Modern object detection
from old-school to Deep learning

Paul Blondel, PhD

UPJV, France

June 15, 2019

What is Object Detection?

- It is a **sub-field of the Computer Vision field**
- Goal: **extract scene information** from still images or frames
- **Detect** one or more **instances** of one or more **object classes**

In the world: the number of cameras increase so is the need to analyze video frames.

Object detectors help analyzing the content of these frames automatically, in a convenient manner.

The job of an object detector is **not so easy** ...

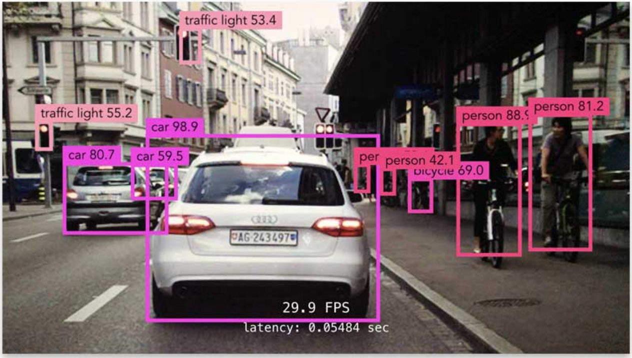
- object instances may have **different scales**
- object instances may be **partially occluded**
- some object **classes maybe be very similar**
  - ex: lions and cats
- within a same object class we may have **different texture, color, etc**
  - ex: human people wearing **different clothes**
- object instances may have **various orientations and postures**
- etc

On top of that, object detectors may be subject to other constraints:

- **real-time** performance
- must work on **embedded systems** with **less powerful** hardware
  - ex: UAVs, etc.
- the **orientation** of the camera **may change**
  - ex: UAVs, etc.
- the **weather conditions** may **affect the quality of the image**
- must work at **night**
- etc

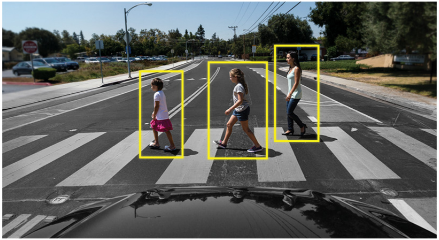
 

As you guessed it, the detected **object instances** are **surrounded by color rectangles**, like here:

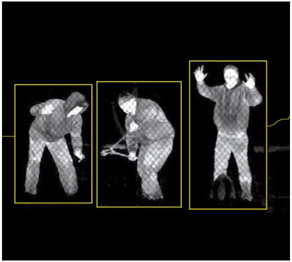


Some examples of applications:

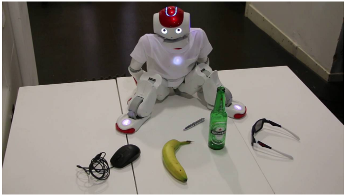
- Advanced Driver Assistance System (ADAS)



- Video surveillance:



- Robot object manipulations



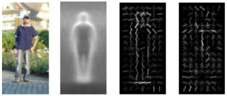
- Face detection in the subway

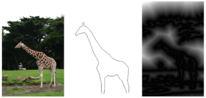
- Face check in train station (+recognition)

First, let's have a look to this image:

What **visual features** are interesting here?

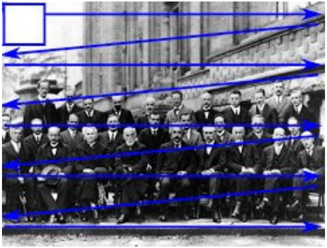
- texture
- shape
- colors
- salience
- movement (when working with frames)
- etc



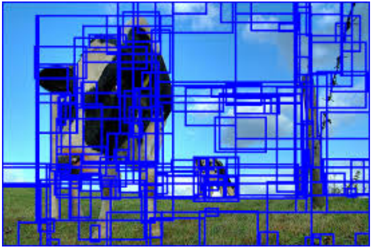


Visual features can be extracted **multiple locations** either using ...

