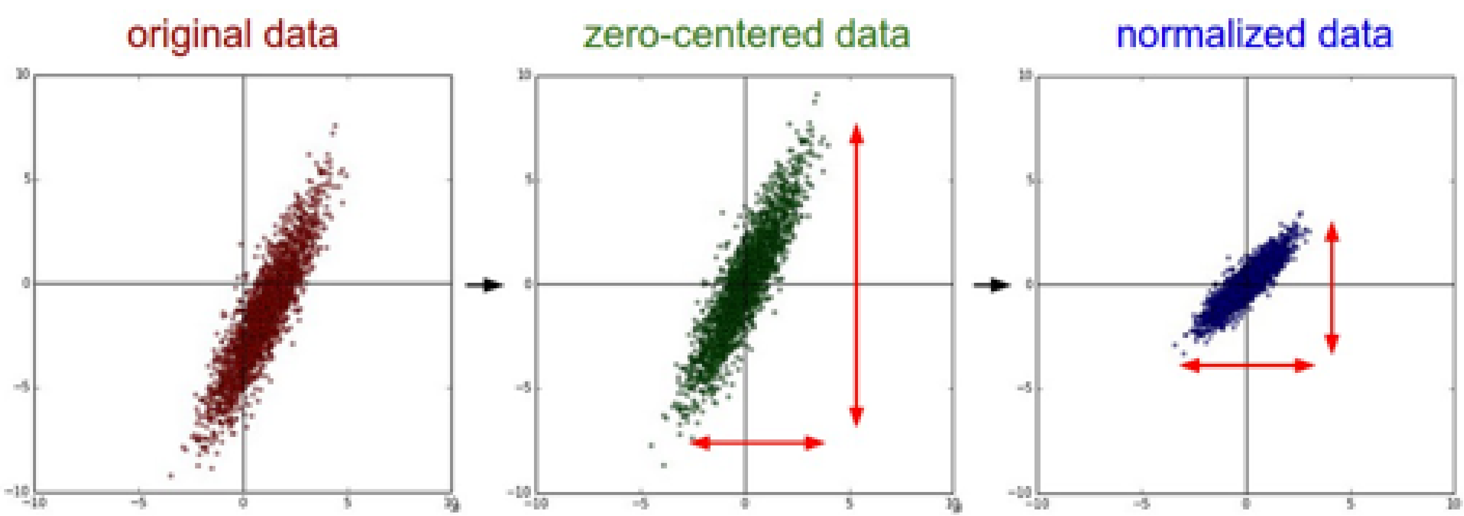
- Batch normalization
  - Normalize data at each layer, for each batch
  - Regularize gradient -> solves most of the problems (no need for pretraining, dropout)

Bengio, 2012: "Practical Recommendations for Gradient-Based Training of Deep Architectures"
Hinton, 2010: "A Practical Guide to Training Restricted Boltzmann Machines"

ENSTA IP PARIS

Training procedure (1/3)

- Use standard training / validation / test sets
- Normalize data
  - Substract mean (computed on training set)
  - Divide by std. dev. (computed on training set)



Training procedure (2/3)

- Choose a Loss function
  - For example for softmax classification, use cross entropy:

$$\hat{y}_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \qquad L(\hat{y}, y) = -\sum_j y_j \log \hat{y}_j$$

z_i: network output; $\hat{y}_i$: estimated prob of class i; y_i: true prob of class i;

- Initialize weights randomly around 0
  - E.g., Gaussian noise: N(0,ε)
- Use one variant of gradient descent (with momentum, ADAM, …)
  - Compute (automatically) gradient to reduce the loss

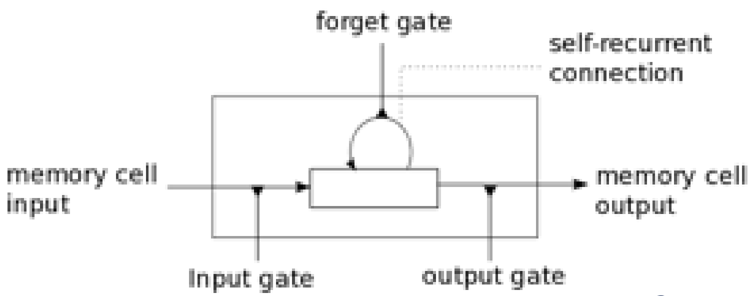
Training procedure (3/3)

- Use mini-batches
  - Compute gradient on a small set of examples
  - Take average (sum), perform one step of gradient descent with this value
- Interest of mini batches
  - Smooth gradient noise -> allow larger steps -> learn faster
  - But too large mini-batches lead to problems (stuck in local min…)
  - Linked to memory size of GPUs
  - Sensitive parameters
- Define a schedule of decreasing learning rates

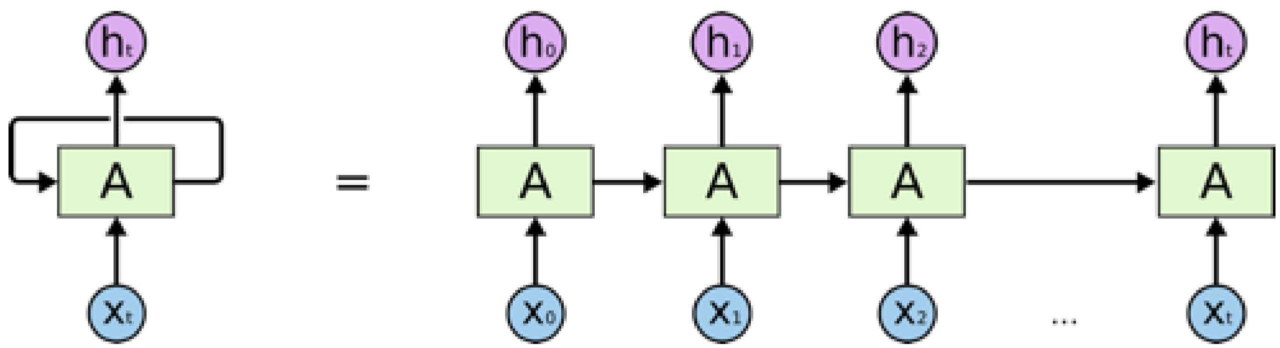
Deep Learning

Many architectures

- Convolutional Neural Networks
  - Specialized for image processing
  - See later
- Recurrent architecture (e.g. LSTM)
  - Processing of temporal data
  - Speech recognition, action recognition
  - Trained by unfolding + supervised learning



LSTM cell

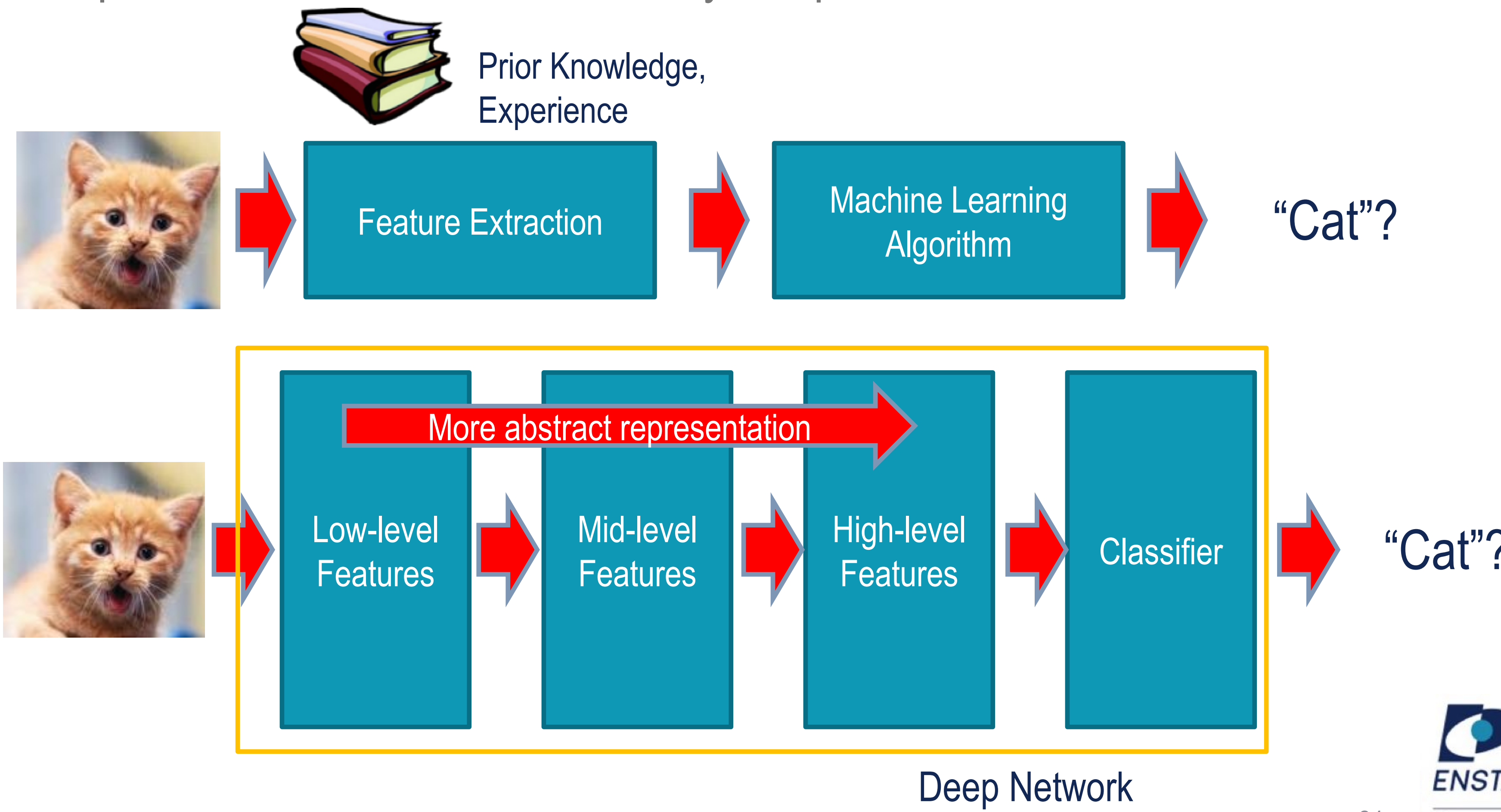


Deep Learning for vision

Deep learning for vision

Avoid manual feature construction

- Replace traditional architecture by deep network



Deep learning for vision

Avoid manual feature construction

- Process raw data directly
- Learn directly relevant feature from data
- Natural increase of feature abstraction
- Adapts to other modalities (depth, IR …)

Problems

- Large image size -> large networks
- Need lots of training data
- Need to reduce network parameters

