Deep Learning for Computer Vision

GAIN project

Tbilisi, Georgia

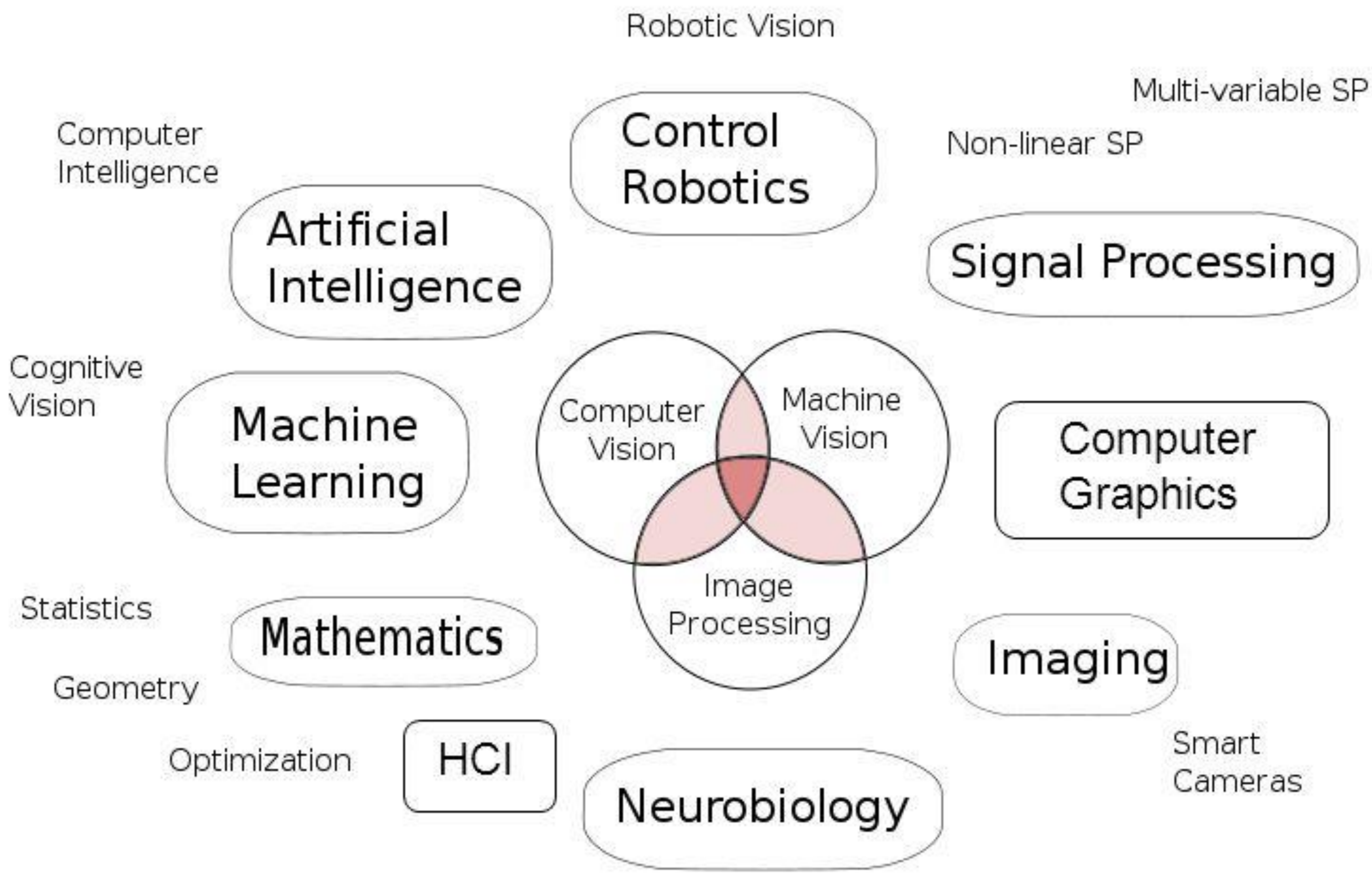
12 August 2024

STARS team

informatics mathematics

UNIVERSITÉ
CÔTE D'AZUR

Vision is multidisciplinary



- **Computer Vision** is a subfield of artificial intelligence as machine learning.
- Techniques in machine learning and other subfields of AI (e.g. NLP) can be borrowed and reused in computer vision.

Computer Vision: many Tasks

Computer Vision is an interdisciplinary scientific field that deals with how computers can be made to gain high-level understanding from digital images or videos.

From the perspective of engineering, it seeks to automate tasks that the human visual system can do. [Wikipedia]

Computer Vision Tasks:

- Recognition of Entities : Images, 2/3D Objects, People/Pose/Face/Gaze or Emotions/Events
  - Classification
  - Detection, segmentation
  - Retrieval
- Motion analysis
  - Optical flow
  - Tracking of objects, ReID
- Image/video synthesis, generation
- Image restoration, super resolution, denoising, 3D geometry
- Biometrics, medical image, remote sensing,..
- etc...

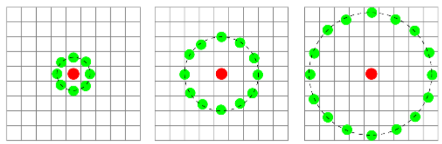
Video Analytics (or VCA) applies CV & ML algorithms to extract/analysis content from videos

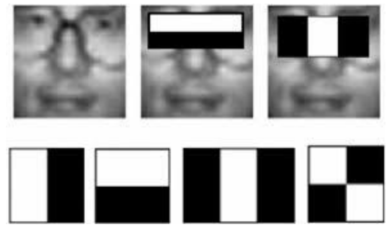
Video Analytics: many research Domains

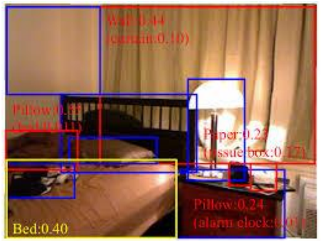
- Smart Sensors: Acquisition (dedicated hardware), thermal, omni-directional, PTZ, cmos, IP, tri CCD, RGBD Kinect, FPGA, DSP, GPU.
- Networking: UDP, scalable compression, secure transmission, indexing and storage.
- Image Processing/**Computer Vision**: feature extraction, Deep CNN, 2D object detection, active vision, tracking of people using 3D geometric approaches
- Event Recognition: Probabilistic approaches HMM, DBN, logics, symbolic constraint networks
- Multi-Sensor Information Fusion: cameras (overlapping, distant) + microphones, contact sensors, physiological sensors, optical cells, RFID
- Reusable Systems: Real-time distributed dependable platform for video surveillance, OSGI, adaptable systems, Machine learning
- System Optimization: complexity reduction (# parameters, Flops) matrix factorization, distillation
- Visualization: 3D animation, ergonomic, video abstraction, annotation, simulation, HCI, interactive surface.

A brief history of Computer Vision

Geometric, Statistics, handcrafted features







LBP, 1994
Local Binary Patterns

Viola & Jones, 2001
Face Detection

Dalal & Triggs, 2005
HOG

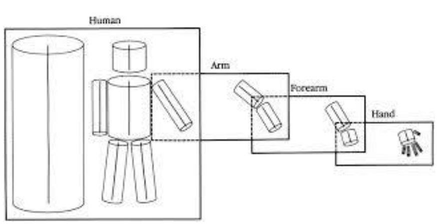
Everingham, 2012
PASCAL Challenge

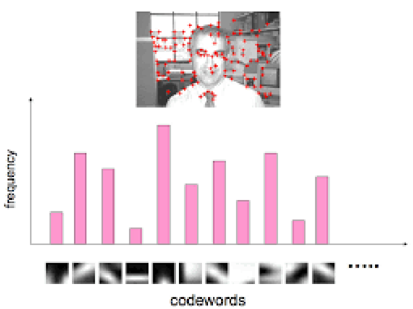
David Marr, 1970s
from images to geometric
blobs, edges, 3-D models

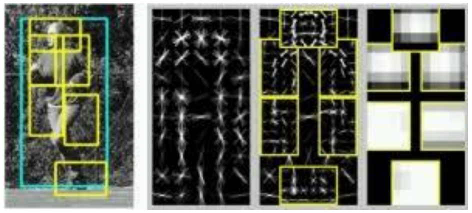
David Lowe, 1999
SIFT

Sivic & Zisserman, 2003
Bags of words

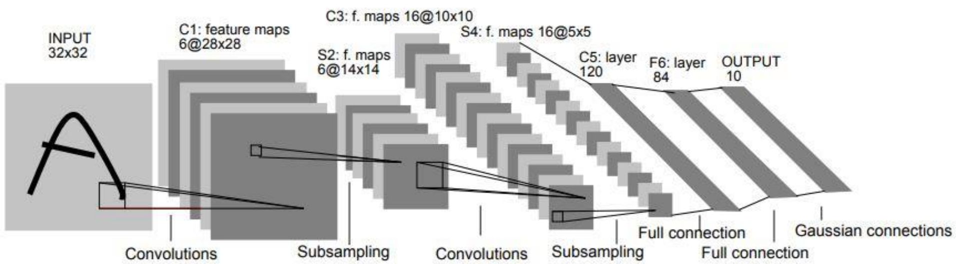
Felzenswalb & Ramanan, 2009
Deformable Part Model

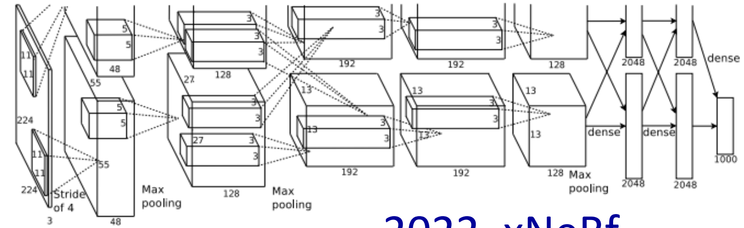






A brief history of Deep Learning





2022, xNeRf
Diffusion M

Minsky & Papert, 1969
perceptron

LeCun, Bengio, 1998
LeNet-5
Gradient-based learning

Krizhevsky, Hinton, 2012
AlexNet

NAS, 2018
3D Conv, GAN

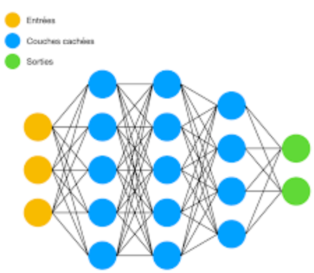
2023
Foundation M
Visual prompt
Adapters

