FRUGAL AI PROJECT

US and French universities and companies collaboration

Project presentation - mai 17th 2023

AMBASSADE DE FRANCE AUX ÉTATS-UNIS
Liberté
Égalité
Fraternité
Service pour la Science et la Technologie
Office for Science and Technology

datacraft*

Annabelle Blangero
Ekimetrics

Marianne Clausel
Université de Lorraine

Walid Erray
Groupe Crédit Agricole

Denis Marraud
Airbus Defence & Space

Emeric Tonnelier
Groupe Crédit Agricole

Romain Godet
datacraft

Xavier Lioneton
datacraft

datacraft Frugal AI project?

What is the ambition?

Develop a joint applied research project on Frugal AI within datacraft and in collaboration with **US companies and universities** that will benefit to all.

Activities

- Conferences, workshops (held in France and US)
- Research collaborations
- Researchers and students exchanges
- Learning expeditions (both in US and France)
- ..

Deliverables

- State-of-the-art tools and methods sharing
- Tools and recommendations for companies
- Awareness-raising actions for companies
- Communication tools available for scholars
- Publications
- ...

Who is involved so far?

Core team

Companies

- Airbus Defence and Space
- Crédit Agricole
- Ecolab/Ministry of Ecological Transition
- Ekimetrics
- FDJ

Researchers

- CentraleSupélec
- Université de Lorraine
- Université Grenoble Alpes

in collaboration with the Science and Technology Service (SST) of the ***French Embassy in the United States****, and in particular with the French Consulate in San Francisco.*

US Partner

In advance discussion with Berkeley
Contact to be taken with Stanford, Seattle, and US companies

French Embassy is in lead to identify and discuss with interested universities and companies on frugal AI to join the project

- cross-fertilization between France and US, workshops, online discussion for sharing the results, learning expedition

A - What is Frugal AI?

1- Context
2- What we will be talking about?
3- Use cases examples

B - Scope of Frugal AI project & State-of-the-art overview

1- Handle low volumes of data
2- Minimize required volume of annotations
3- Reduce environmental impact
4- Focus on measuring tools

C - Next steps!

What's Frugal AI?

Context

Impressive performance leaps of Deep Learning based capabilities

from Artzone ai blog

Segment Anything Model (SAM) - Meta

REQUIRING…

huge volumes of data → … not always existing or accessible

supervised annotations → … long & tedious work / precarious jobs / biases

huge compute infrastructures → …not environment friendly

NEED FOR MORE EFFICIENT, LESS DATA & COMPUTE HUNGRY LEARNING & INFERENCE METHODS

datacraft*

Context

the 4 components in the design and operation of AI solutions
Crédit Agricole Group

Data

- **Evaluate** and **optimize** databases, **Compress** and **serialize**
- **Restrict perimeters** : limit transfer and storage
- Use **only necessary data:**
  - when **building** the AI solution
  - when **operating** the AI solution

Algorithmes

- Capitalize on **existing models**
- **Avoid "brute-force"** methods and start with **sample**
- Capitalize on **existing use cases** through **meta-learning**
  - Average **80% reduction in computation time**
  - keep statistical **performance stable**

software

- Optimize performance by using **predefined templates**
- Using **up-to-date AI frameworks** and packages, etc.
- Optimize technical and application choices to **reduce network traffic**, **communication protocol** & **data compression**

Infrastructure

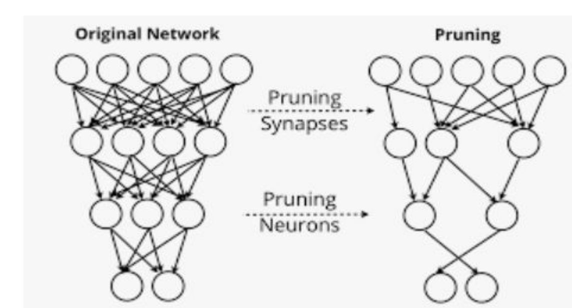
- **Measure** and **monitor** energy **consumption**
- **Optimize the sizing of infrastructures** to minimize the resources
- **Plan** the different jobs **to optimize** the use of the **infrastructure** over time and do **more with less**

- **Global action plan that integrates the 4 components**
- **Integrating the carbon footprint as a quality criterion for an AI Solution**

Frugal AI, what we will be talking about?