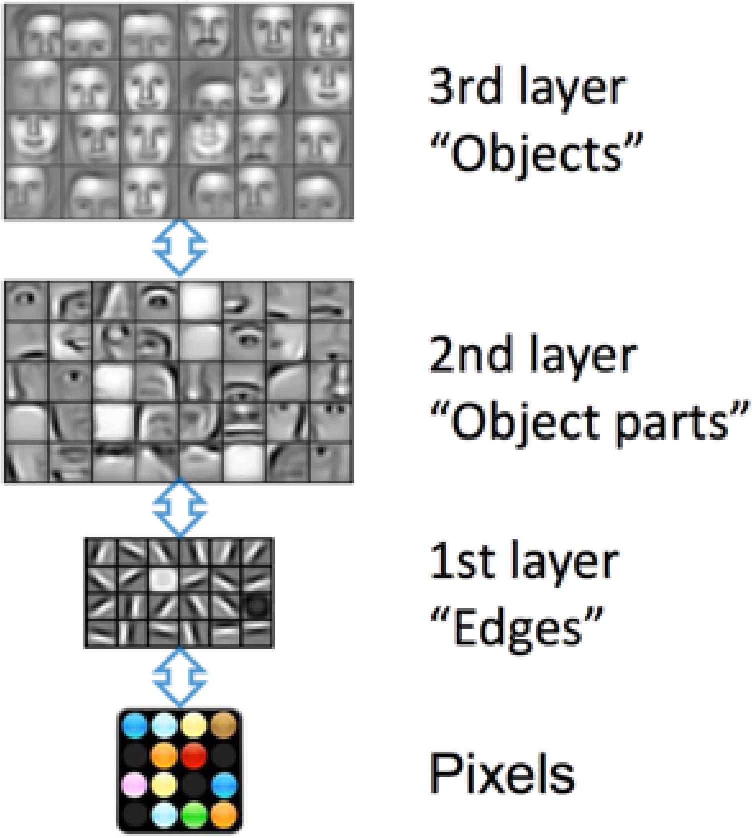
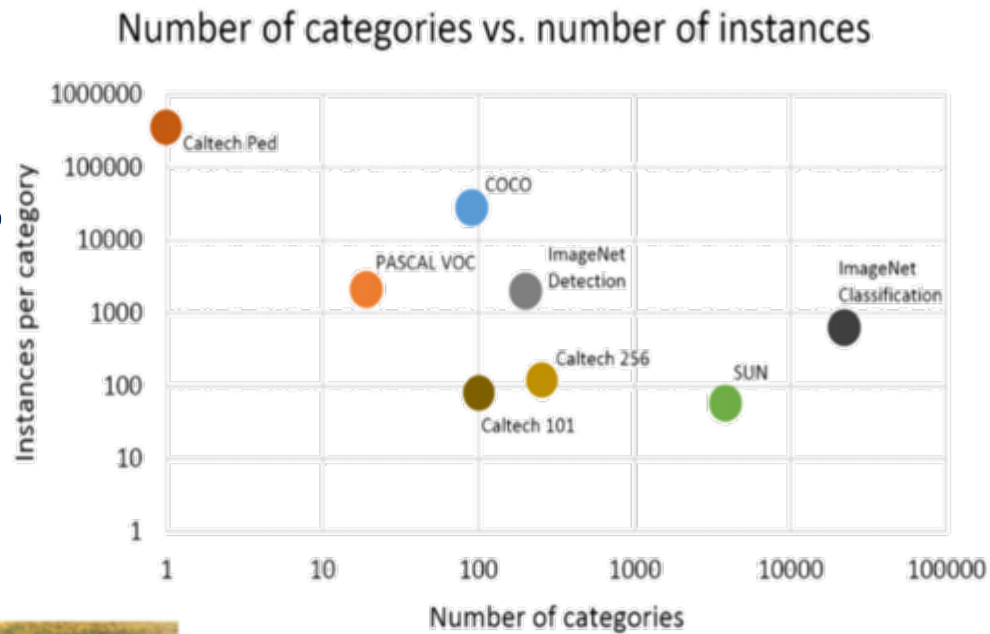
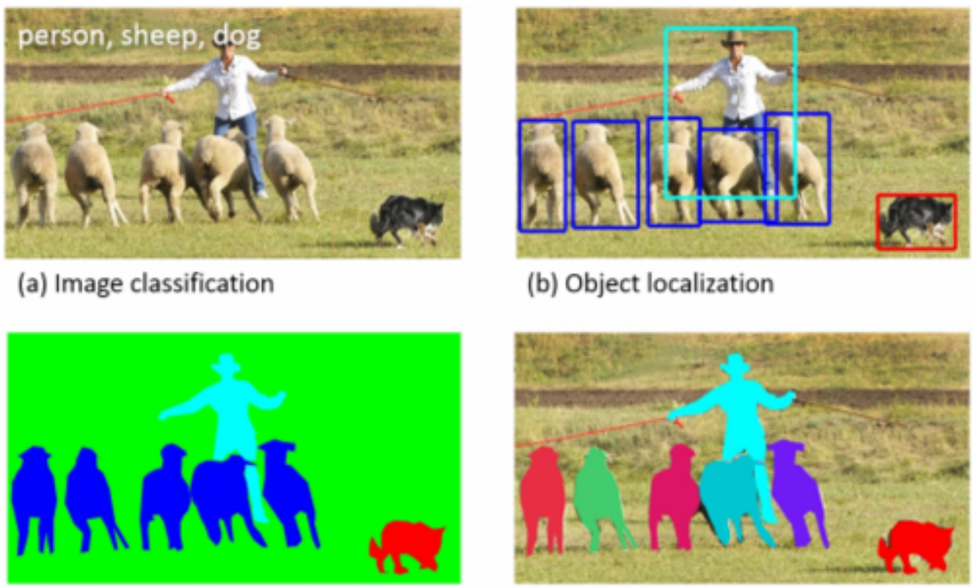


Image databases

Several large scale databases

- Ex : Microsoft COCO
  Common Objects in Context
- ImageNet …





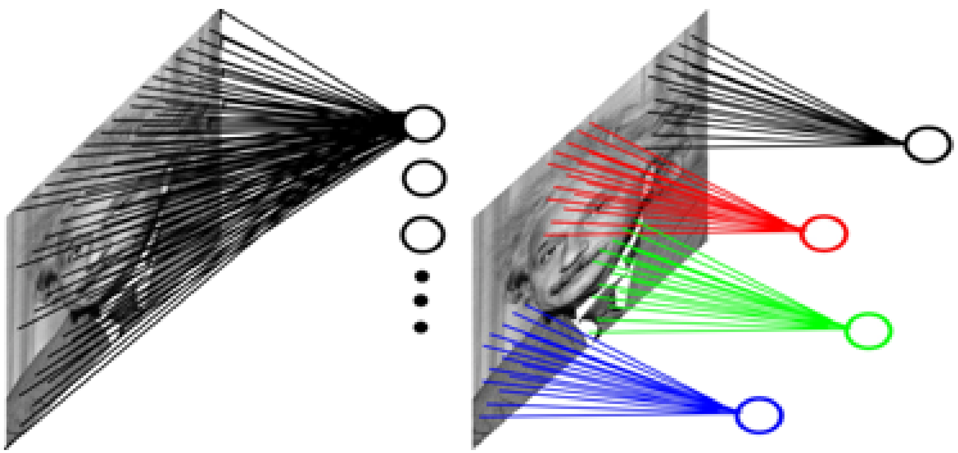
(a) Image classification (b) Object localization

(c) Semantic segmentation (d) This work

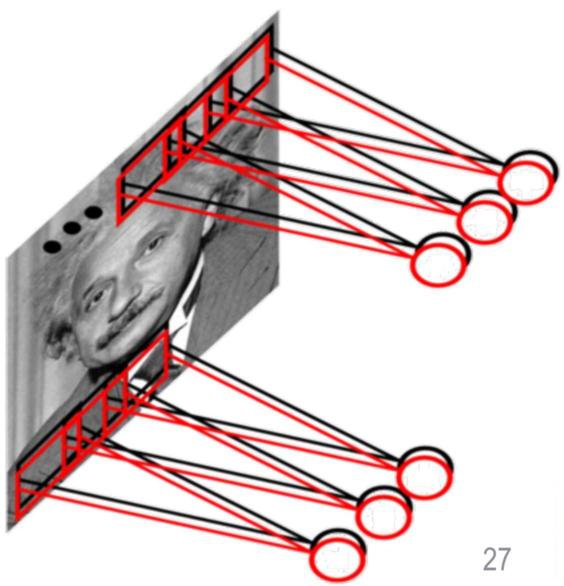
Convolutional Neural Networks

Reducing number of network parameters

- Exploiting image invariance to translation
- Use only limited local support
- Use same local weights for all positions -> convolution