- ... the **sliding-window** approach

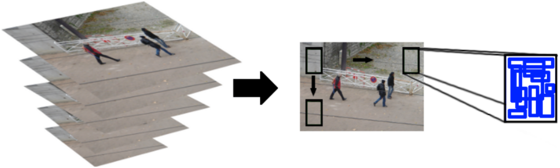


- ... or proposed **regions of interest (region proposals)**



The **object instances** we want to detect may have **different scales**:

One way to deal with that is by building a **image pyramid**:



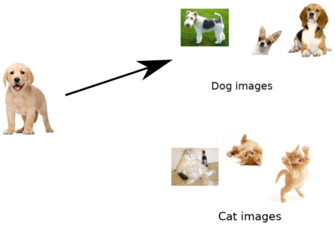
Indeed, the **detection window** must always has the **same size** so:

- up-scaled levels permit to detect small instances
- down-scaled levels permit to detect big instances

We now can extract **visual features** at **multiple locations** and **scales** and:

- compare them to **databases of images**
- or, compare them to **templates**
- or, analyze them with a **pre-built model**
- ...

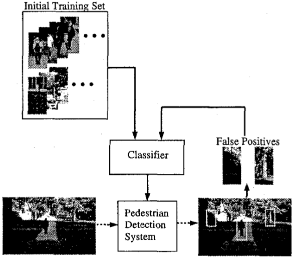
The **goal** of this step is to **find** the **nearest object class**.



Because **detection windows are analyzed at nearby locations** the detector may trigger **several detections nearby object instances**:

One way to only **keep the best detections** (having the highest scores) is to use: **Non-Maximum Suppression** (NMS).

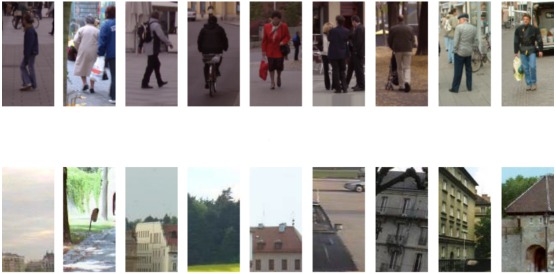
- Before the use of Machine Learning: **poor performances** due to the **weakness of the model**
- In 1998: Papageorgiou et al[1] proposed to train a model based on visual features
- This was the **beginning** of the use of **Machine Learning** algorithms for Object Detection
- Papageorgiou et al: SVM Machine Learning algorithm to **train a model fed with visual features**



[1] **A general framework for object detection**, ICCV 1998

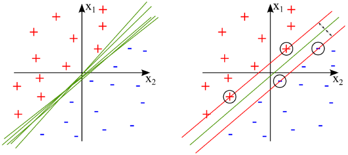
At first, to **train the model** we **need a lot of image examples**:

- images of **object instances** (ex: images of people)
- images of **random background images**

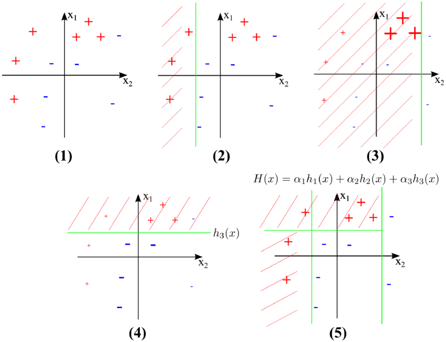


Two very famous Machine Learning algorithms:

- Support Vector Machine (SVM)

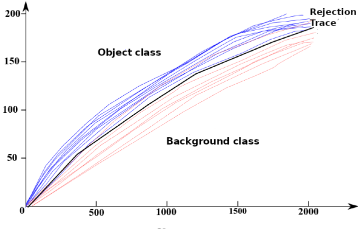


- Boosting



From 1998 to 2014, there have been **numerous model improvements**, such as:

- The Cascade and Soft-Cascade Boosting (ICF, ACF, etc) for speed

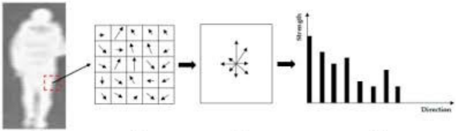


- Or the Latent-SVM (DPM) for robustness

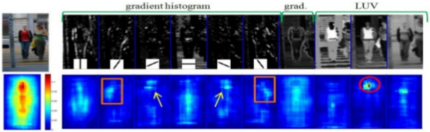
- ...