FRUGAL METHODS

- AT TRAINING TIME
  - MANAGE LOW VOLUMES OF DATA
  - MINIMIZE REQUIRED VOLUMES OF ANNOTATIONS
  - REDUCE ENVIRONMENTAL IMPACT
- AT INFERENCE TIME
  - MANAGE LIMITED TARGET HW PERFORMANCES
  - REDUCE INFERENCE COSTS
  - REDUCE ENVIRONMENTAL IMPACT



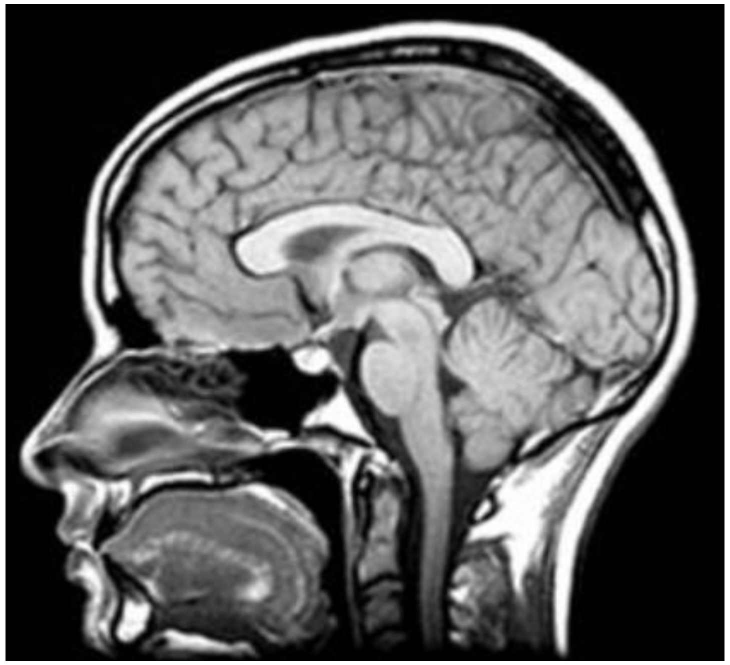
VAE Zero shot Few shot	Simulation Self, weakly supervised Active Learning	Dataset distillation	Model compression Model distillation