- Use several convolutions
  at each position -> multiple features layers

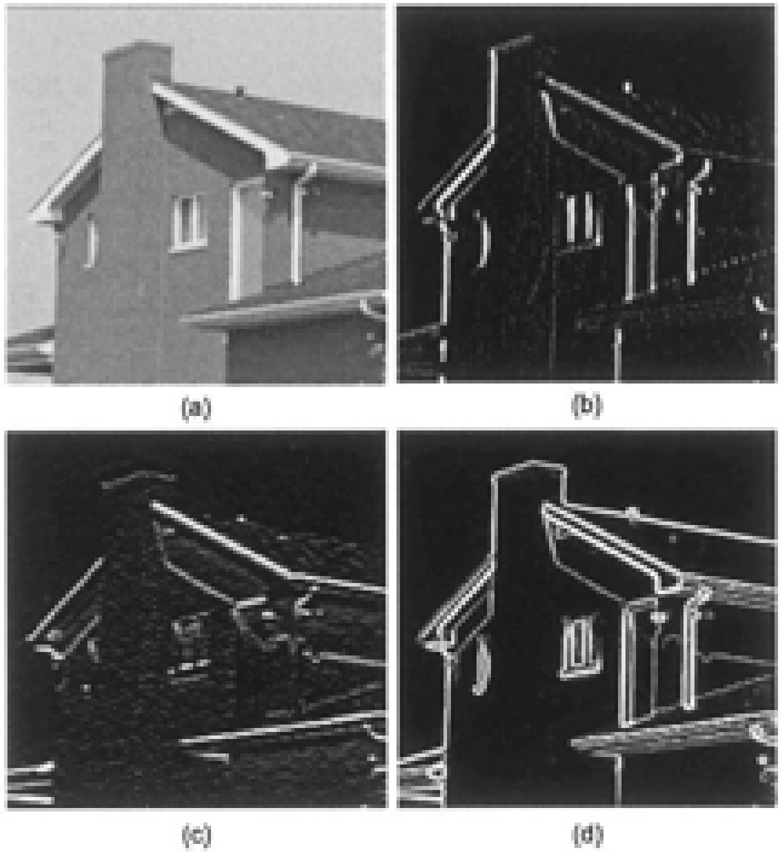


Convolution in image processing

Sobel edge detector

- Convolution with 'Hand made' filters

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$



$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

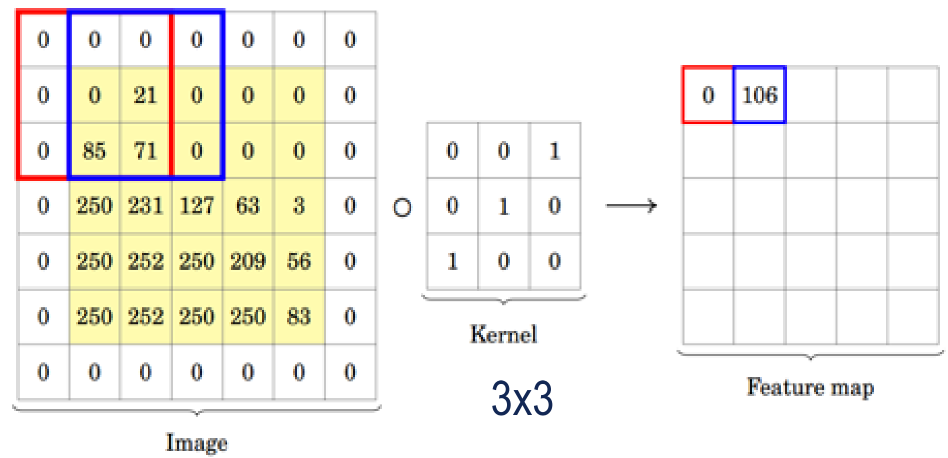
$$\sqrt{G_x^2 + G_y^2}$$

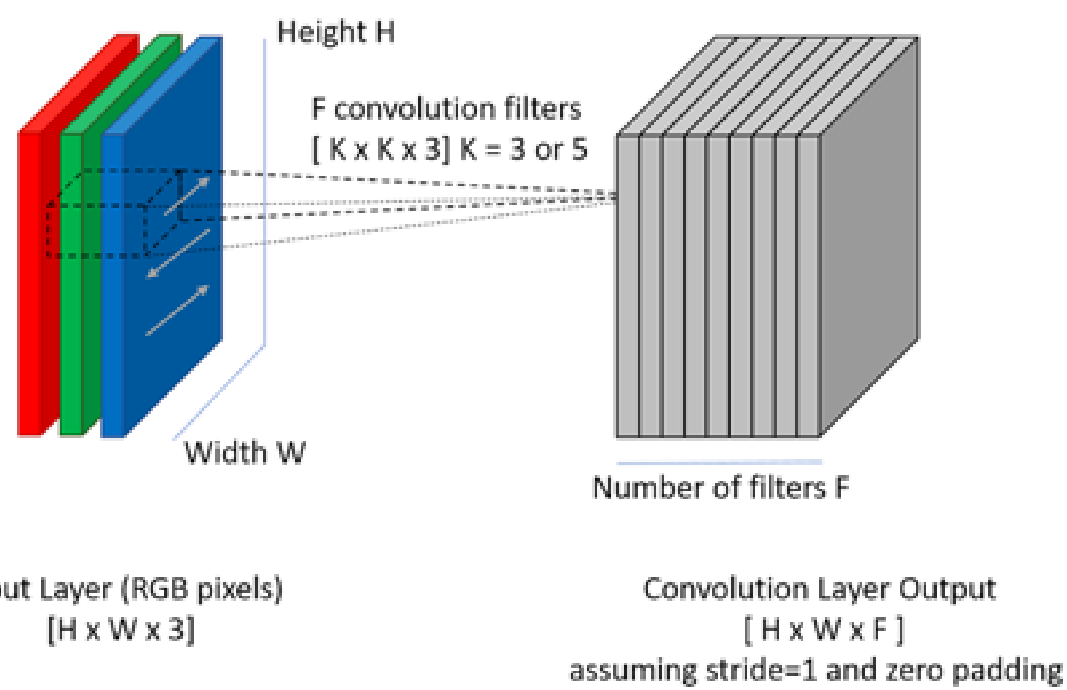
Convolutional Neural Networks

Convolution layer parameters

- Kernel size / padding
- Number of Input/output

feature maps

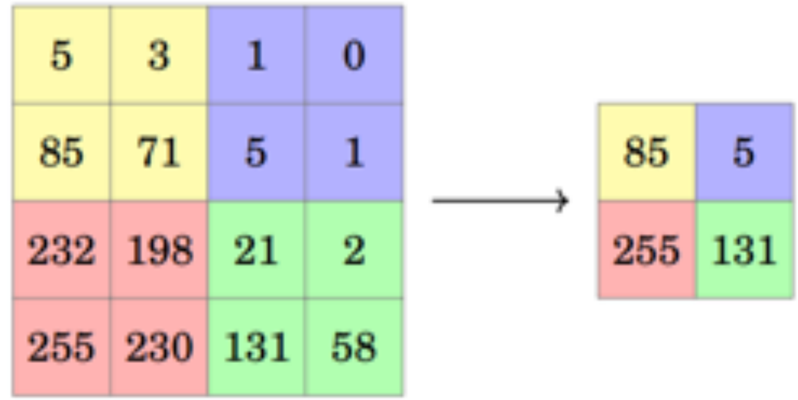




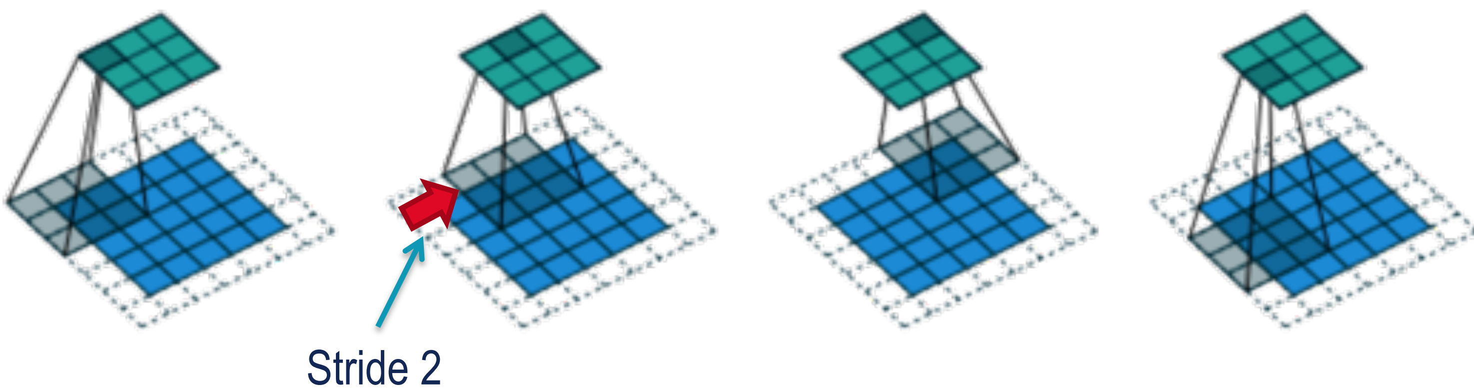
Convolutional Neural Networks

Reducing feature map size

- Pooling (**Max** / average)



- Stride



- Layer size

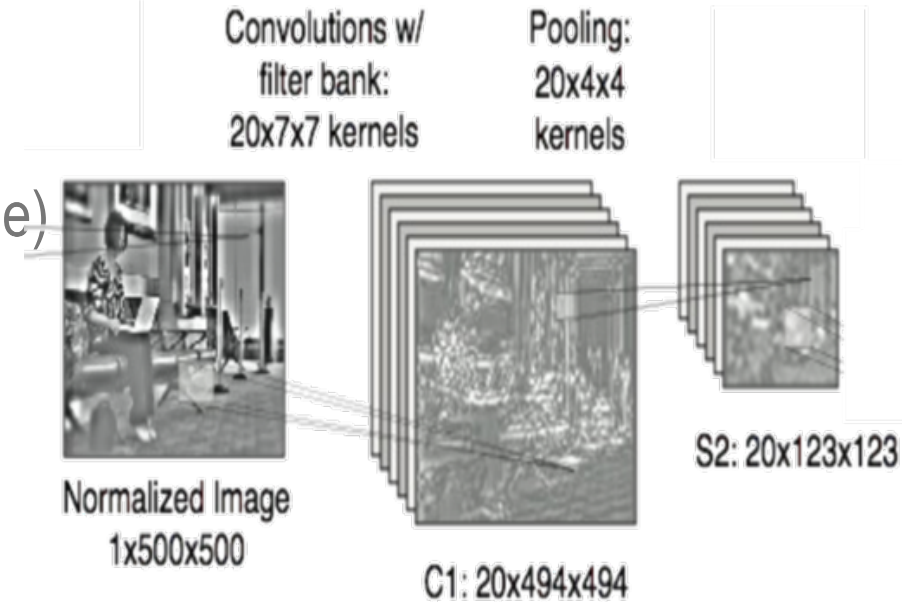
$$H_2 = \lfloor \frac{H_1 - kernel\_size + 2 \times padding}{stride} \rfloor + 1$$

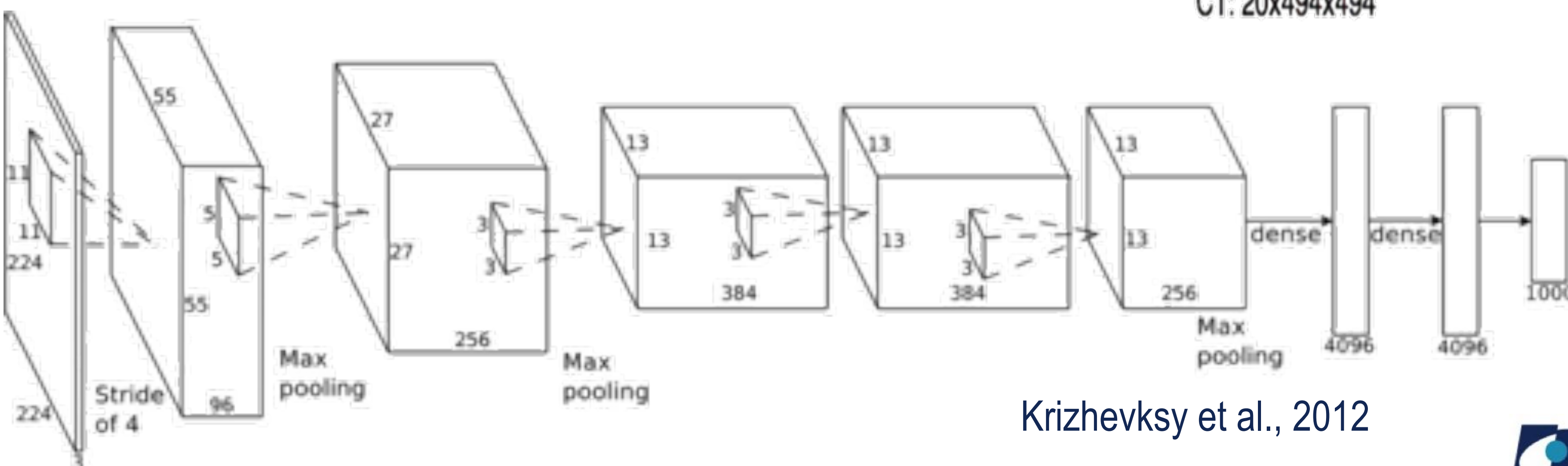
$$W_2 = \lfloor \frac{W_1 - kernel\_size + 2 \times padding}{stride} \rfloor + 1$$