LeCun, 1990
convolutional networks

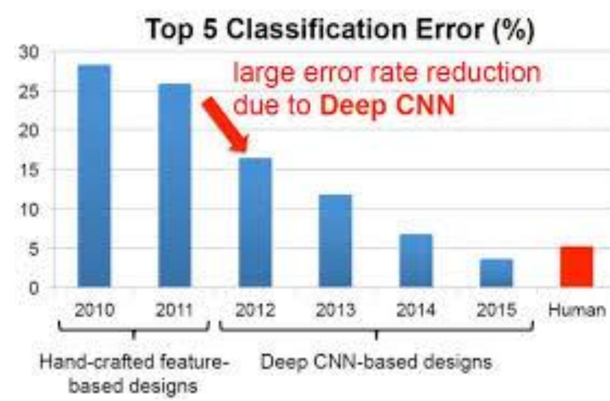
Li Fei-Fei, 2009
Image-net
22K categories and 15M images

Ross Girshick, 2016
Faster RCNN, ResNet

2020
Transformers
DETR



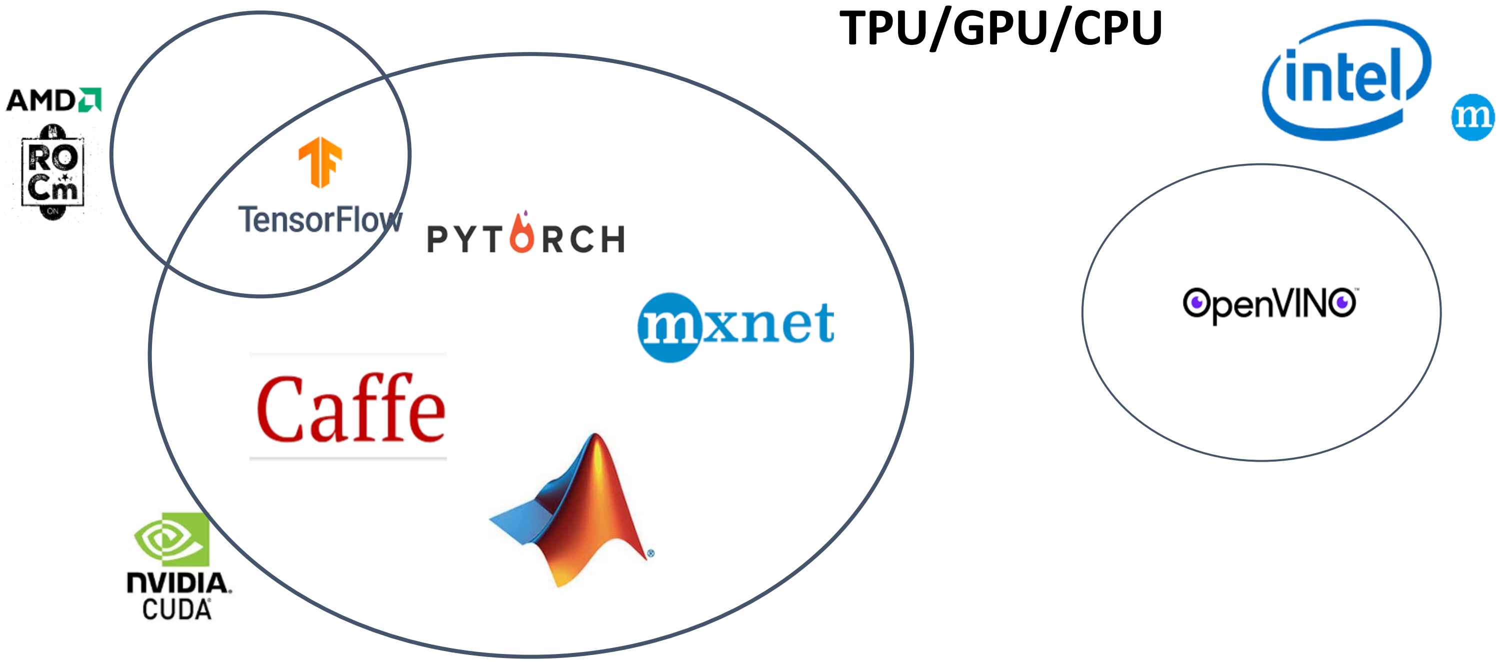
ImageNet Large Scale Visual Recognition Challenge
Russakovsky et al. IJCV 2015



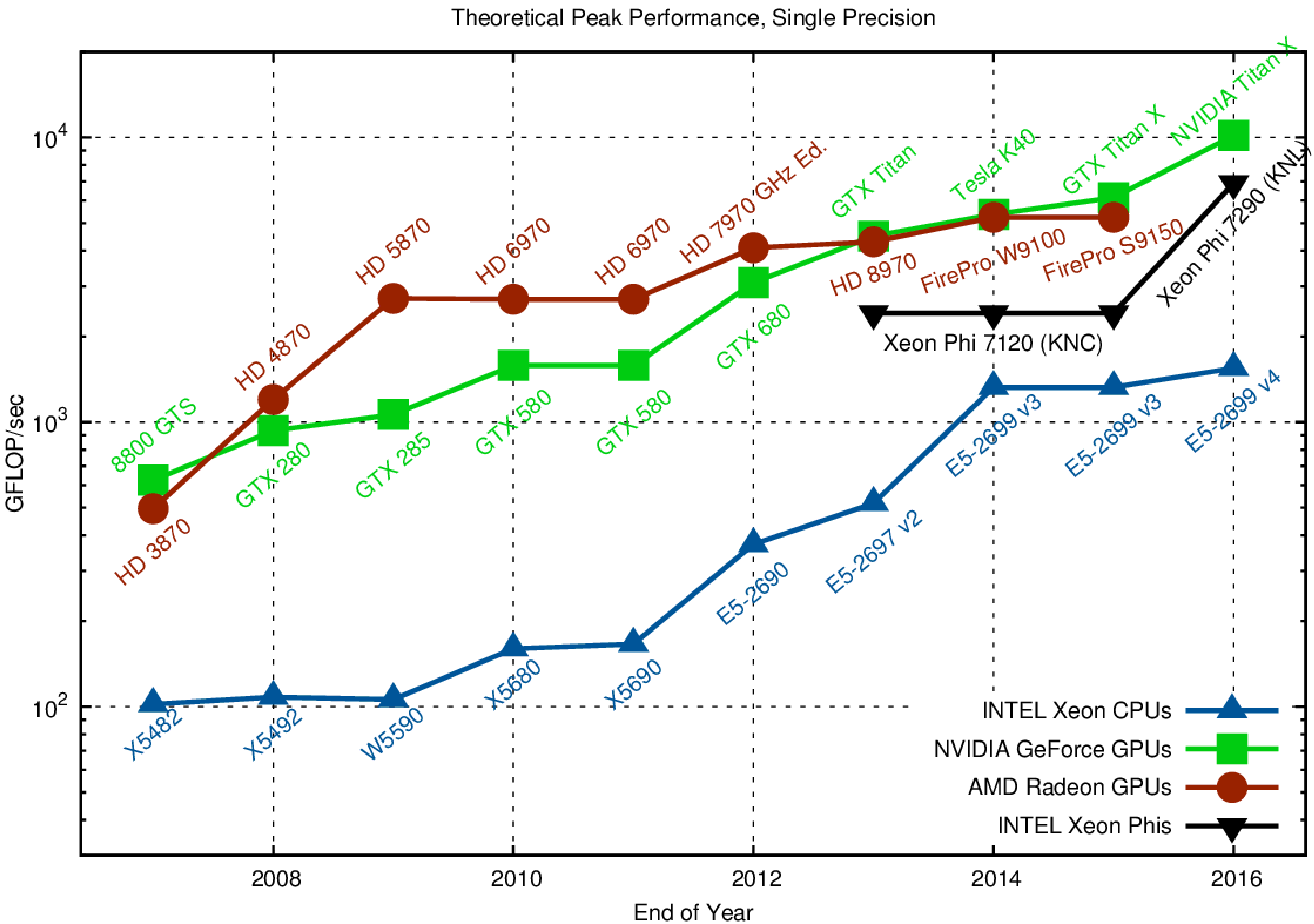
Components for Deep Learning

3 Components for Deep Learning:

- Hardware: High Computation
- Software: Deep Learning Algorithms, Libraries
- Data : Images, Videos, Annotation



Deep Learning Hardware



Deep Learning Software

Libraries (high level API)

- Caffe — (Berkeley Vision Lab)
- **TensorFlow** — (Google)
- CNTK — (Microsoft) - discontinued
- Torch — (Facebook) - discontinued
  - **PyTorch** — (Facebook/Meta)
- Theano — (MILA) – discontinued
- MXNet – Apache Software Foundation
- built on top of other libraries:
  - **Keras** — (Individual initiative + Google push)

Networks/Architectures

A neural network consisting of convolutional or recurrent layers or both, which extracts features from an image/video.

- VGG16, Alexnet,
- Siamese, Hourglass Network, VAE, [coupled networks]
- RNN, GRU, LSTM
- ResNet, Inception, Inception-Resnet, DenseNet, [parallel branches, bottleneck, skip conn., residual link]
- I3D, 3DResNet, R(2+1)D, 3D-DenseNet, ResNeXt, [ST separation, channel group]
- Videos: TCN, Slow-Fast, FPN
- NAS: AssembleNet

Models/Framework

A complete end-to-end system performing a well-defined vision task

- FRCNN, Mask-RCNN; SSD, YOLO, RetinaNet (detection/segmentation),
- FCNN (Fully Convolutional, segmentation)
- GAN, U-Net, HourGlass, Diffusion M

Data : machine learning

Image DataSets - Challenges

- CIFAR10 (CIFAR100, MNIST)
  - 10 classes/ 50,000 training images/ 10,000 testing images [1998 - 2006]
- Pascal VOC
  - 20 object categories, 11.5K images, detection + segmentation [2006 - 2012]
- Image-net - ILSVRC
  - 22K categories and 15M images; (subset) 1K categories and 1.2M images [2009 – 2012]
- MS COCO
  - 90 object categories, 183 K images, detection + segmentation + keypoints [2014]
- OpenImages
  - 600 object categories, 1.7 – 10 M images, detection – weakly annotated [2018-2019]

Video DataSets

- Kinetics
  - 400-600-700 action classes, 325-650K video clips [2017-2019]
- ActivityNet-200
  - 200 action classes, 20K untrimmed videos, 31K action instances [2016]
- MSRDailyActivity3D:
  - 16 action classes, 320 video clips [2012]
- NTU RGB+D
  - 60/120 action classes, 56880/120K videos [2016/2019]
- Toyota Smarthome
  - 31/51 action classes, 16129/536 videos, 41K action instances [2019/20]

Data : machine learning

Machine Learning : Data-Driven Approach

- Collect a dataset of images and labels – expansive – to be purified
- Use Machine Learning to train a classifier [training&validation] risk of overfitting
- Evaluate/test the classifier on new unseen images [testing/inference] within distribution

Machine Learning : Few Paradigms

- supervised learning
  - Learn to map an input (data) to known labels (ground-truth), which can be discrete (classification) or continuous (regression)
  - Transfer learning: pre-training + finetuning
- unsupervised learning
  - Learn a compact representation (i.e. distribution) of the data that can be useful for other tasks, e.g. density estimation, clustering, sampling, dimension reduction,
    - but in some cases, labels can be obtained automatically, transforming an unsupervised task to supervised
  - Domain Adaptation: labels for a source domain, but no-labels for the target domain
  - Domain Generalization: life-long learning, unknown target domain (runtime)
  - Self-Supervision: a form of unsupervised learning (generic) where the data provides the supervision, normalization, regularization (add constraints, penalty)
- semi-supervised
  - Semi (partial, zero-one-few-shots) - weakly supervised (generic or ambiguous/noisy labels),
- reinforcement learning
  - learn to predict the next actions, supervised by rewards.