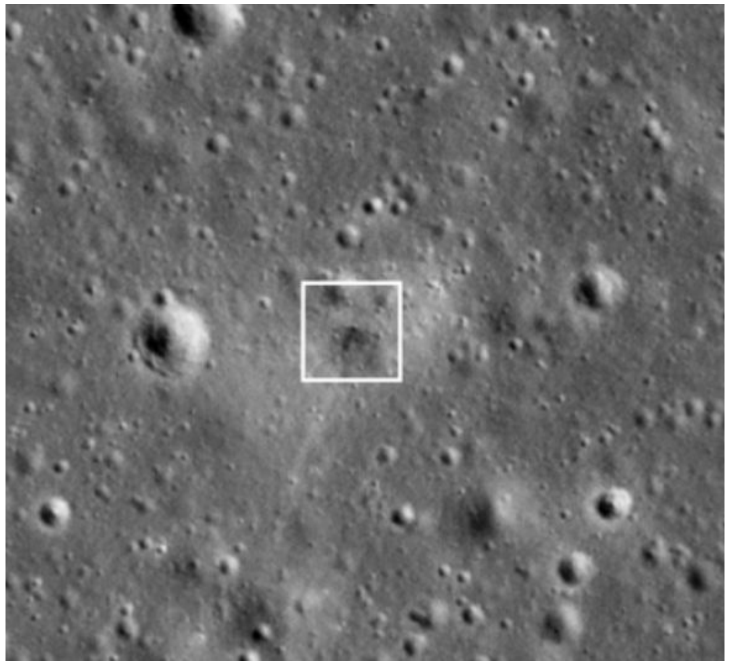
datacraft*

Use cases examples: low data volume

RARE OBJECTS
RARE EVENTS

SENSITIVE DATA
DIFFICULT TO ACCESS DATA

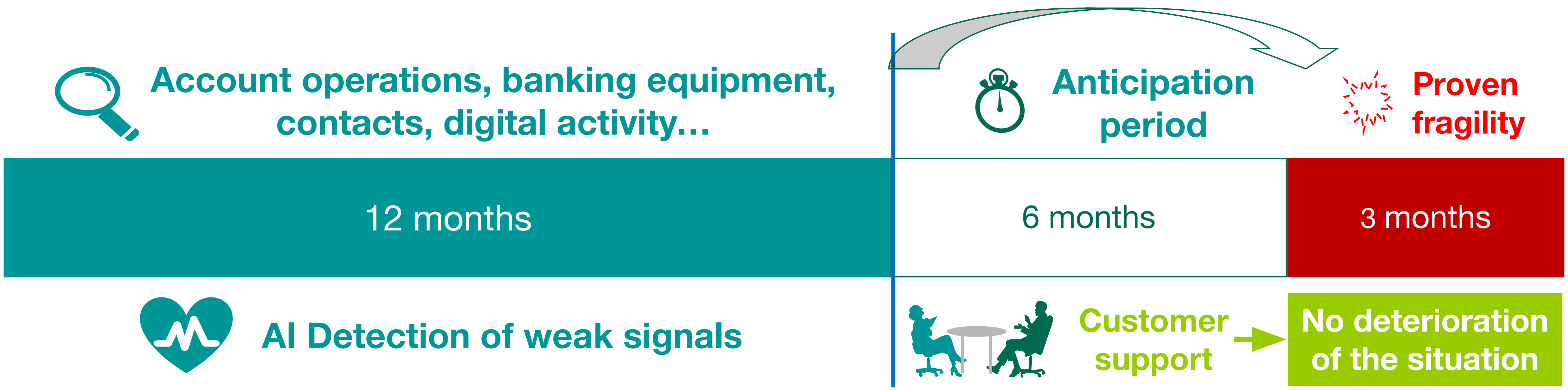




How to efficiently detect anomalies / objects rarely observed?