Convolutional Neural Networks

Stack of basic layers

- Convolutions with a given step (stride)
- Non linearity (ReLu)
- Pooling (Reduce resolution)
- Finish with fully connected layers





Krizhevksy et al., 2012

Pre-training

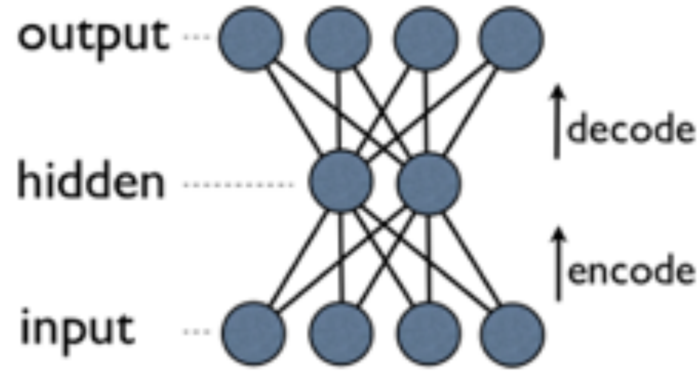
Facing the lack of data

- Good labeled data are expensive to get
- Often, related dataset exists, or unlabeled data are cheap
- Training can be started on these and finished on you problem

Pre-training on a large dataset

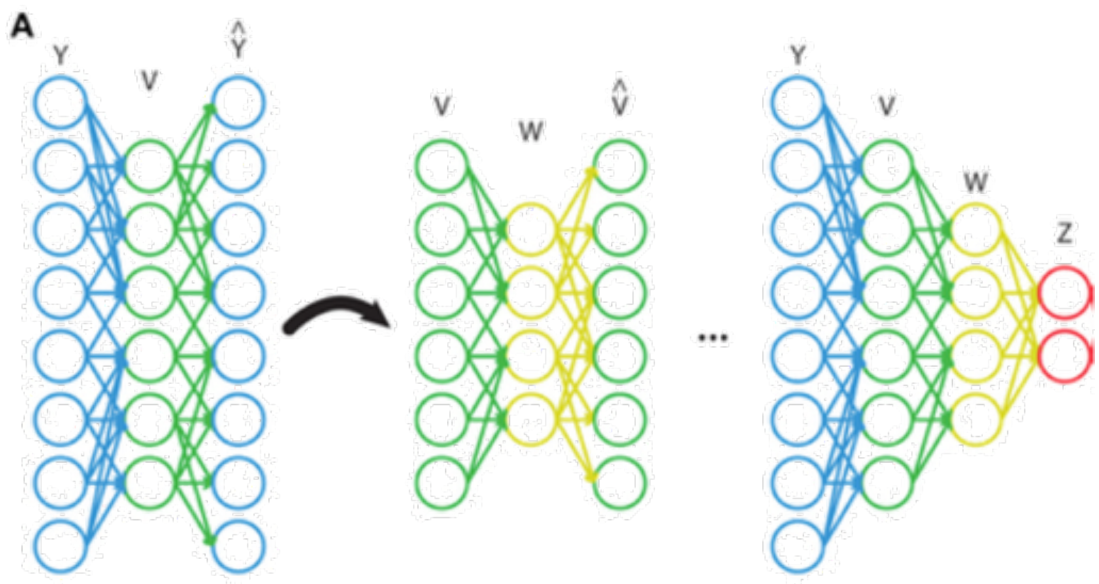
- Train on a large related dataset (e.g. ImageNet when working on image processing)
- Fine-tune (continue training) on your specific problem (with limited data)
- Very common way of starting on a new task

Pre-training



Unsupervised pre-training

- Train on unlabelled data
- Train auto-encoders to reconstruct data with limited information
- Use regularization, dropout…
- Repeat process with hidden layer as input
- Stack the resulting netwoks and fine tune



Applications of Deep Learning for vision

Image categorization

ImageNet Classification

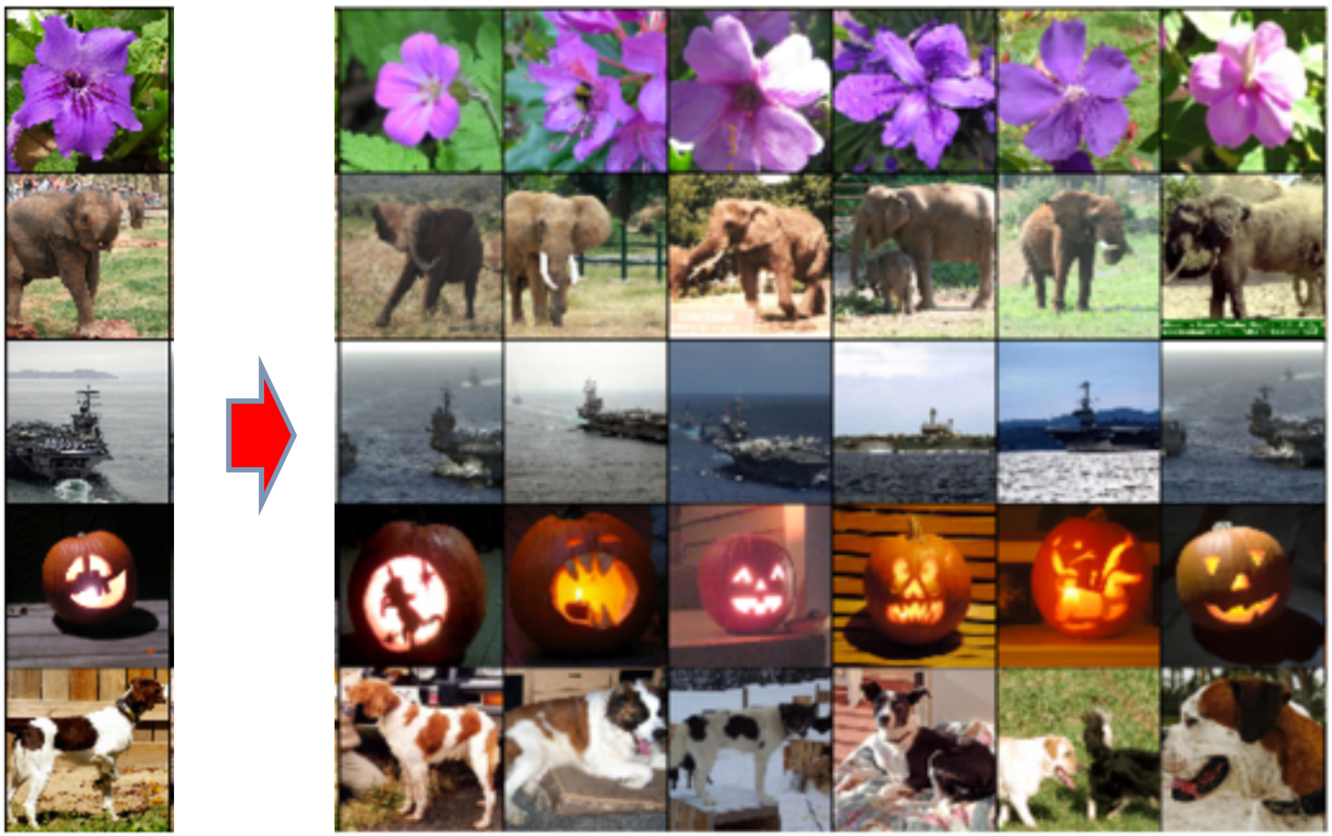
- Large Convnet for classification on ImageNet
- 650K neurons, 832M synapses, 60M parameters Trained with backprop on GPU
- Error rate: 15% (Previous state of the art: 25%)
- Acquired by Google Jan 2013, Deployed for Photo Tagging May 2013

ImageNet Classification with Deep Convolutional Neural Networks Krizhevksy, Sutskever, Hinton 2012

Image categorization

ImageNet Classification

- Last layer feature show strong 'semantic' invariance

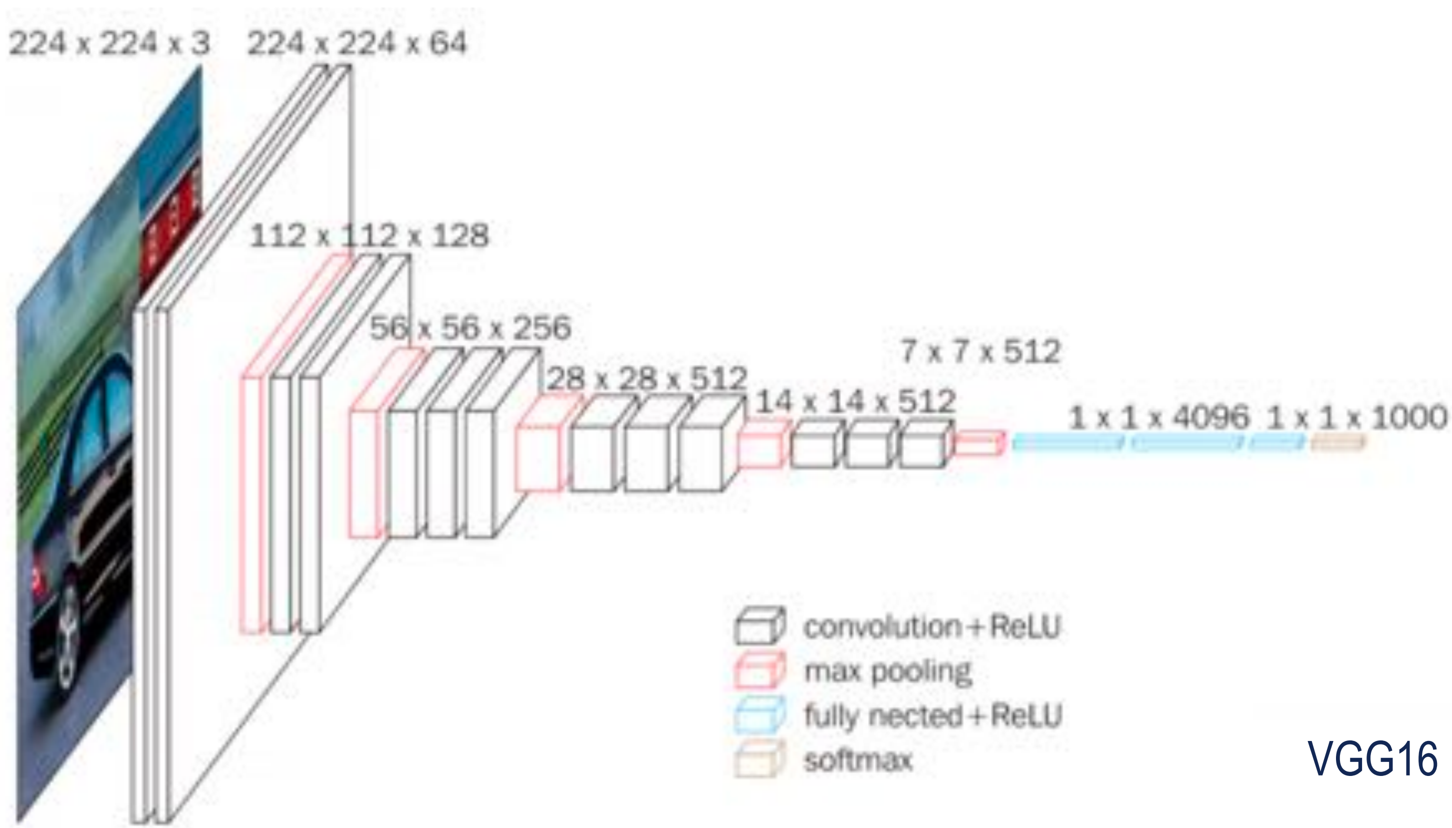


ImageNet Classification with Deep Convolutional Neural Networks Krizhevksy, Sutskever, Hinton 2012