And there have been been **numerous feature engineering improvements**, such as:

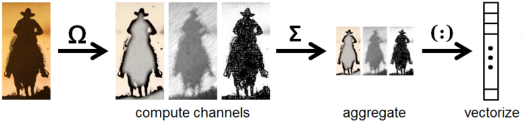
- SIFT-like Histogram of Oriented Gradients (HOG)



- Integral Channel Features (ICF)



- Aggregate Channel Features (ACF)

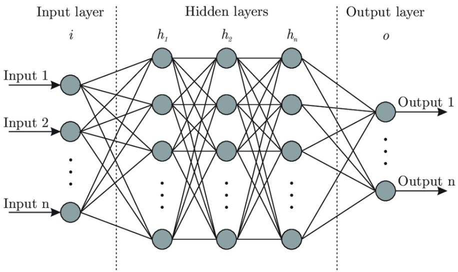


The **come back of Artificial Neural Networks**:

- Artificial Neural Networks (ANN) **exist for a very long time** (50's)
- The more recent SVM eclipsed ANN for a while
- It was due to many things combined:
  - No adequate learning approaches (now: back propagation)
  - Vanishing gradient problem (now: new activation functions, batch norm, etc)
  - Require a lot more training data (now: Amazon Mechanical Turks)
  - Training require powerful computers (now: GPGPU, processing power is cheaper)
  - Over-fitting (now: dropout layers, etc)
- Among all ANN: **Convolutional Neural Network** (CNN) is the **most suitable for Computer Vision**

New learning approaches, fixing the Vanishing gradient problem, having more labeled data and more powerful computers: this all contributed to the emergence of Deep learning (Deep means more than 4/5 layers)

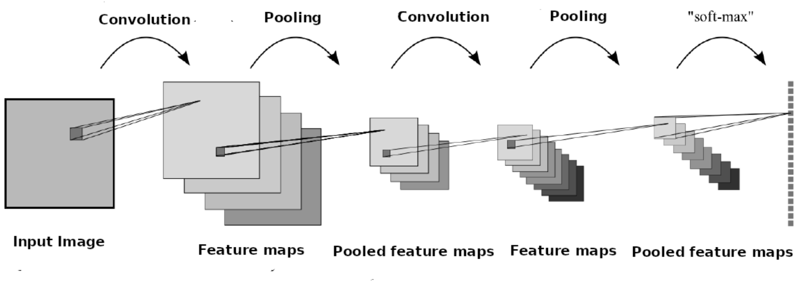
How works an Artificial Neural Network?



- The **weights of the hidden layers form the model**
- Inner nodes output a value **with respect to an activation function**
- The **weights of the hidden layers** are trained by **back propagation**:
  - gradient descent adapted for ANN training
  - for each training sample gradient is computed and weights are adjusted

This architecture is **more suitable for Computer Vision tasks**

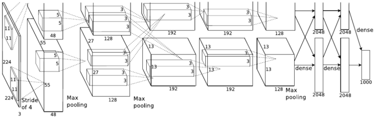
The Convolutional Neural Network (CNN) is as follow:

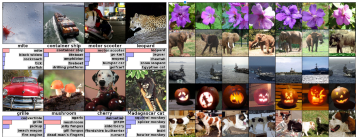


- Convolution: **only local pixels** are **connected towards the same pool node**
- Pooling: **features** computed in the convolution layers **are aggregated** (max or average over a pool)
- Images = many pixels, thanks to CNN: we **don't need a tremendeous number of connections**

Using a Deep CNN on ImageNet:

- The ImageNet dataset contains 1000 object classes!
  - 1.6 millions of classified images
  - Thanks to Amazon Mechanical Turks
- Krizhevsky et al won the contest in 2012 with a DCNN [2]
  - 5 convolutional layers (first two followed by pooling layers)
  - Followed by 4 dense layers
  - Output 1000 object classes
  - 37.5% error rate (previous best: 45.1%)



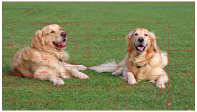


But here this is not object detection this is object classification

[2] Imagenet classification with deep convolution neural networks, NIPS 2012

What about object detection?