Use cases examples: low* data volume

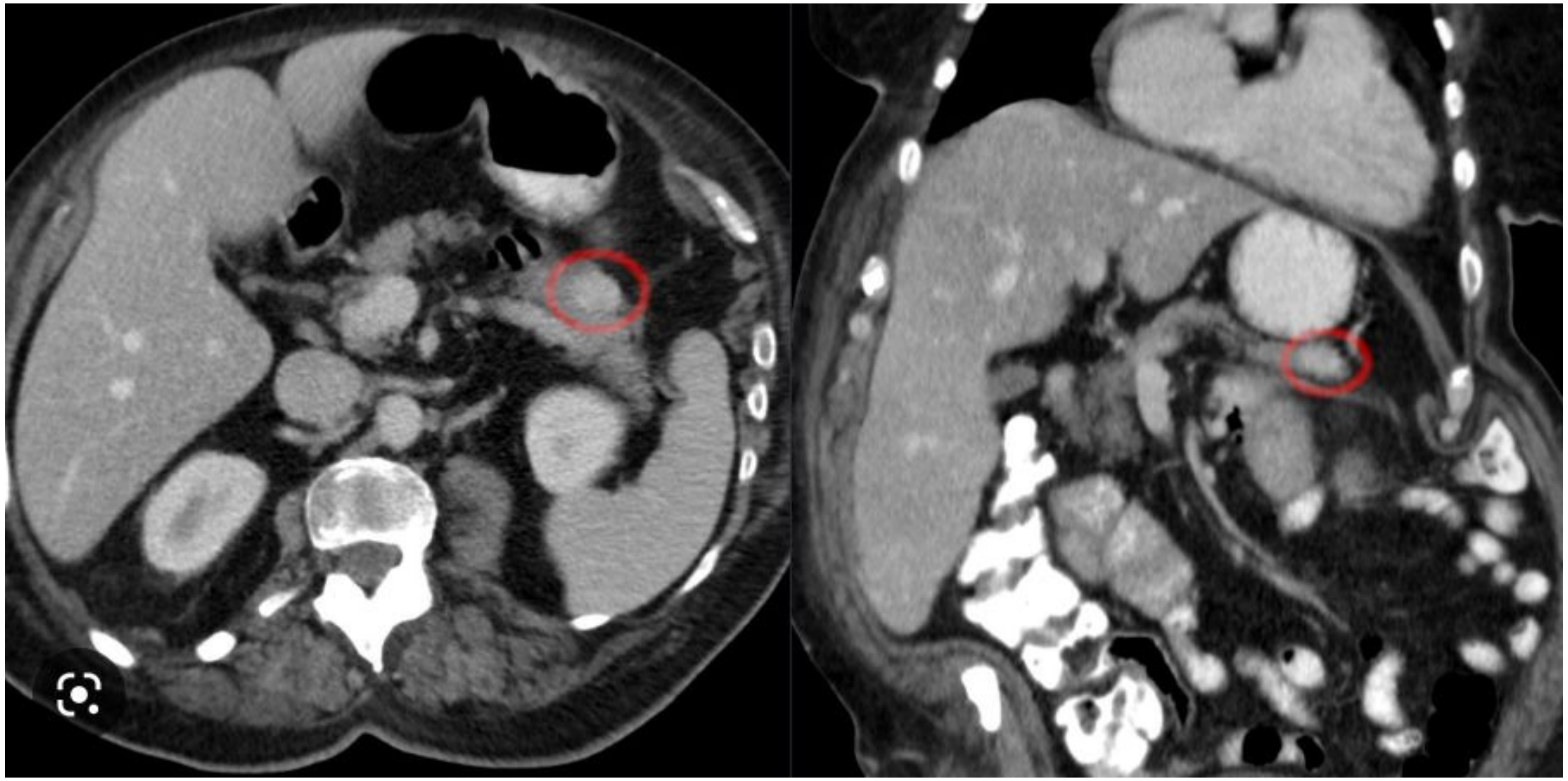
Early Detection of Customers in a Situation of Financial Fragility
Crédit Agricole Group



- **Large overall volume of data (hundreds of GB)**
- **Heterogeneous data (tabular data, time series, logs…)**
- ***AI models based on a low % of positive observations**