Image categorization

VGG architecture

- Standard architecture, often used as reference



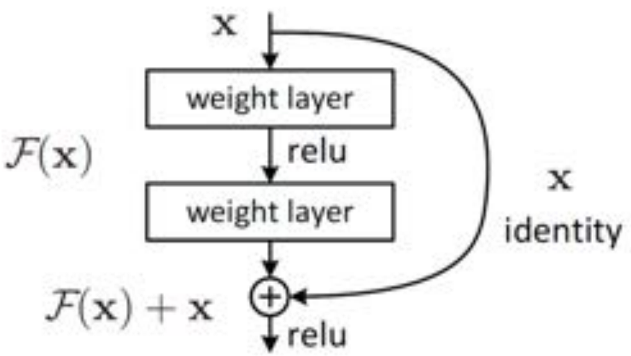
VGG16

K. Simonyan and A. Zisserman “Very Deep Convolutional Networks for Large-Scale Image Recognition”, 2014

Image categorization

ResNet architecture

- Introducing ‘residual layers’ improves performances and training stability



method	top-1 err.	top-5 err.
VGG [41] (ILSVRC'14)	-	8.43†
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian (2015-12-10). "Deep Residual Learning for Image Recognition"

Image categorization

DenseNet architecture

- Generalize densenet by connecting to several forward layers
- Concatenate information instead of summation
- Overall smaller networks because number of layers can le reduced

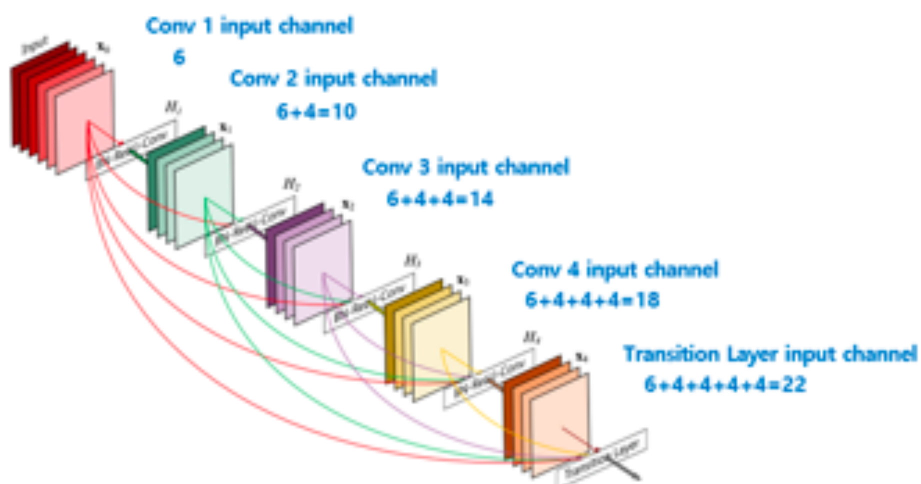
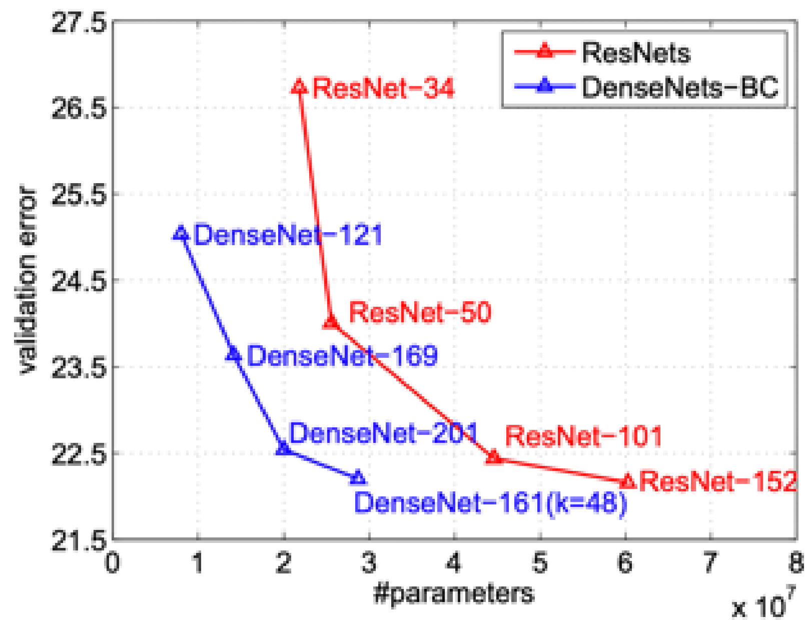


Figure 1: A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature-maps as input.

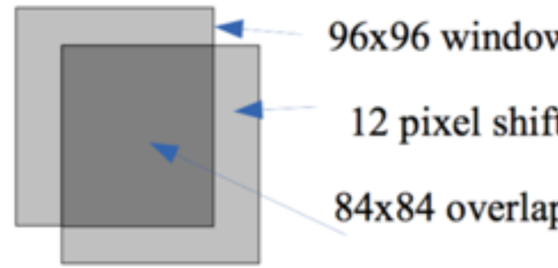


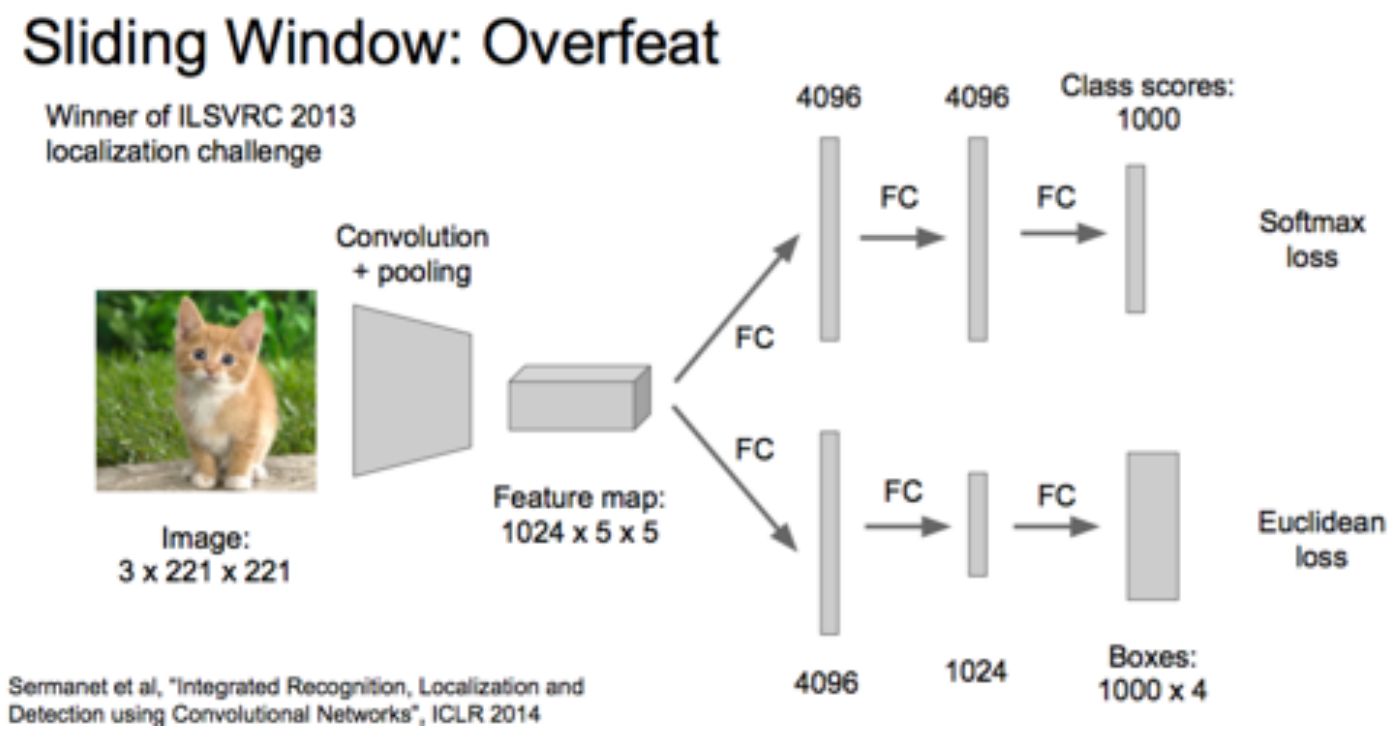
Densely Connected Convolutional Networks, Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger, 2017