STARS Inria Research Team

Objective: designing vision systems for the recognition of human activities

Challenges:

- Perception of Human Activities : **robustness**
  - Long term activities (from sec to months),
  - Real-world scenarios,
  - Real-time processing with high resolution.

- Semantic Activity Recognition : **semantic gap**
  - From pixels to semantics, uncertainty management,
  - Human activities including complex interactions with many agents, vehicles, …
  - Fine grained facial expressions, rich 3D spatio-temporal relationships.

- Learning representation: **effective models**
  - Combining Multi-modalities: RGB, 2D/3D Pose, Flow, bio-signals, voice, …
  - Cross spatial and temporal dimensions : LSTM, TCN, Transformers, mamba,…
  - Using learning mechanisms: fusion, multi-tasks, guided-Attention, Self-Attention, Knowledge Distillation, contrastive learning,
  - In various learning modes : supervised, weakly-supervised, cross-datasets, unsupervised, self-learning, life long learning

- **Applications** : Safety & Health (CoBTeK from Nice Hospital : Behavior Disorder)

Where to find more course material

- Course Website:
  - http://www-sop.inria.fr/members/Francois.Bremond/MSclass/deepLearningWinterSchool23/UCA_master/index.html
  - Syllabus, lecture slides, schedule, videos, etc

- Emails:
  - Tomasz Stanczyk: tomasz.stanczyk@inria.fr
  - Valeriya Strizhkova: valeriya.strizhkova@inria.fr
  - Snehashis Majhi : snehashis.majhi@inria.fr
  - Francois Bremond: francois.bremond@inria.fr




Image Classification

Artificial Neuron

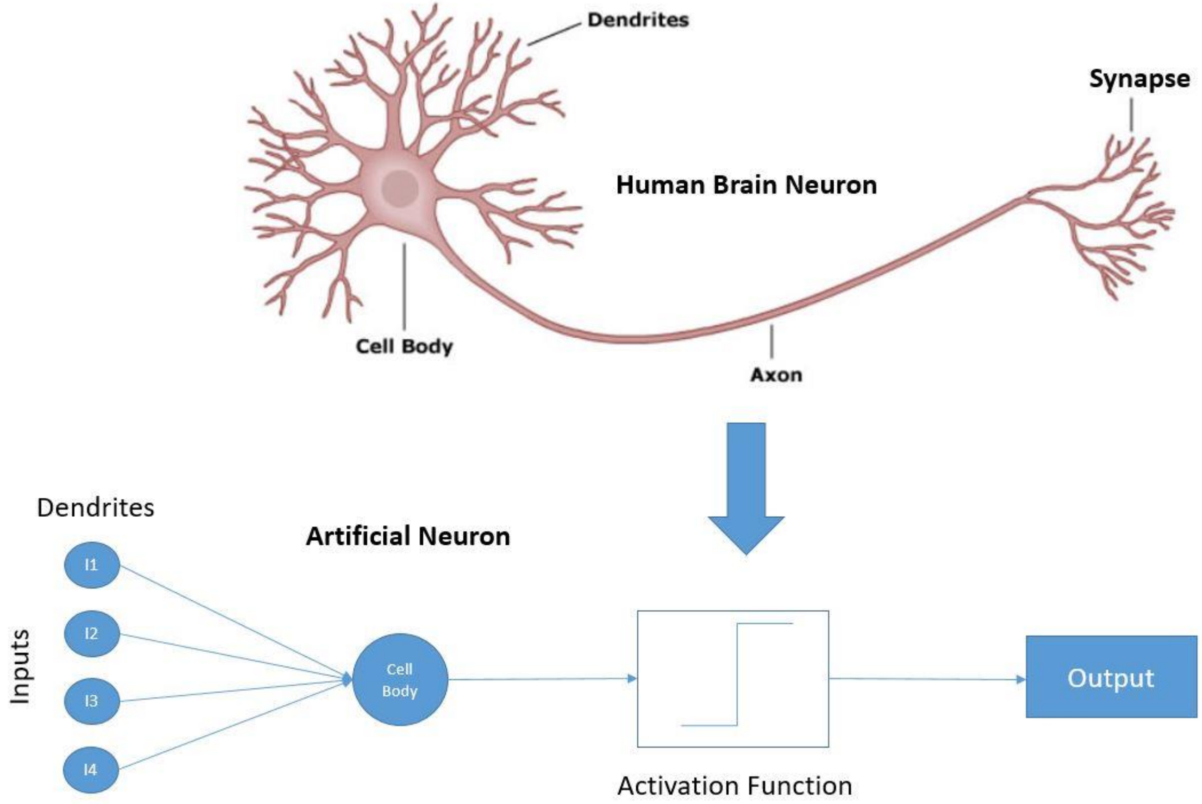
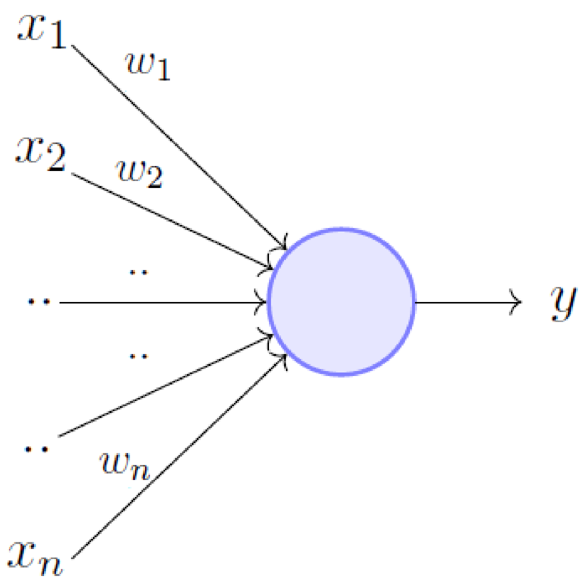


Image Classification



$$y = 1 \quad if \sum_{i=1}^{n} w_i * x_i \geq \theta$$
$$= 0 \quad if \sum_{i=1}^{n} w_i * x_i < \theta$$

Rewriting the above,

$$y = 1 \quad if \sum_{i=1}^{n} w_i * x_i - \theta \geq 0$$
$$= 0 \quad if \sum_{i=1}^{n} w_i * x_i - \theta < 0$$

θ is the activation threshold.

Image Classification

In the previous example, the **activation function** is binary step function (also called Heaviside)

$$\sigma(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

Because it's not continuous at 0, in practice, we usually **use sigmoid function** (also called logistic)

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

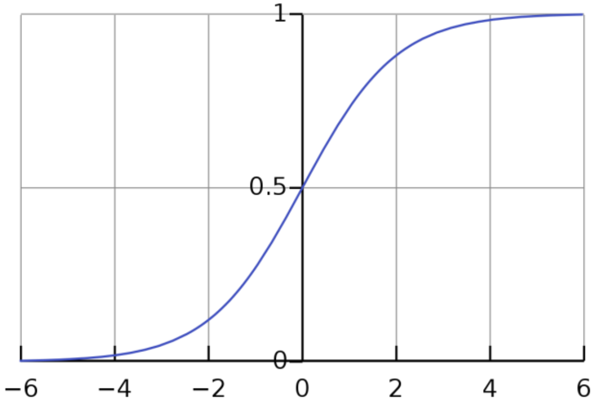
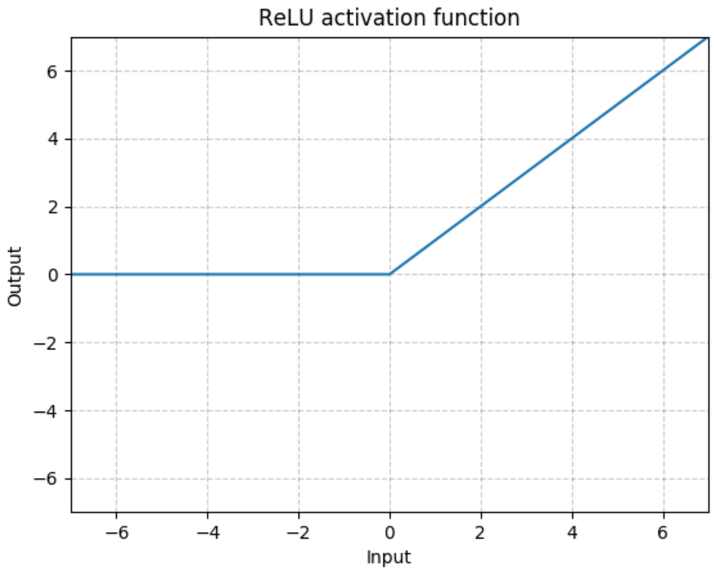


Image Classification

Activation Functions: ReLU (Rectified Linear Unit)

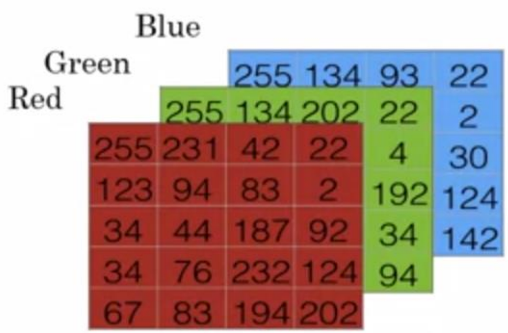
$$ReLU(x) = max(0, x)$$



Range from 0 to infinity, which keeps high activation.