Use cases examples: minimize annotations

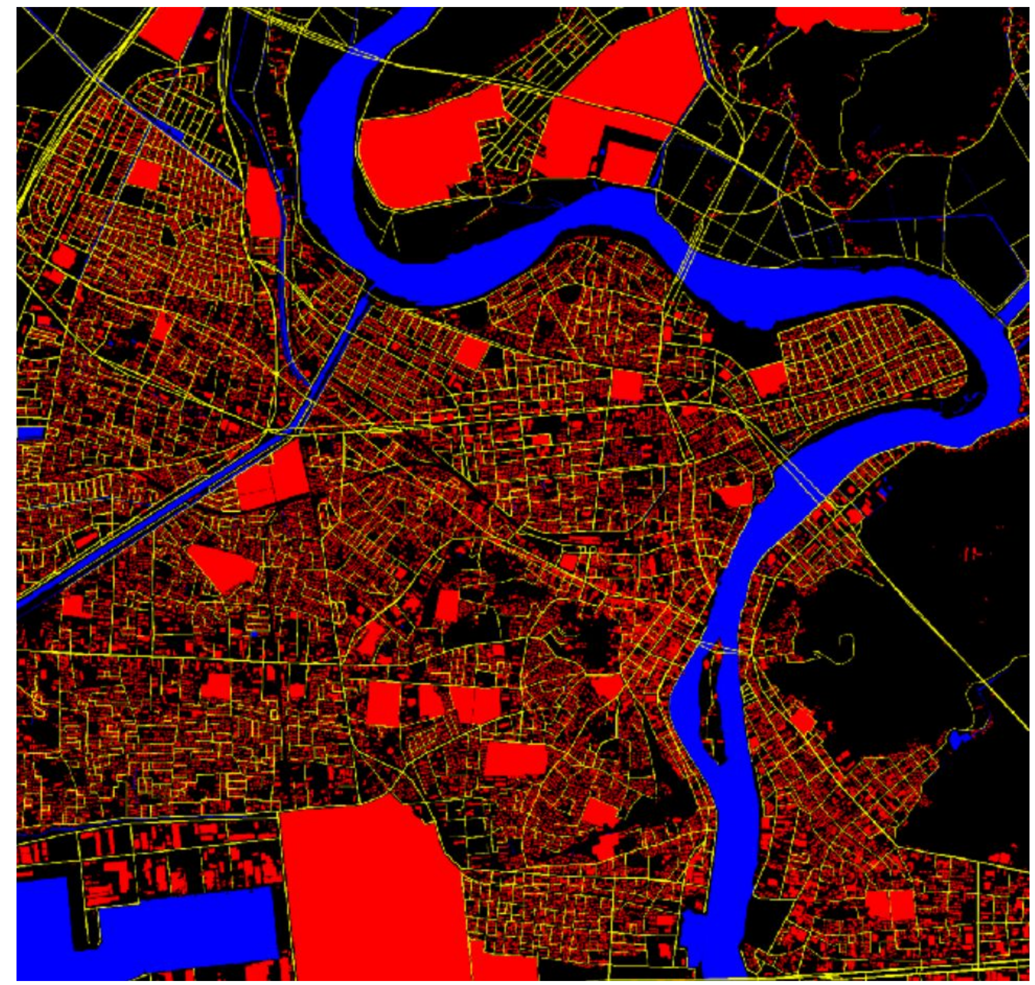
COSTLY ANNOTATIONS RELYING ON
SUBJECT MATTER EXPERTS



How to minimize the time spent by experts to annotate complex phenomena?

Use cases examples: minimize annotations

AIRBUS EXAMPLE: SAT IMAGE SEGMENTATION / CHANGE DETECTION
COSTLY / TEDIOUS DATA ANNOTATION

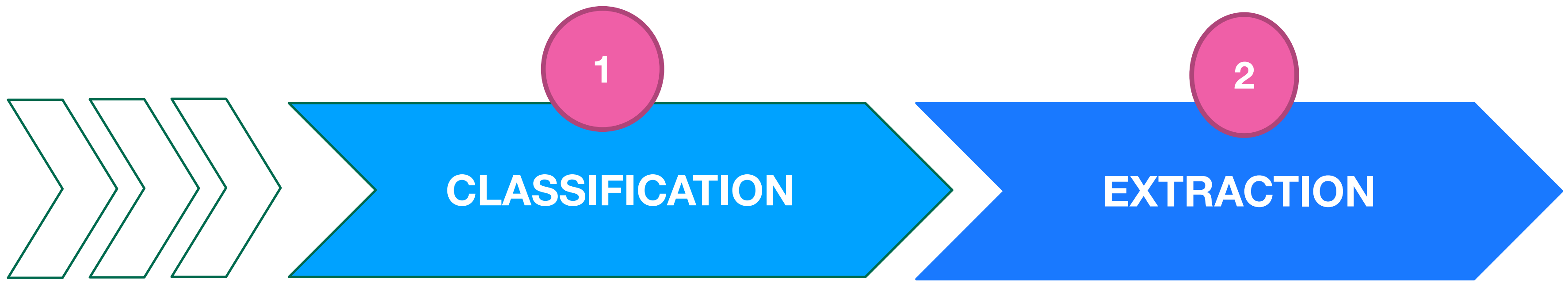


DOMAIN ADAPTATION/EXTENSION:
how to adapt/extend a capability from an initial domain to another/ a larger domain with minimum inputs?

DOMAIN SPECIALIZATION (transductive learning):
How to rapidly optimize an existing generic capability on a restricted domain with minimum inputs?