Object detection

Application in sliding window approach

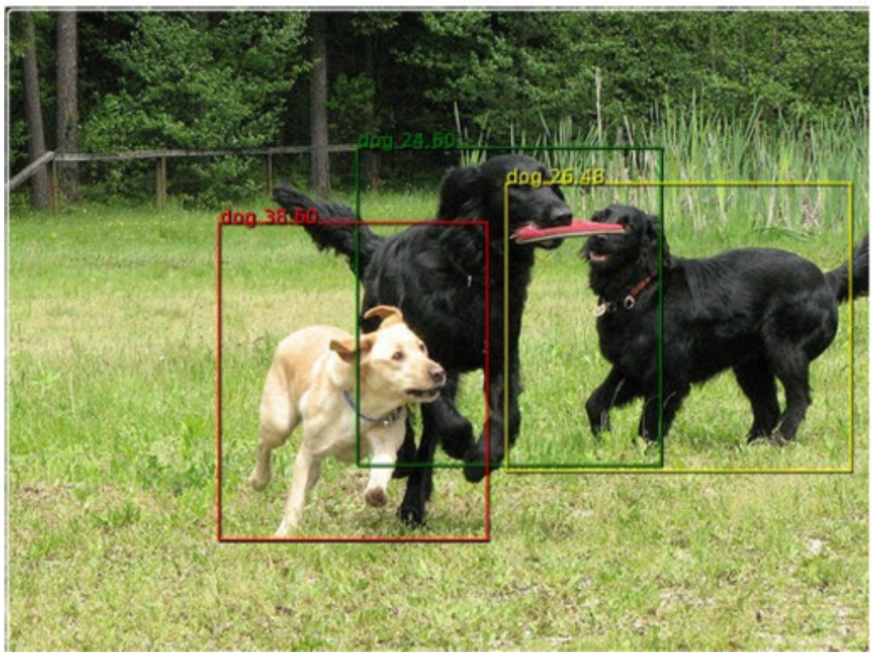
- Convnet can be optimized for sliding windows
- OverFeat (Sermanet et al., 2014)
- Additional output for bounding box regression

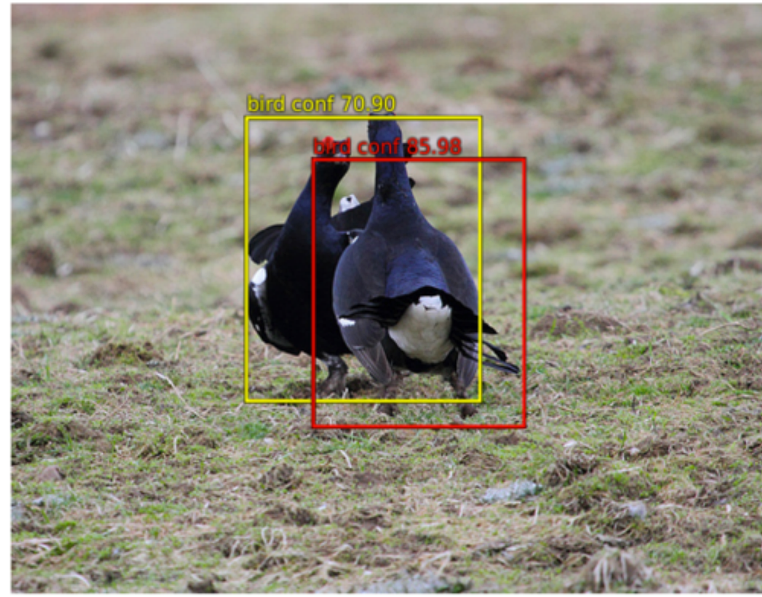




Object detection

Overfeat



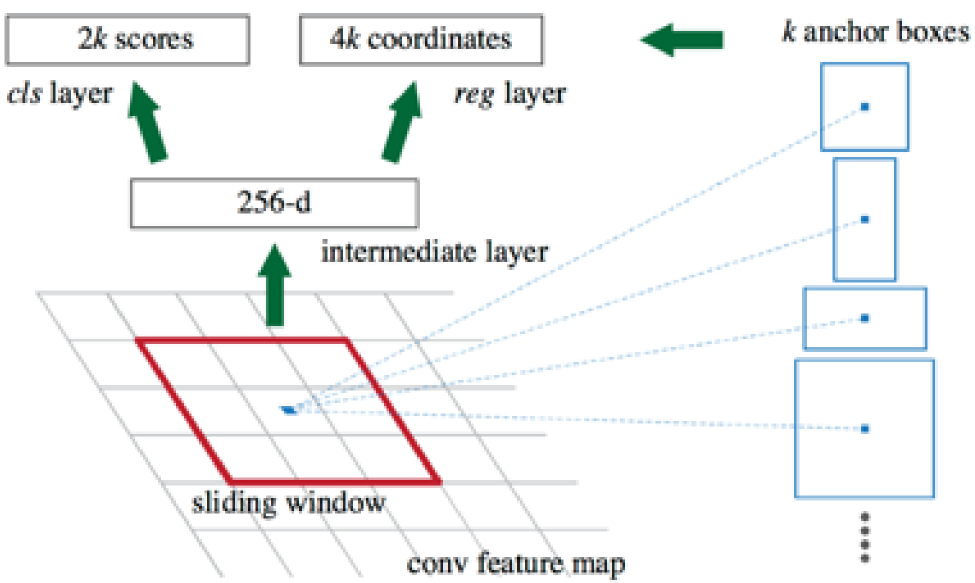


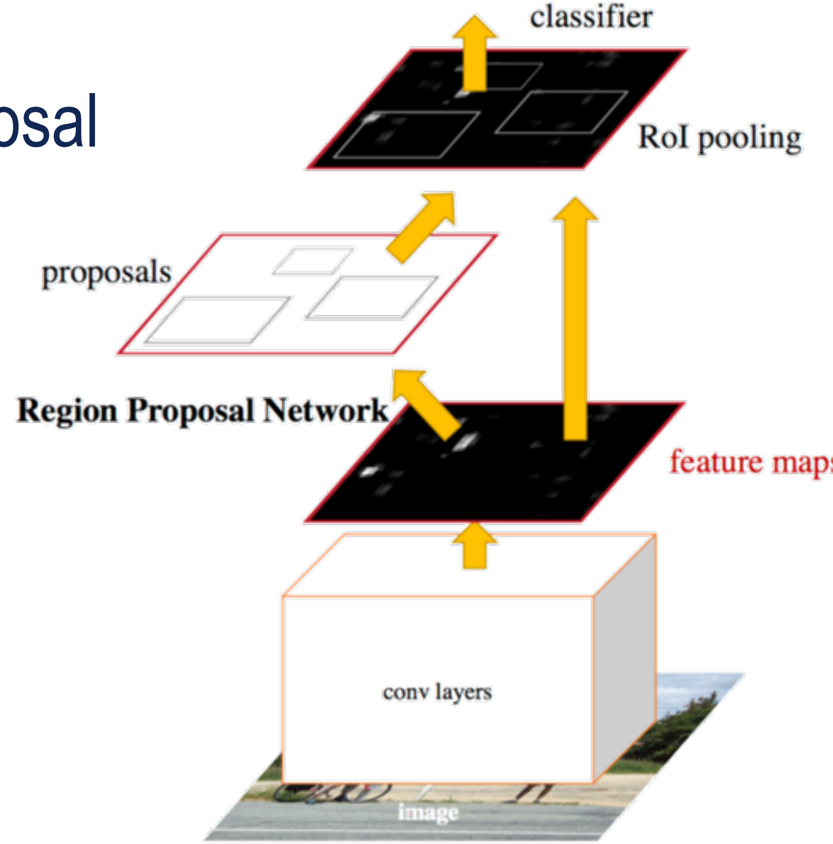
OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks
Pierre Sermanet et al., 2014

Object detection

Application with bounding box proposal

- Predict likely object boxes
- Categorize box content



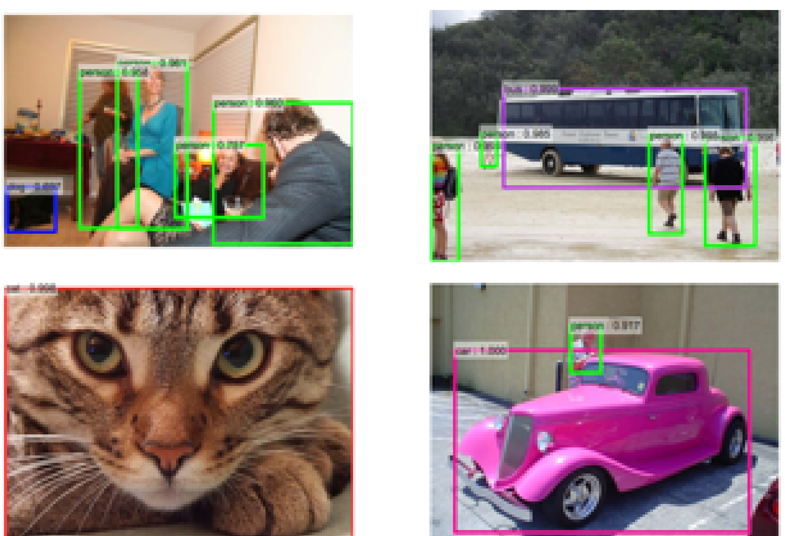


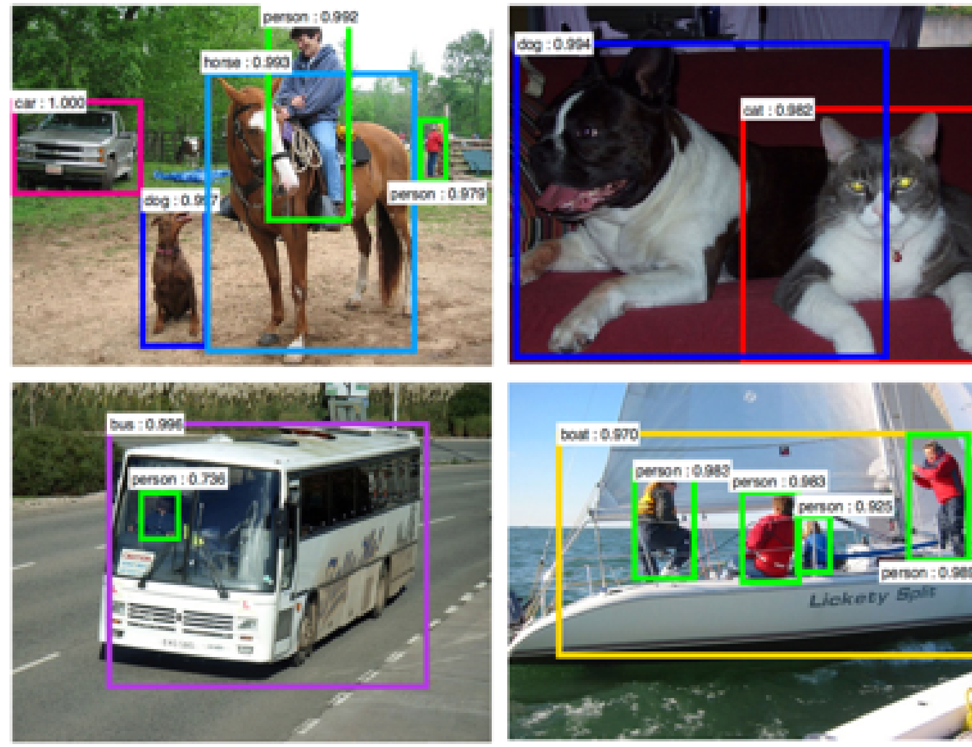
Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks
Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, 2015