Image Classification

Representation: image matrix



For image # i

$$y^{(i)} \in \{0, 1\}$$

$$x^{(i)} = \begin{bmatrix} 255 \\ 231 \\ 42 \\ \vdots \\ 142 \end{bmatrix}$$

For m training samples

$$\mathbf{y} = \begin{bmatrix} y^{(1)} & \dots & y^{(m)} \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} x^{(1)} & \dots & x^{(m)} \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,m} \end{bmatrix}$$

Image Classification

Formalization:

1. Each neuron can be regarded as a **linear function**.

$$\mathbf{z} = \omega^{\mathbf{T}}\mathbf{x} + b$$

2. Activation function make it possible to learn **non-linear complex** functional mappings. We introduce non-linear properties to the Network, which is important for solving complex visual tasks.

$$\mathbf{y} = \sigma(\mathbf{z}) = \sigma(\omega^{\mathbf{T}}\mathbf{x} + b)$$

Image Classification

Logistic Regression with Cost Function

From previous slide, we have a **logistic** regression model.

$$\hat{\mathbf{y}} = \sigma(\omega^{\mathbf{T}}\mathbf{x} + b), where\ \sigma(x) = \frac{1}{1+e^{-x}}$$

Given **m** samples,

$$\mathbf{x} = \begin{bmatrix} x^{(1)} & \dots & x^{(m)} \end{bmatrix}$$

$$\mathbf{y} = \begin{bmatrix} y^{(1)} & \dots & y^{(m)} \end{bmatrix}$$

We want to minimize the distance between the **prediction** and the **ground truth**,

$$\hat{y}^{(i)} \approx y^{(i)}$$

Image Classification

Logistic Regression with Cost Function

Define the distance with a loss function, for example, one half a **square** error

$$L(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$$

However, people don't usually use this to learn parameters. In practice, we usually use the **cross entropy** loss to maximize the likelihood of classifying the input data correctly.

$$L(\hat{y}, y) = -(y \log \hat{y} + (1 - y) \log (1 - \hat{y}))$$

The cost function on the m samples will be

$$J(\omega, b) = \frac{1}{m} \sum_{i=1}^{m} L(\hat{y}, y)$$

Image Classification

Cross Entropy

Let's look deeper into the **cross entropy** loss.

$$L(\hat{y}, y) = \begin{cases} -log(1-\hat{y}),\ if\ y=0 \\ -log\hat{y},\ if\ y=1 \end{cases}$$

When your **prediction** get further to the true label, your **loss** will grow exponentially.

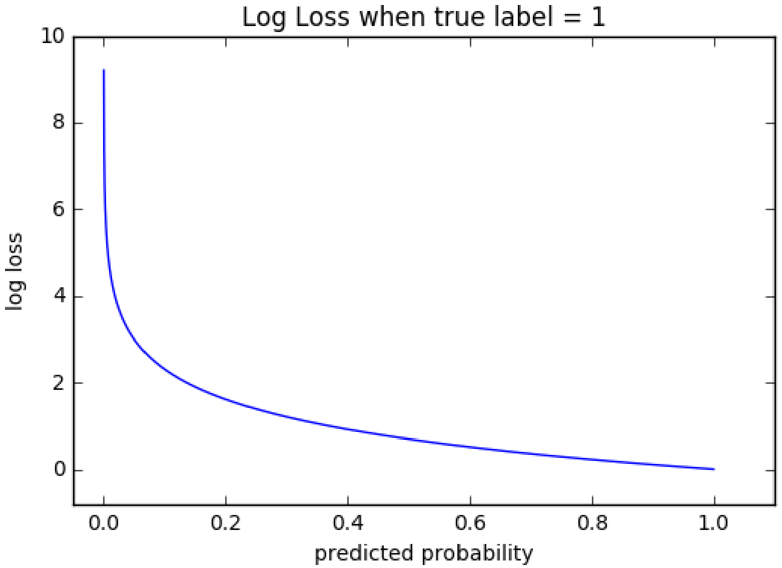
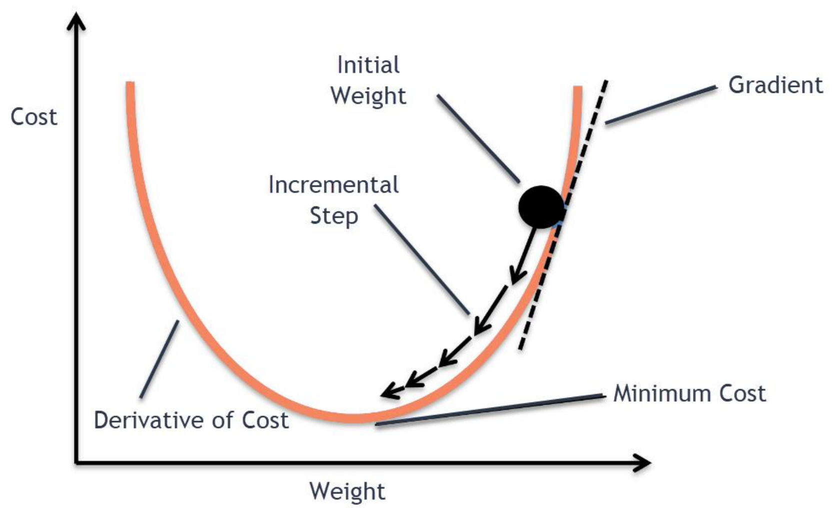


Image Classification

Gradient Descent

Our objective is to find ω, b that minimize $$J(\omega, b) = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}, y)$$



Repeat $$\begin{cases} \omega = \omega - \alpha \frac{\partial J(\omega,b)}{\partial \omega} \\ b = b - \alpha \frac{\partial J(\omega,b)}{\partial b} \end{cases}$$

Reminder: $\frac{\partial J(\omega,b)}{\partial \omega}$ is the partial derivative of the **cost** with respect to **ω**, which gives the slope of the tangent line to the graph of the function at that point.

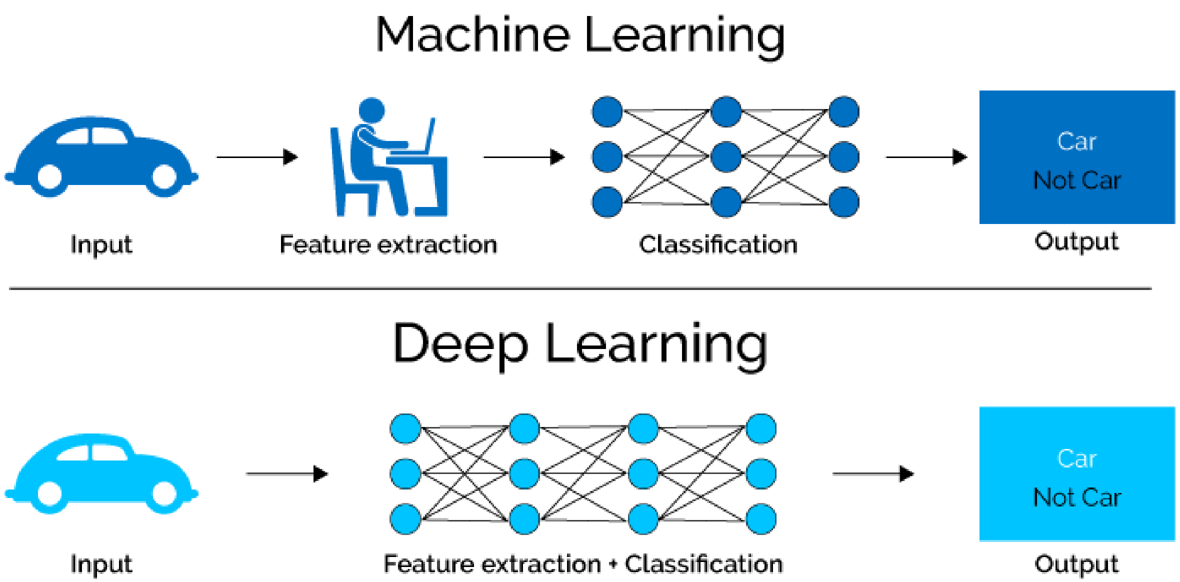
Image Classification

Desiging a classifier is a 2 steps process:

1. An artificial neuron is composed by a **linear transformation** and an **activation function**.
2. To adjust parameters in an artificial neuron, we need to define a **cost/loss function**. By decreasing the cost function with gradient descent, the parameters get updated step by step.

Image Classification

Difference between machine learning and deep learning

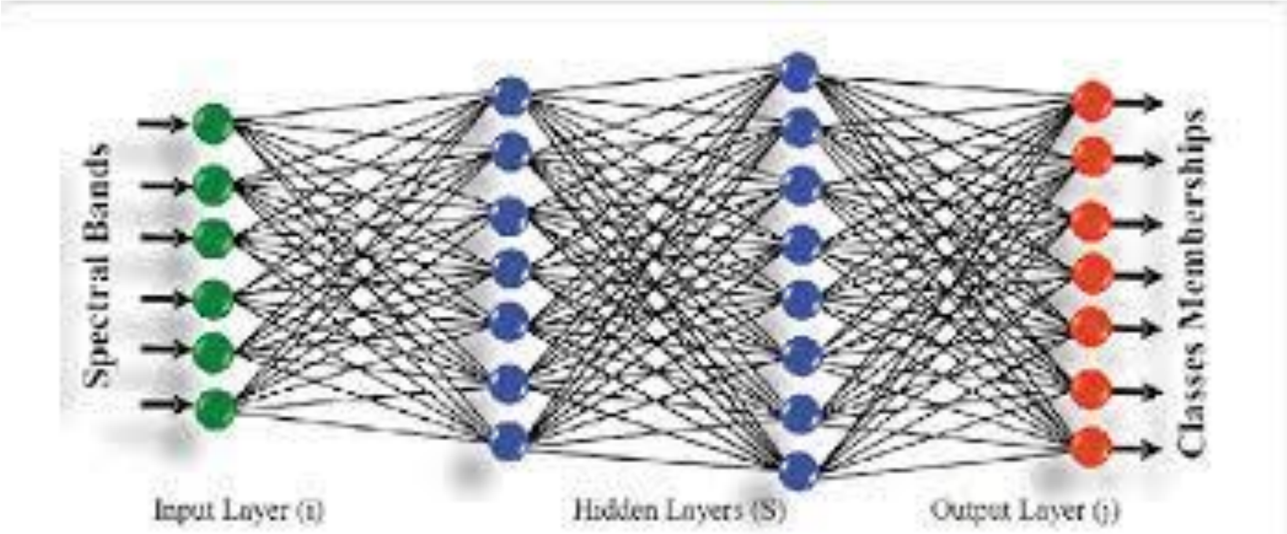


Traditional machine learning methods usually work on hand-crafted features (texture, geometry, intensity features ...).

Deep learning methods combine hand designed feature extraction and classification steps.