Use cases examples: minimize annotations

Document classification and information extraction
Crédit Agricole Group



Example

Annotation

- ❑ **Tax notice**
- ❑ **Salary slips**
- ❑ **National Identity Card**
- ❑ **Proof of address**
- ❑ **...**

Validation

Valid card belonging to the borrower

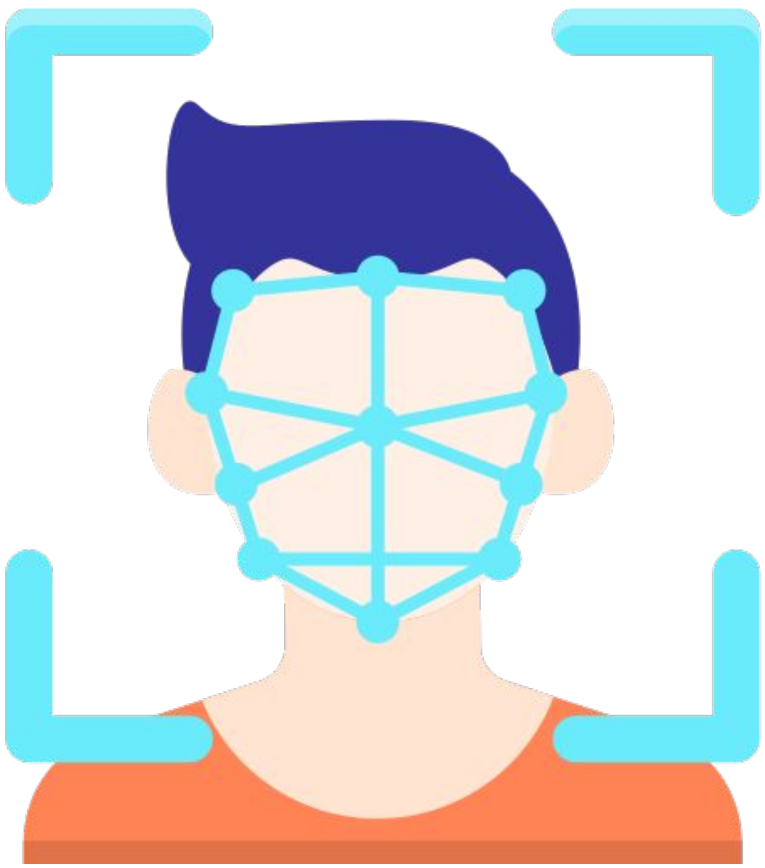
Examples of other pension information:

- **Dates** of payslip
- **Signatures and initials** contract

- **Build Advanced Deep Learning models**
- **How to minimize document annotation ?**

Use cases examples: network compression

EXAMPLE: cell phone facial recognition



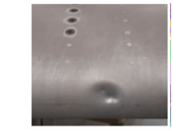
Facial recognition involves multiple very deep neural networks