- [RECALL] In order to detect objects anywhere in an image, two approaches:
  - A **sliding detection window**
  - Or analyze **region proposals**
- In 2014, Girshick et al proposed the R-CNN [3]:
  - Use **region proposals** (Selective Search)
    - class independent object proposals
  - Feature are computed with a Deep CNN
  - Eventually features are classified with multiple SVMs
  - 53.7% of mAP PASCAL 2010 (previous best: 33.4%)

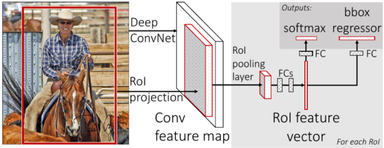


R-CNN is the first object detector based on deep learning!

[3] Rich feature hierarchies for accurate object detection and semantic segmentation CVPR 2014

A fast version of R-CNN ...

- With R-CNN: **features are extracted for each proposal**
- With Fast R-CNN[4]:
  - **features are computed once** and shared for all proposals
  - the whole image is **processed once** (not all proposals)
  - this means faster analysis of the scene!



One step closer to a 100% NN detector: the Fast R-CNN takes in input one image and some proposals

[4] Fast R-CNN, ICCV 2015

An **even FASTER** version of R-CNN:

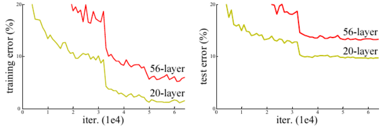
- With Fast R-CNN and R-CNN: region proposals are generated by the Selective Search algorithm
- But **Selective Search: very slow** and not optimized (2s/image on CPU)!
- With Faster R-CNN[5]:
  - **First part of the NN** dedicated to **generate region proposals** (Region Proposal Network)
  - **Second part of the NN** dedicated to **feature analysis and decision making**
  - Region proposals generation and classification in the **same NN**

Another step towards a 100% NN detector: regions proposals are generated within the NN

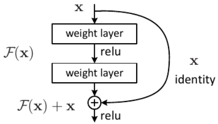
[5] Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, NIPS 2015

Bigger NN: a priori, having a deeper NN permits a better learning:

- But: **very deep networks** are **more difficult to train**!
  - despite having resolved the vanishing gradient problem
  - when the depth increase the **accuracy gets saturated**



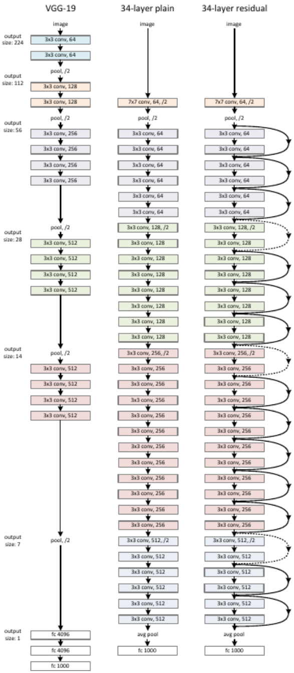
- He et al[6] proposed residual learning to solve that
  - they observed that it is easier to learn a residual mapping $(F(x) + X)$



  - we can **stack many of these blocks** for a **bigger NN** (=>100 layers!)
  - 19.38% error rate on ImageNet! (previous best: 37.5%)

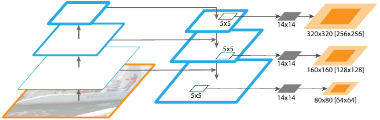
Residual learning: A step towards more powerful NN!

[6] Deep Residual Learning for Image Recognition, CVPR 2016


VGG-19
34-layer plain
34-layer residual
image
output size: 224
3x3 conv, 64
pool, /2
output size: 112
3x3 conv, 128
7x7 conv, 64, /2
output size: 56
3x3 conv, 256
output size: 28
3x3 conv, 512
3x3 conv, 128, /2
output size: 14
3x3 conv, 256, /2
output size: 7
3x3 conv, 512, /2
output size: 1
fc 4096
fc 1000
avg pool

FPN[7] for better detection performance at multiple scales:

- Improving scaling robustness = better detection performance
- Since R-CNN: the **various scales** of objects are **implicitly learned**
- As mentioned before: **image pyramids explicitly handle scales**
- With FPN:
    - **image pyramid** is **incorporated within the NN**



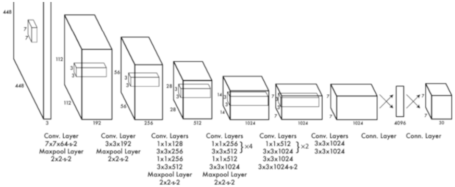
    - Faster R-CNN VS FPN: Improve by 2% in AP (COCO dataset)

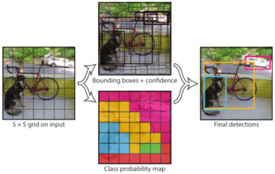
Contrary to what researchers thought: explicitly managing multiple scales improve significantly the performance.