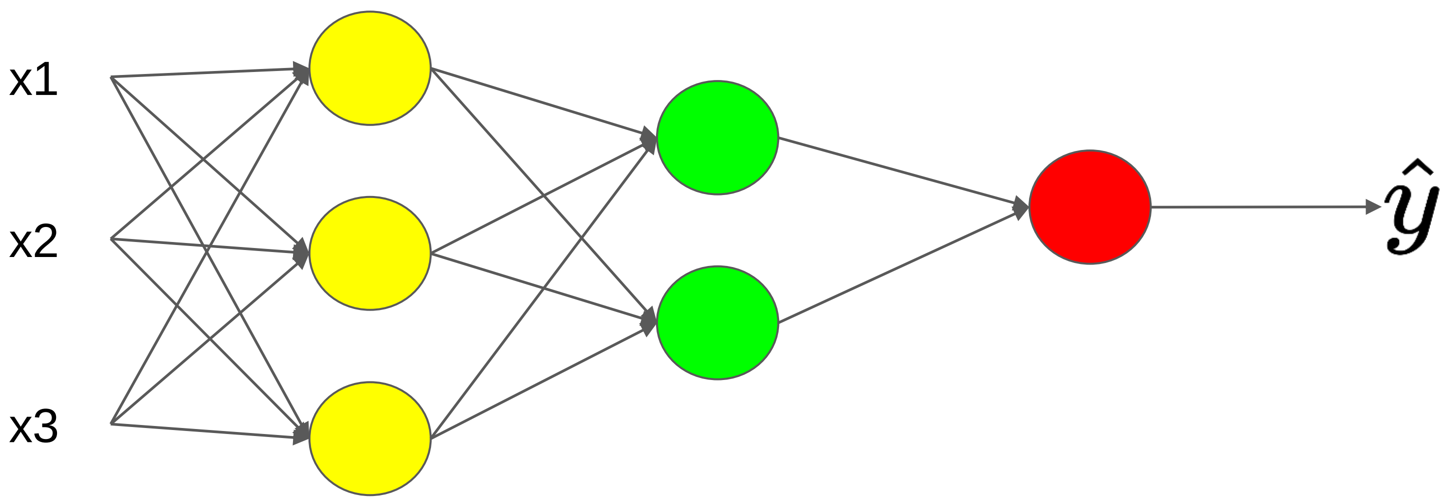
This is also called “end-to-end model”.

MultiLayer Perceptron (MLP) Network with fully-connected (FC) layers



FC: Image Classification

Deeper neural network



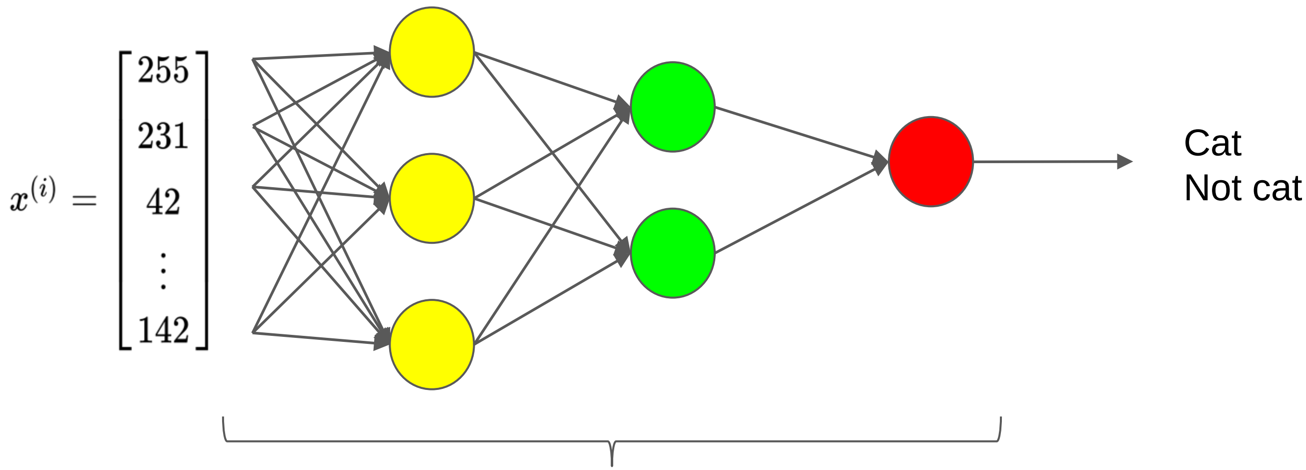
$$\boxed{\mathbf{a}^{[\mathbf{0}]} = \sigma(\omega^{[\mathbf{0}]}\mathbf{x} + b^{[0]})} \rightarrow \boxed{\mathbf{a}^{[\mathbf{1}]} = \sigma(\omega^{[\mathbf{1}]}\mathbf{a}^{[\mathbf{0}]} + b^{[1]})} \rightarrow \boxed{\hat{\mathbf{y}} = \sigma(\omega^{[\mathbf{2}]}\mathbf{a}^{[\mathbf{1}]} + b^{[2]})}$$

Parameters get updated layer by layer via back-propagation.
These are **fully-connected (FC)** layers

Convolutional Neural Network (CNN)

CNN: Image Classification

Fully-connected layers

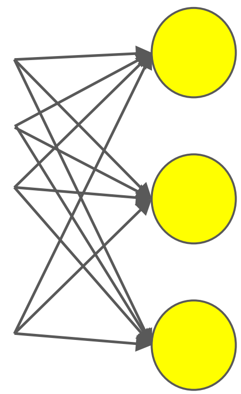


From previous slides, we can see fully-connected (FC) layers connect every neuron in one layer to every neuron in the previous layer.

CNN: Image Classification

Drawback of fully-connected layer

$$x^{(i)} = \underbrace{\begin{bmatrix} 255 \\ 231 \\ 42 \\ \vdots \\ 142 \end{bmatrix}}_{3N^2}$$



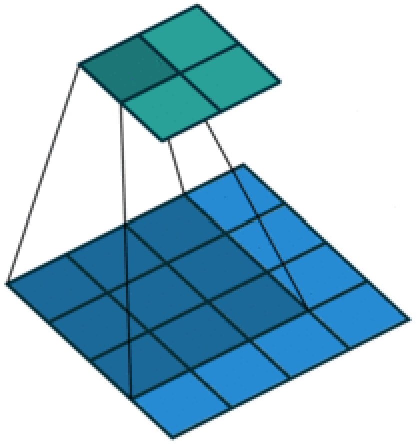
x.shape=(3N*N,1)   w.shape=(3N*N,3)

- For low-quality image, e.g. N=100, w.shape=(30K,3), it’s ok.
- But for high-quality image, e.g. N=1K, w.shape=(3M,3), much more computational resources will be needed.

CNN: Image Classification

Convolution

Instead of connecting to every neuron in the previous layer, a neuron in the convolutional layer only connects to neurons within a small region.



Advantages:

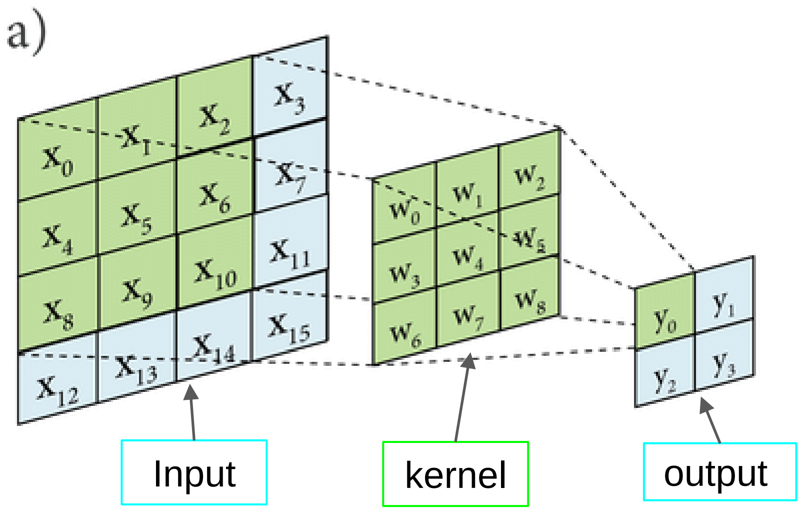
1. Spatial coherence is kept.
2. Lower computational complexity.

CNN: Image Classification

Convolution

We don’t have to flatten the input, so the **spatial coherence** is kept.

A **kernel** (also called **filter**) slides across the input feature map. At each location, the product between each element of the kernel and the input element is computed and summed up as the output in the current location.



CNN: Image Classification

3D volumes of neurons

A **convolutional layer** has neurons arranged in 3 dimensions:

- Height
- Width
- Depth (also called **channel**)

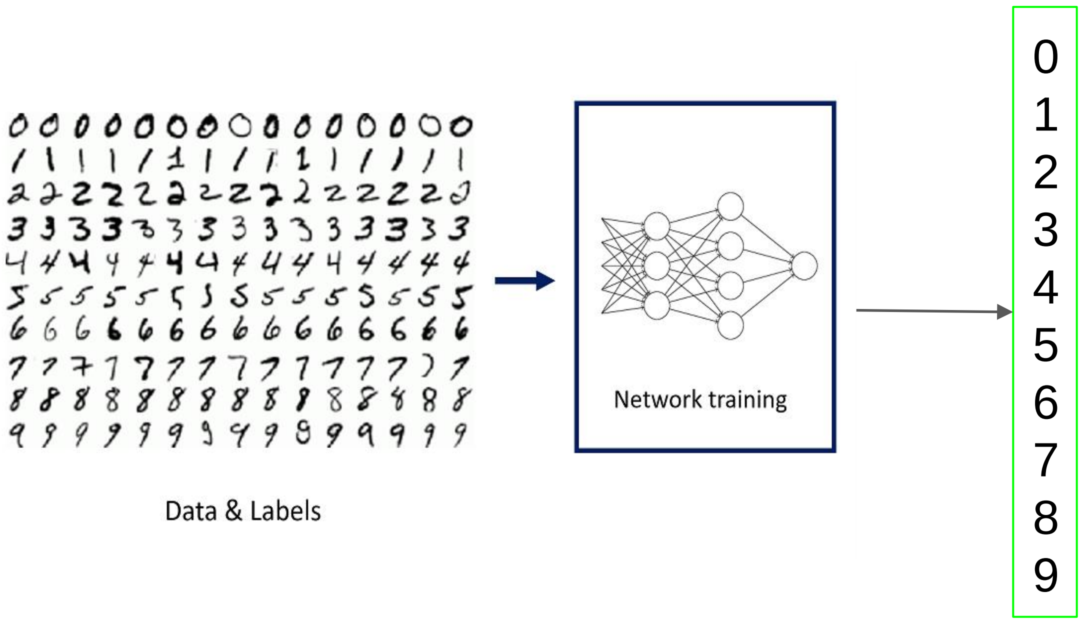
The initial **depth** of a RGB image is 3. For example, in CIFAR-10, images are of size 32*32*3 (32 wide, 32 high, 3 color channels).