Object detection

Faster R-CNN

- Winner of 2015 ILSVRC
- VGG for convolution (13 layers)
- 200 ms/image on GPU
- Deals with large scale variation



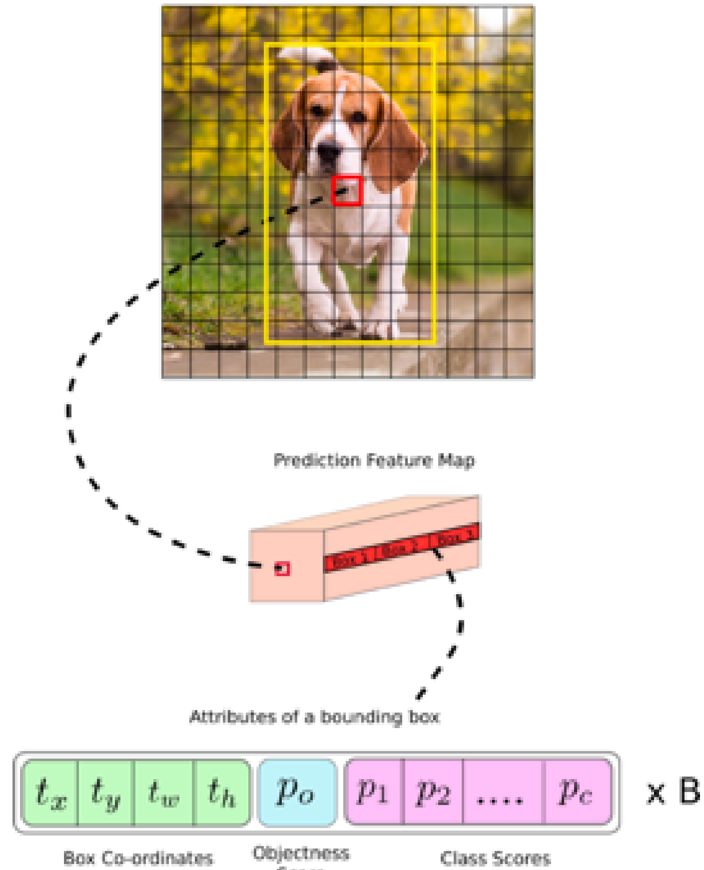


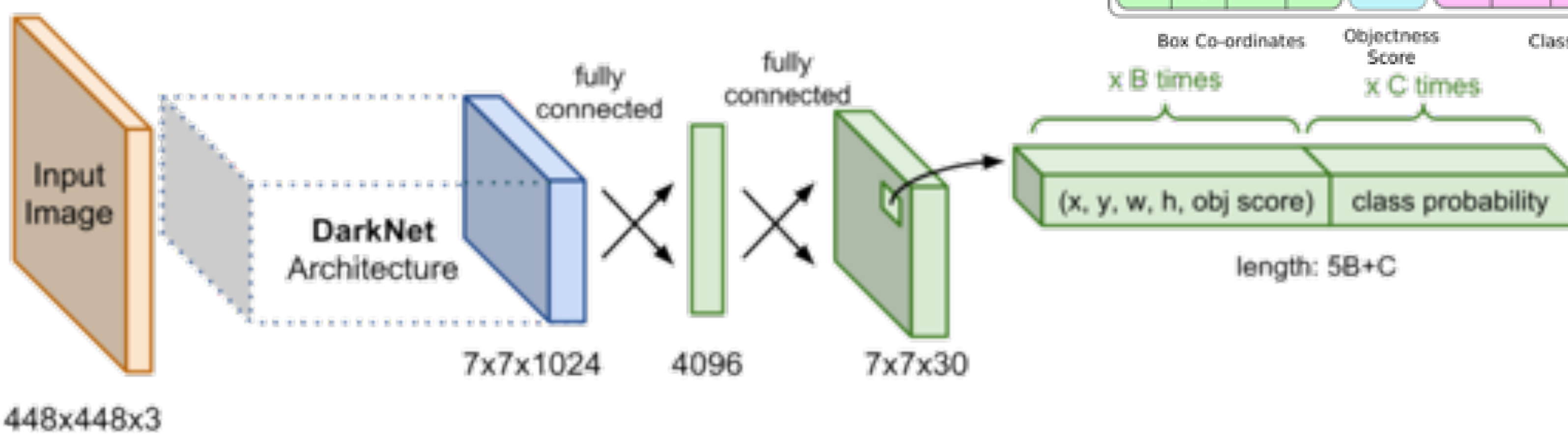
Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks
Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, 2015

Object detection

YOLO et al.

- Predict boxes/object for all position at once
- Only one forward pass
- Much faster than R-CNN
- Similar to SSD, MobileNet …

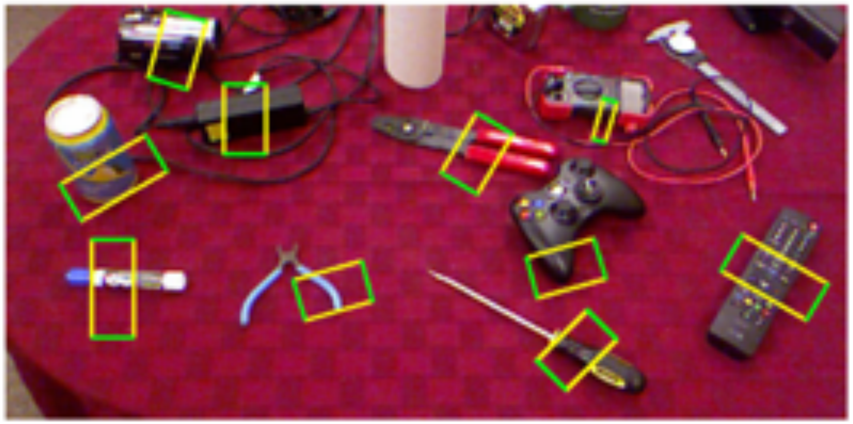




You Only Look Once: Unified, Real-Time Object Detection Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, 2016

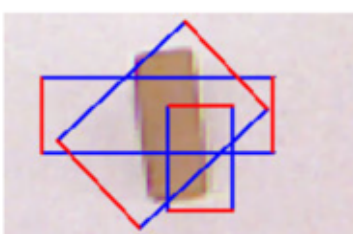
Object Grasping

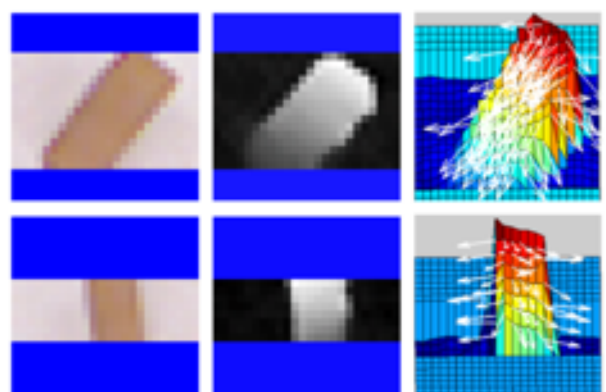
Finding grasp position from RGB-D images



Classification task : graspable/not graspable

Multimodal input : RGB + Depth + Normals



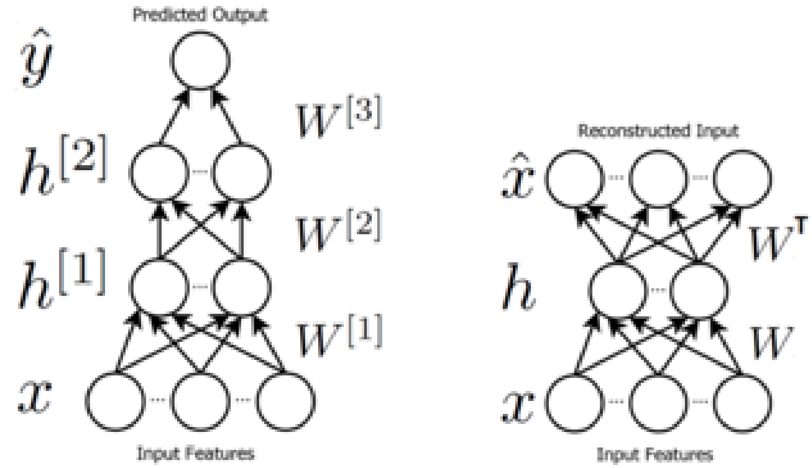


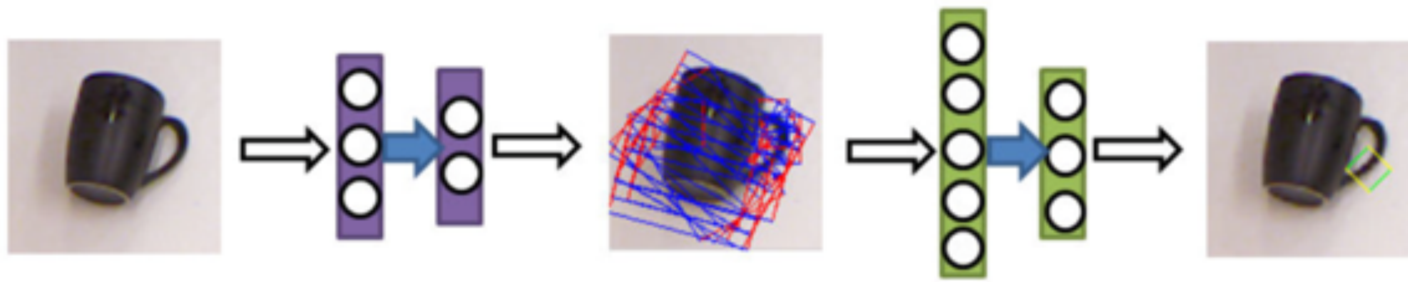
Deep Learning for Detecting Robotic Grasps
Ian Lenz, Honglak Lee and Ashutosh Saxena, 2014

Object Grasping

Simple "Deep" network

- Two layers …
- Pre-training with auto-encoder
- Cascade of 2 networks to speed-up computation