Limited memory and processing power

- **How to reduce prediction time ?**

Scope of Frugal AI project & State-of-the-art overview

MANAGE
LOW VOLUMES
OF DATA



handle low volumes of data

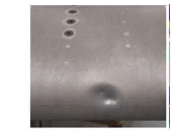
FEW SHOT LEARNING

- One-Shot Learning
- Zero Shot Learning

DATA AUGMENTATION

- Variational Auto-Encoder

MANAGE
LOW VOLUMES
OF DATA

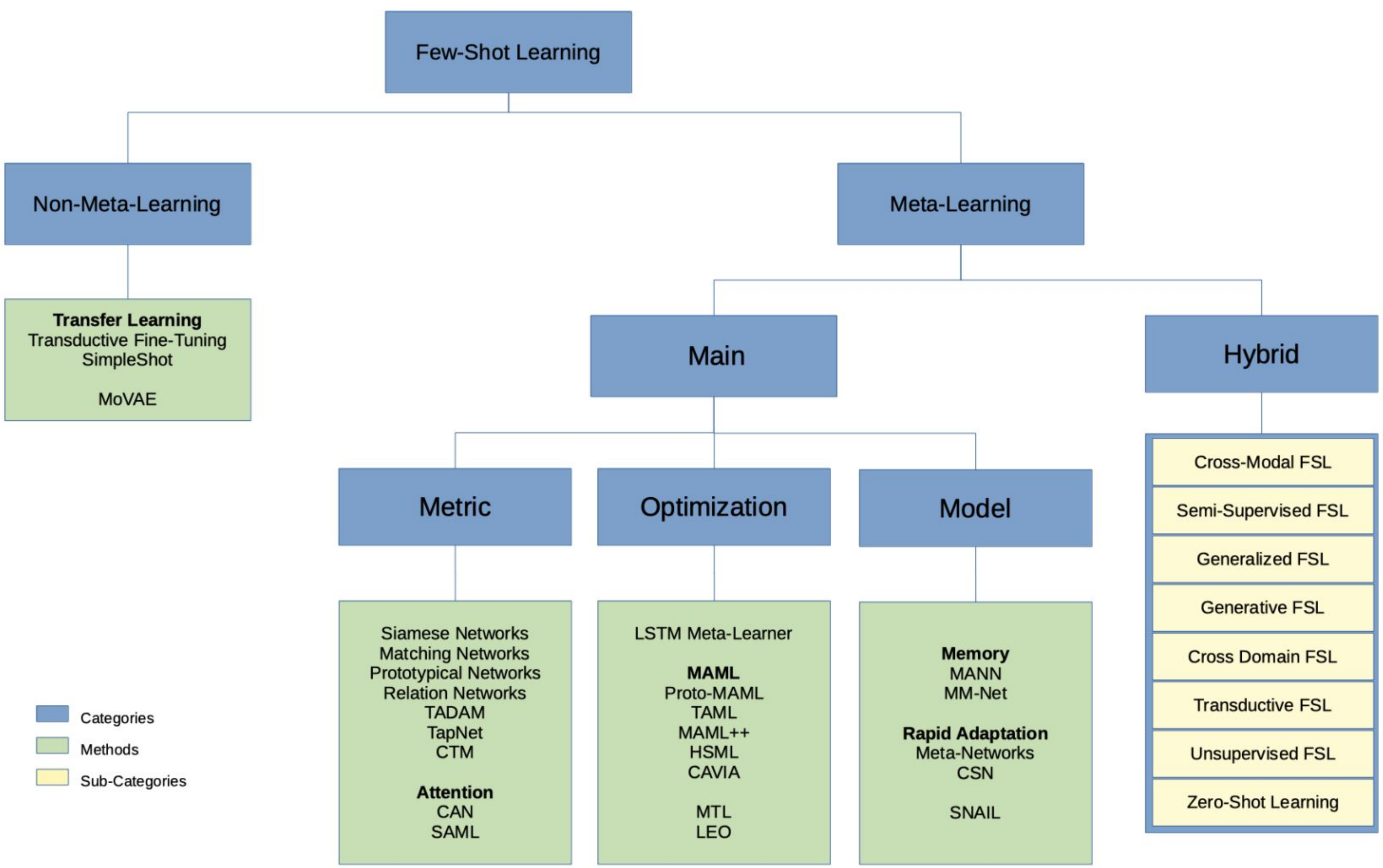


handle low volumes of data

Few-Shot Learning

capacity of a Machine Learning classification model to recognize a class which has been seen few times.
useful when you have low volume of data

Exemple : Machine Learning model which classifies rare objects or rare events



Metric : task of learning a distance function over data samples to discriminate the different classes (embedding)

Optimization : task of finding optimization-based methods that enables optimization procedure s.t. Gradient Descent to work on limited training examples

Model : task of finding modele architectures tailored for fast learning wich are distinguished in 3 categories (memory, rapid adaptation and miscellaneous)

[NLP example]

datacraft*

handle low volumes of data

One-Shot Learning

capacity of a Machine Learning classification model to recognize a class which has been seen only once.
useful when you have low volume of data and some classes have only one sample

Exemple : Facial recognition model: *face_recognition* library https://pypi.org/project/face-recognition/