In this case, the kernel has to be 3-dimensional. It will slide across the height, width and depth of the input feature map.

CNN: Image Classification

CNN example

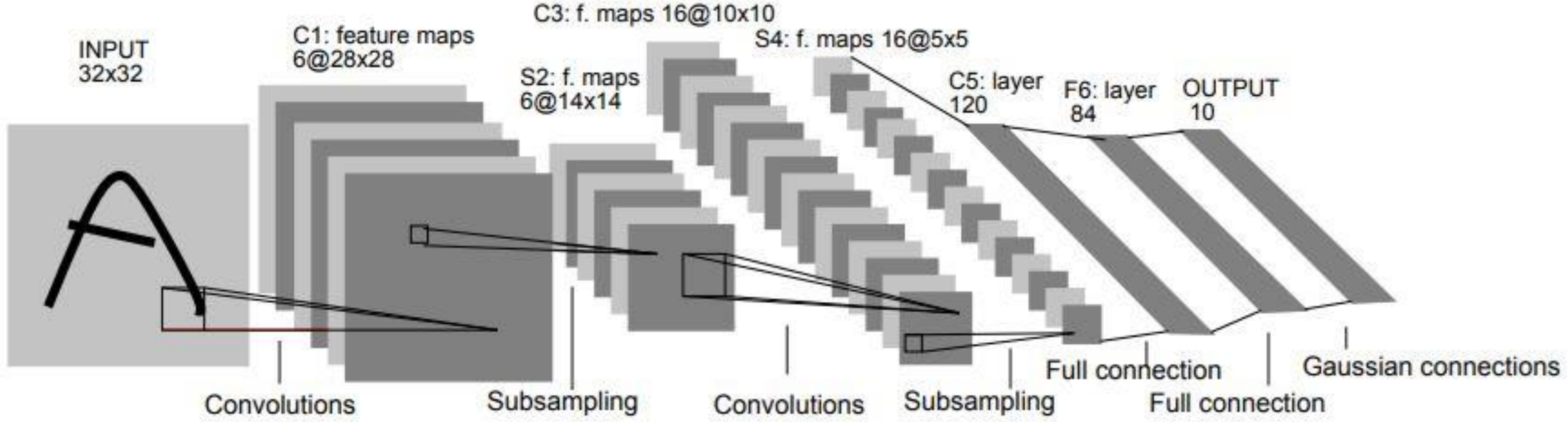
LeNet-5 [1] is proposed by Yann LeCun, Leon Bottou, Yoshua Bengio and Patrick Haffner in 1990’s for handwritten and machine-printed character recognition.



[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, 86(11):2278-2324, November 1998.

CNN: Image Classification

LeNet-5

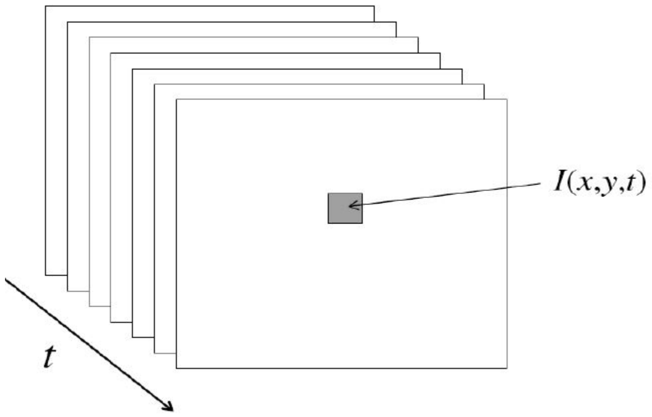


In LeNet-5, subsampling operation corresponds to an average pooling. Basically, LeNet-5 is a combination of convolution, pooling and fully-connected layers.

Video Classification

Video Representation:

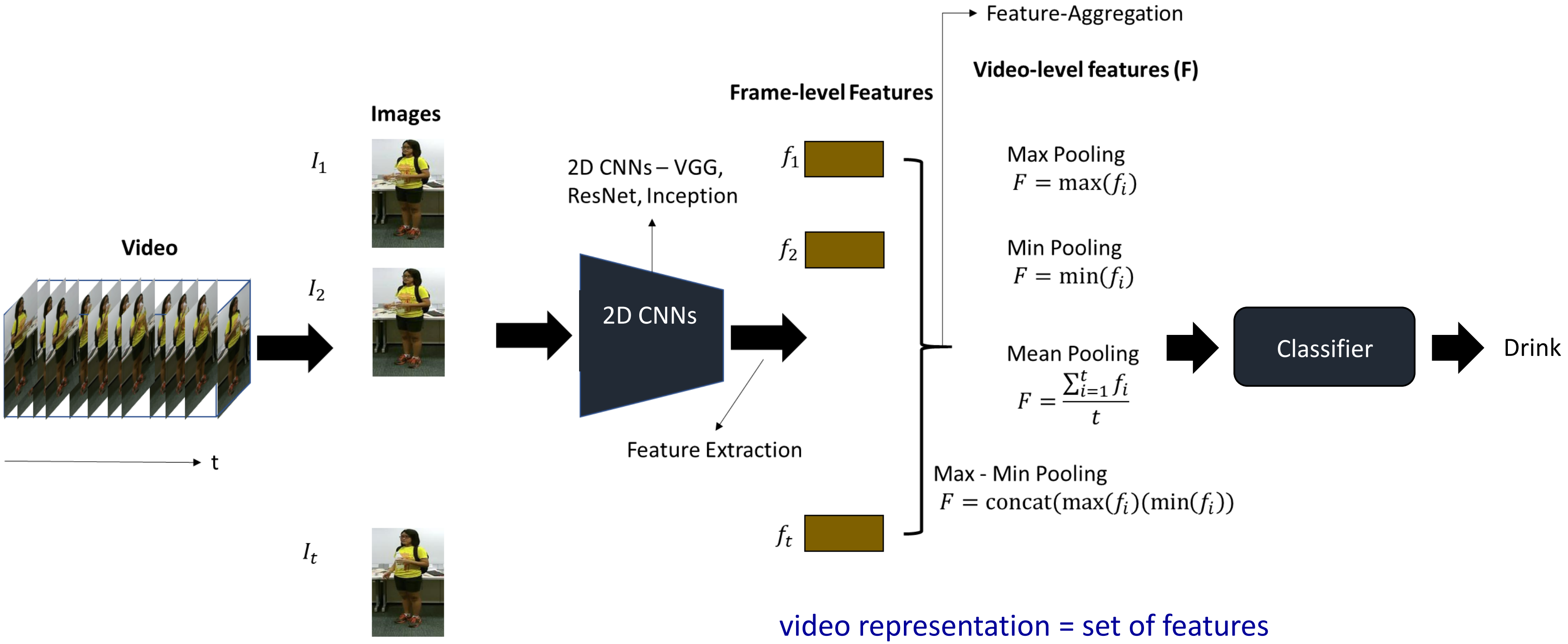
- Formally, a **video is a 3D signal** with:
  - **Spatial Coordinates:** x, y
  - **Temporal Coordinates:** t



- A video can be seen as a **sequence of Images/Frames.**

Video Classification

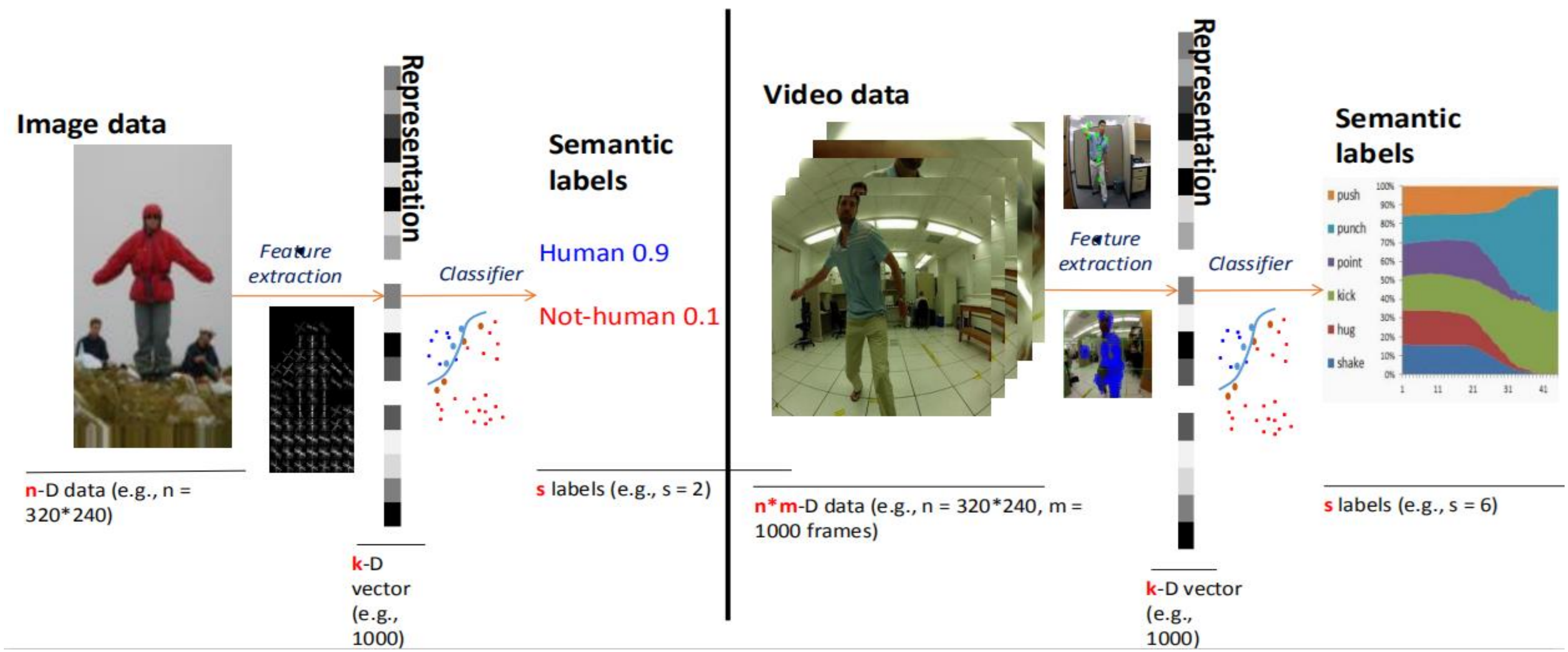
2D CNN: feature extraction + classification



video representation = set of features

Video Classification

Image versus Video Classification :