[7] Feature Pyramid Networks for Object Detection , CVPR 2017

You Only Look Once detector (YOLO)[8], an alternative approach:

- Here: object detection is seen as a **regression problem**
  - 1 NN predicts **bounding boxes** AND **class probabilities** (full images)
  - there is **no need to generate region proposals**!
- YOLO learns a more **general representations of the objects** (context is also learned!)
- YOLO is more **easy to train** (no need to learn and merge two NNs)
- Performances (compared to the classification approaches):
  - makes more localization errors... but handles better the background
  - much faster: 45 FPS on Titan GPU!

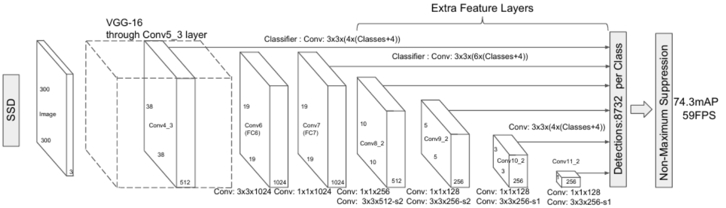




[8] You Only Look Once: Unified, Real-Time Object Detection, CVPR 2016

The SDD[9] detector, an improvement of the regression approach:

- With **YOLO** the detection **accuracy is not competitive**
- SSD resolves this issue by...
  - ... **combining** the predictions obtained at **multiple scales**
  - thus the **accuracy increases**! (74.3 mAP VS 63.4 mAP on PASCAL VOC 2007)
  - Liu et al. also simplified the network to be **speeder** (59 FPS VS 45 FPS)

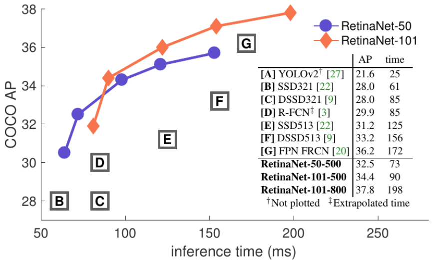


The "single shot" / regression approach becomes more competitive

[9] Ssd: Single shot multibox detector, ECCV 2016

RetinaNet[10] is a VERY competitive regression object detector:

- **Regression detectors** have **low accuracy because of class-imbalance**
- Lin et al proposed a new training loss function:
  - The new loss is called Focal Loss
  - It helps **focusing** the learning on **hard background images**
- Outperforms all classification approaches on COCO (speed VS AP)!



	AP	time
[A] YOLOv2† [27]	21.6	25
[B] SSD321 [22]	28.0	61
[C] DSSD321 [9]	28.0	85
[D] R-FCN‡ [3]	29.9	85
[E] SSD513 [22]	31.2	125
[F] DSSD513 [9]	33.2	156
[G] FPN FRCN [20]	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	37.8	198

†Not plotted ‡Extrapolated time

A great step towards an all-in-one NN object detector

[10] Focal Loss for Dense Object Detection , ICCV 2017

Object detection:

- Most former Machine Learning approaches are obsolete
- Performances have considerably improved since Deep Learning
- Year after year, Deep Learning-based Object Detection becomes ...
  - ... simpler (one step training, etc.)
  - ... more accessible (cheaper and cheaper powerful GPU, etc.)
  - ... more accurate (new optimizations, etc.)
  - ... speeder.
- A clear trend: having an unique NN performing most detection steps

Note that in 2018, the authors of YOLO released YOLOv3[11] which is approximately two to three times faster than RetinaNet, with comparable accuracy ...

The course continue: the ceiling is not yet reached!

[11] YOLOv3: An Incremental Improvement , technical report on arXiv, 2018

Questions?