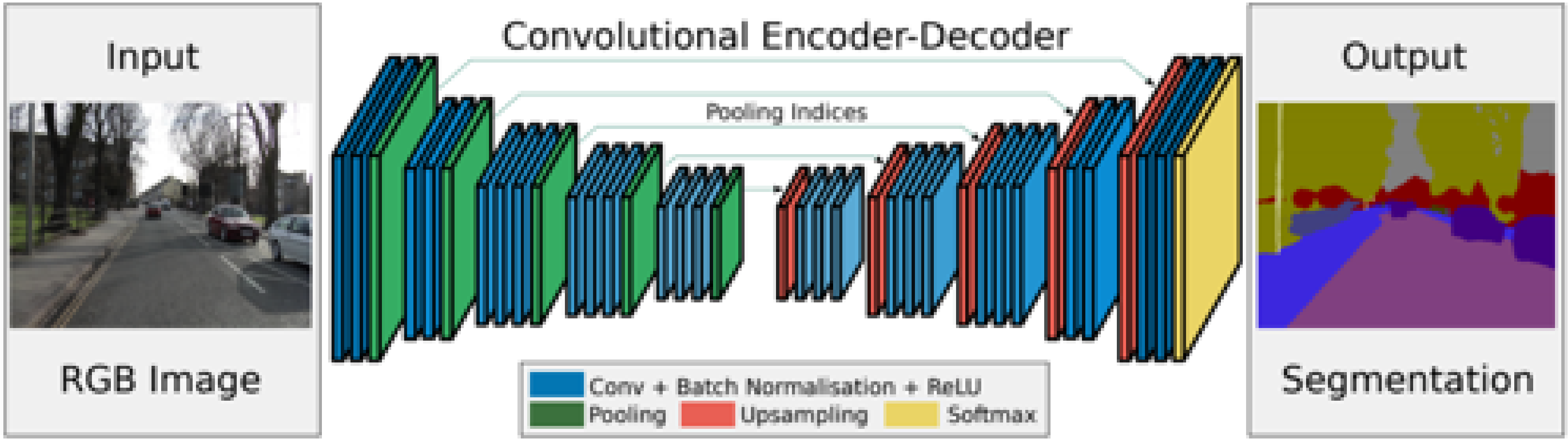


Deep Learning for Detecting Robotic Grasps
Ian Lenz, Honglak Lee and Ashutosh Saxena, 2014

Semantic segmentation

Encoder / decoder network

- Use convolution/pooling
- Then generate label image using upsampling/unpooling
- Standard training using gradient descent



Segnet: A deep convolutional encoder-decoder architecture for image segmentation
V Badrinarayanan, A Kendall, R Cipolla - 2015

Semantic segmentation

Encoder / decoder network

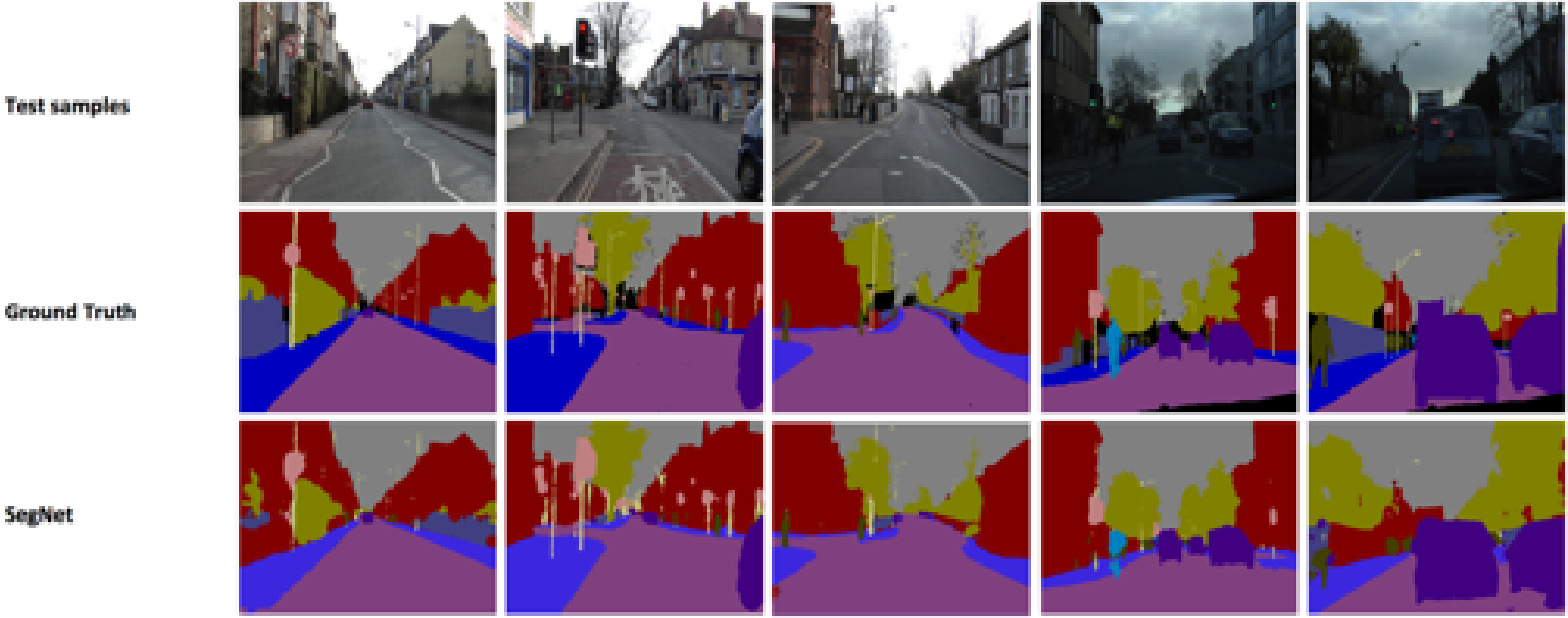
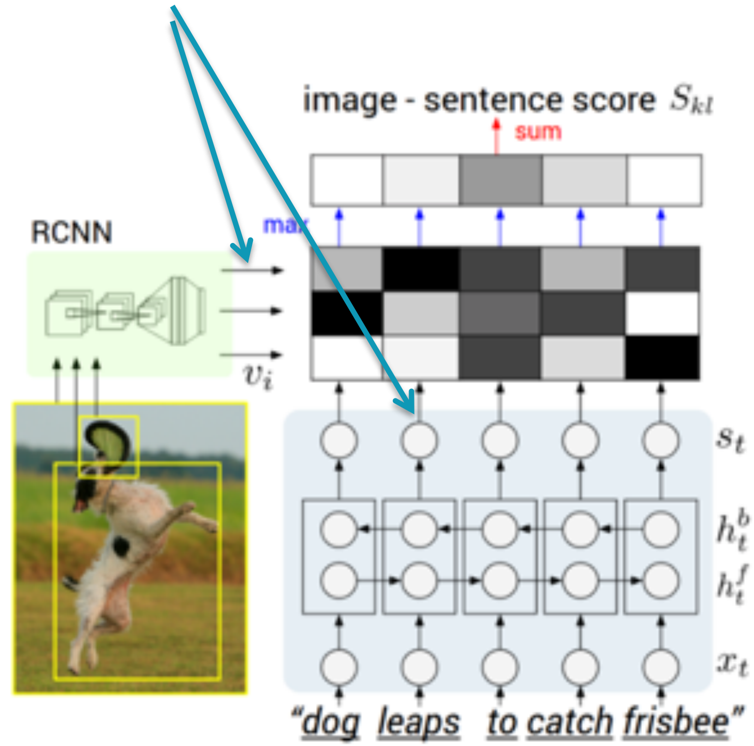


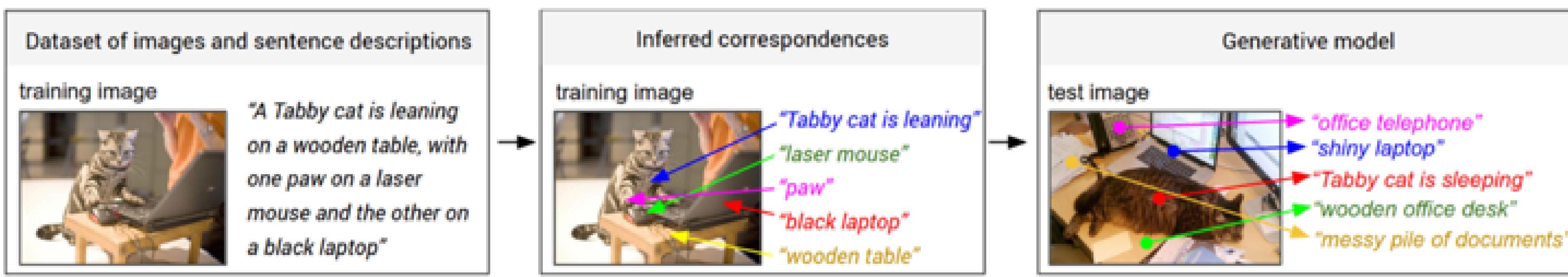
Image captioning

Generating image description

- Learn to describe images from examples
- Learning first aligns objects with words sequences
- Creates a common representation for words and images

Same representation space





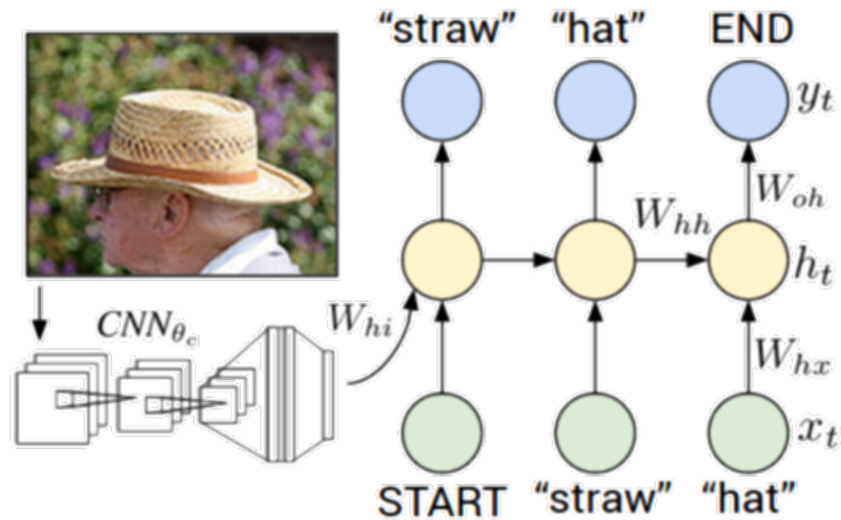
Deep Visual-Semantic Alignments for Generating Image Descriptions
Andrej Karpathy Li Fei-Fei

Image captioning

Generating image description

- A recurrent Neural network is trained

To generate sentences starting from image encoding



man in black shirt is playing guitar.

construction worker in orange safety vest is working on road.

two young girls are playing with lego toy.

boy is doing backflip on wakeboard.

Deep Visual-Semantic Alignments for Generating Image Descriptions
Andrej Karpathy Li Fei-Fei, CVPR 2015

Deep Learning: summary

Deep learning works well

- Can be applied to lots of different tasks
- Very versatile approach
- Best performances in many vision tasks

But be aware of

- Very computationally intensive (can be optimized though)
- Need a lots of training data
- Quite sensitive parameters and open architectural possibilities

Deep Learning: practical

COLAB practical

- Go to : https://colab.research.google.com/drive/1cKLQfym5i2eQXOb0kjpgYtsRl6vNanRU
- In the ‘File’ menu, select ‘Save a copy in Drive’
- Follow the notebook
- Send a PDF report to david.filliat@ensta-paris.fr

Fin