People Detection in real world situations

People Tracking in real world situations

People Tracking in real world situations

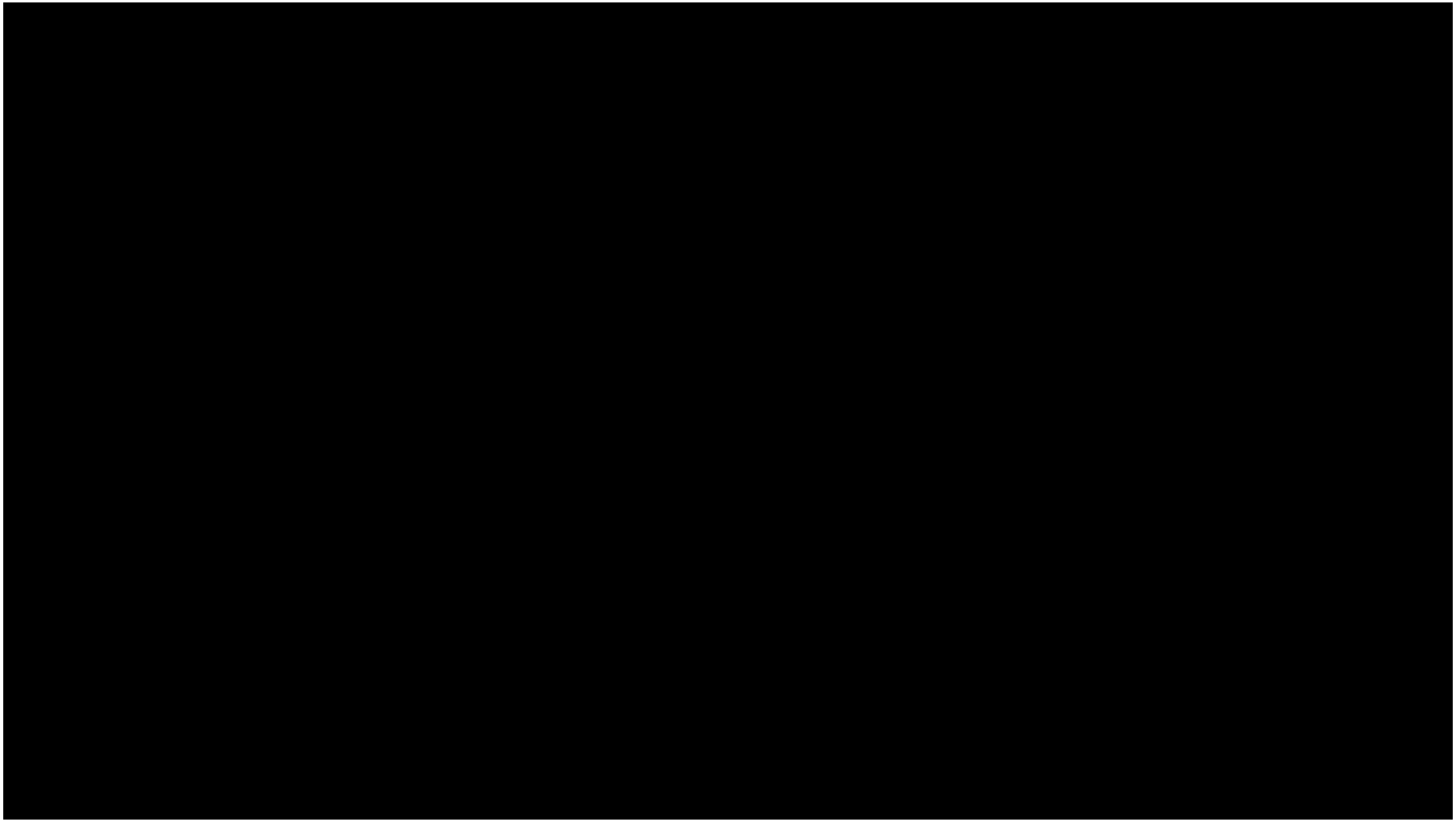
People Tracking and Segmentation on MOT

Grounded DINO + Segment Anything (SAM) + Track Anything

Analysis of trichogramma behavior with video tracking

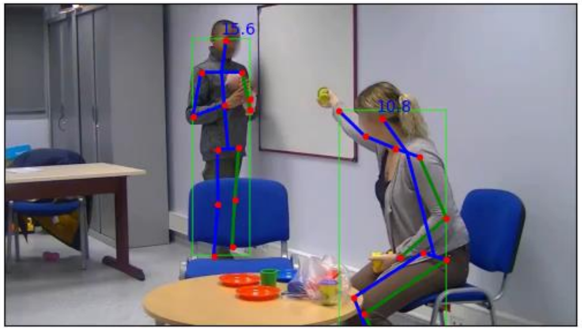
Activity monitoring at ICP with AD patients

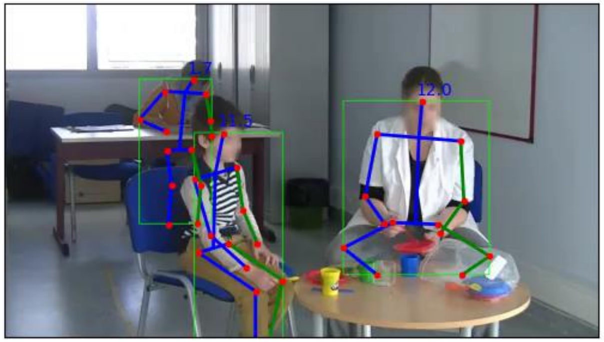
Visualization of older adult performance while accomplishing the semi-guided tasks.



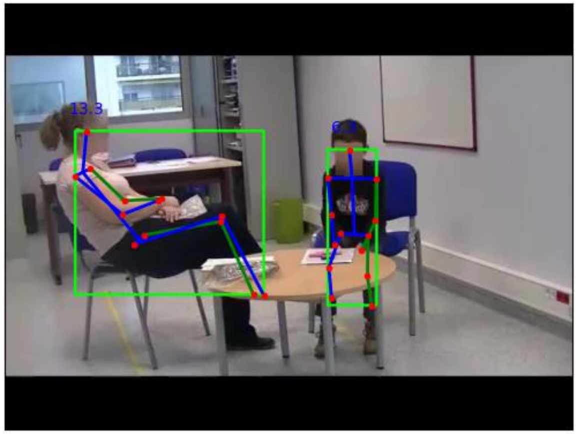
ACt4autism: children behavior

Objective quantification of atypical behaviors (stereotypies) on which the diagnosis of autism (ADOS) is based.





- Analysis of the atypical postures of the child with ASD.
- Global analysis of the movements of the child with ASD with agitation.
- Eye tracker analysis to measure joint attention.



Toyota Smart-Home
Large scale daily living dataset

Example 1

Challenges :

1. Composite Activities
e.g. Cook

3. Low Camera Framing
e.g. Dump in Trash

Person 02

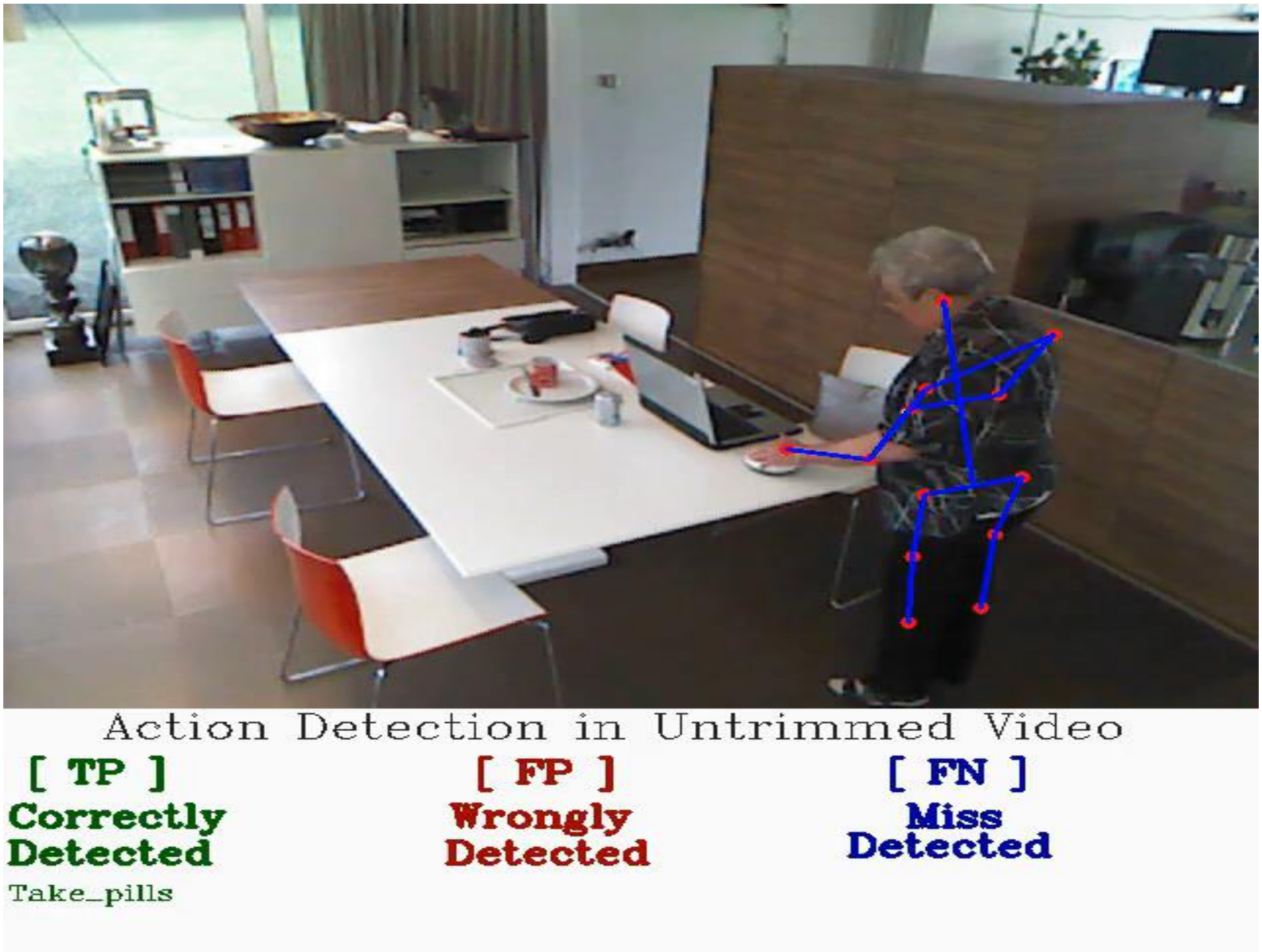
Camera 03

Frame 2379

Annotated Activities By Category

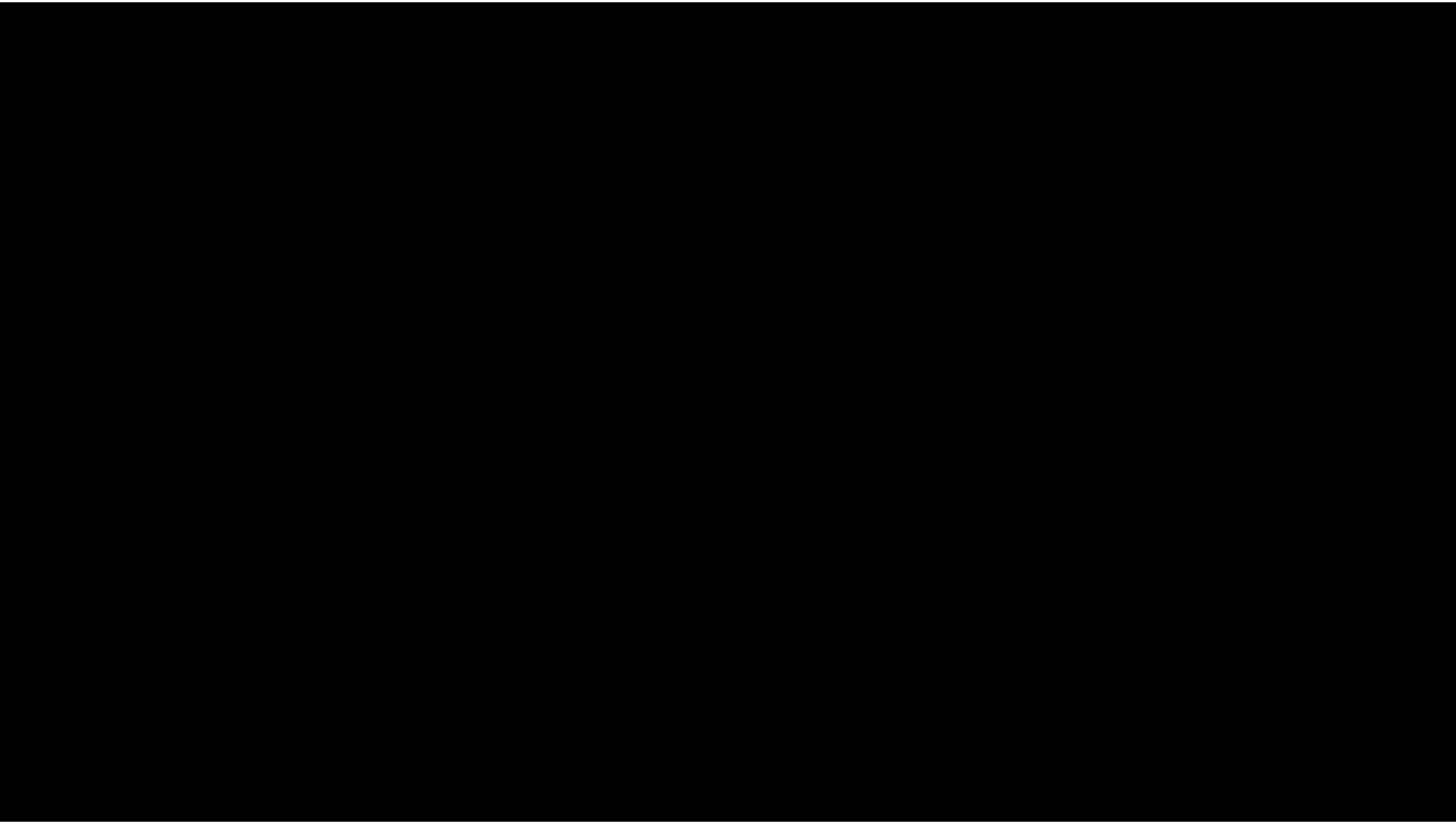
Single	**Composite & Elementary**	**Object–based**
Take_sth._off_table Walk	Cook	

Toyota Smart-Home
Large scale daily living dataset



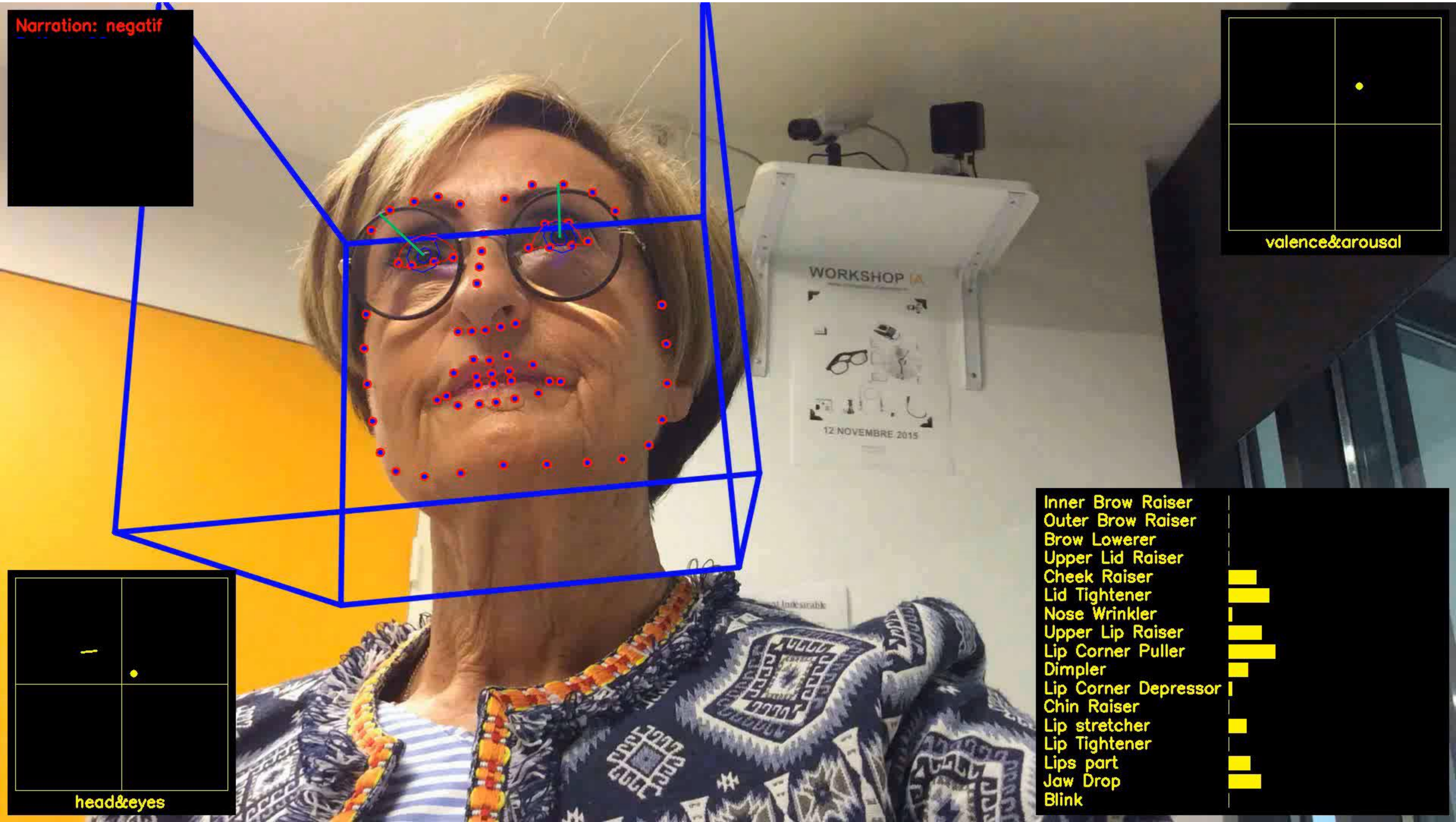
Praxis and Gesture Recognition

(short demo)



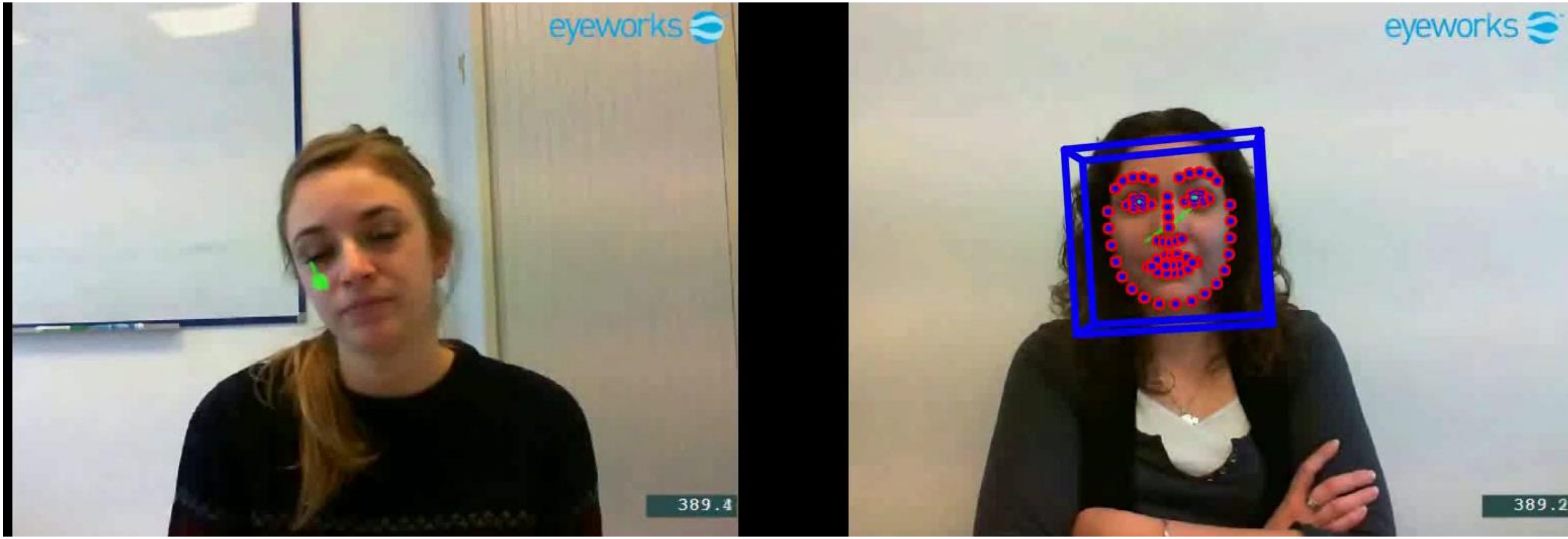
Emotion Recognition : Facial Expression Recognition

Characterizing the state of Apathy using Facial Motion and Emotion



Emotion Recognition : gaze estimation

Characterization of gaze (attention) during speech: case of schizophrenia (rupture of content).



Green dot: eye tracker

Video generation to increase facial expressions