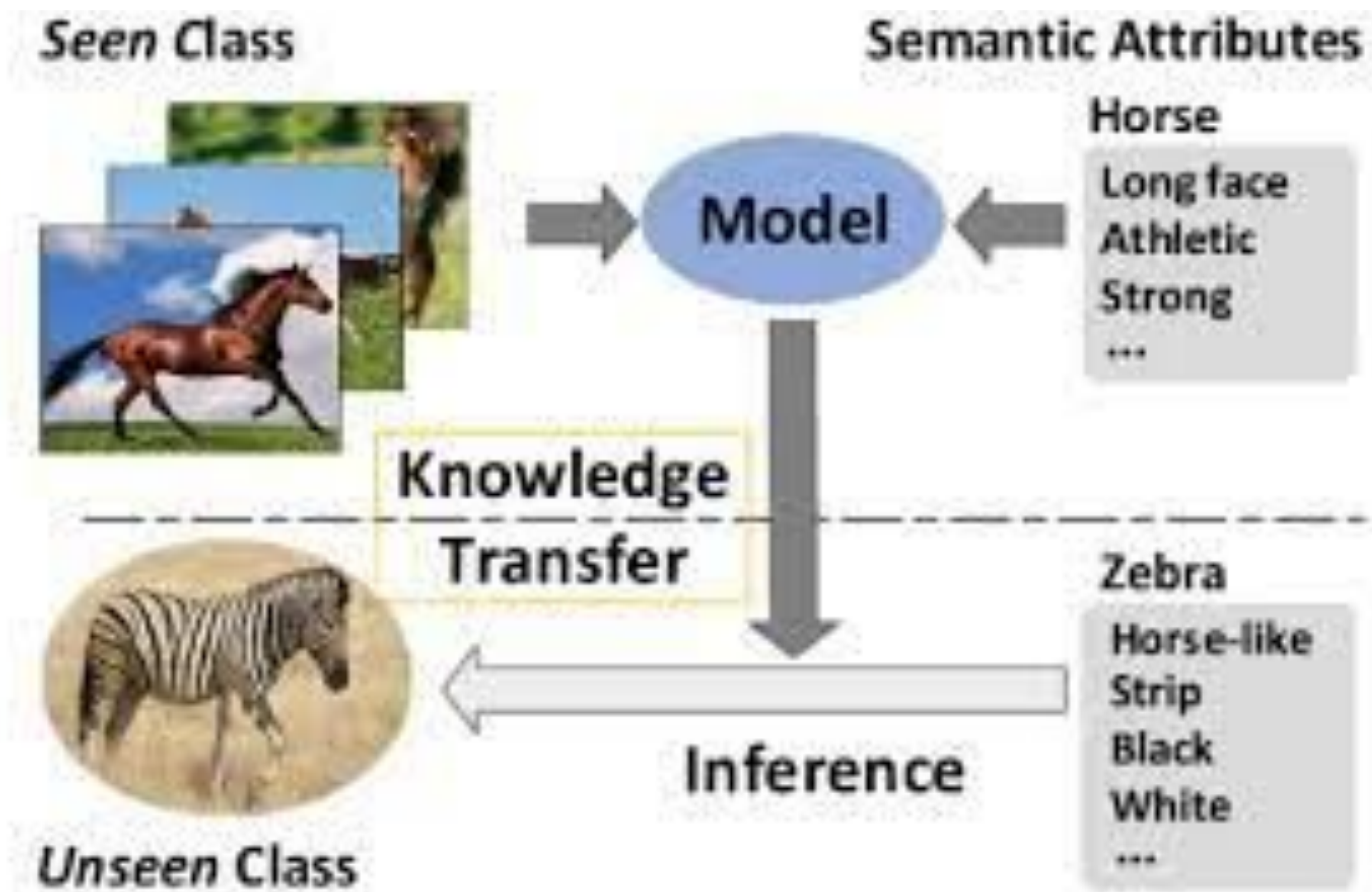
Extraction of facial landmarks -> distance measurement -> probability label assignment

handle low volumes of data

Zero-Shot Learning

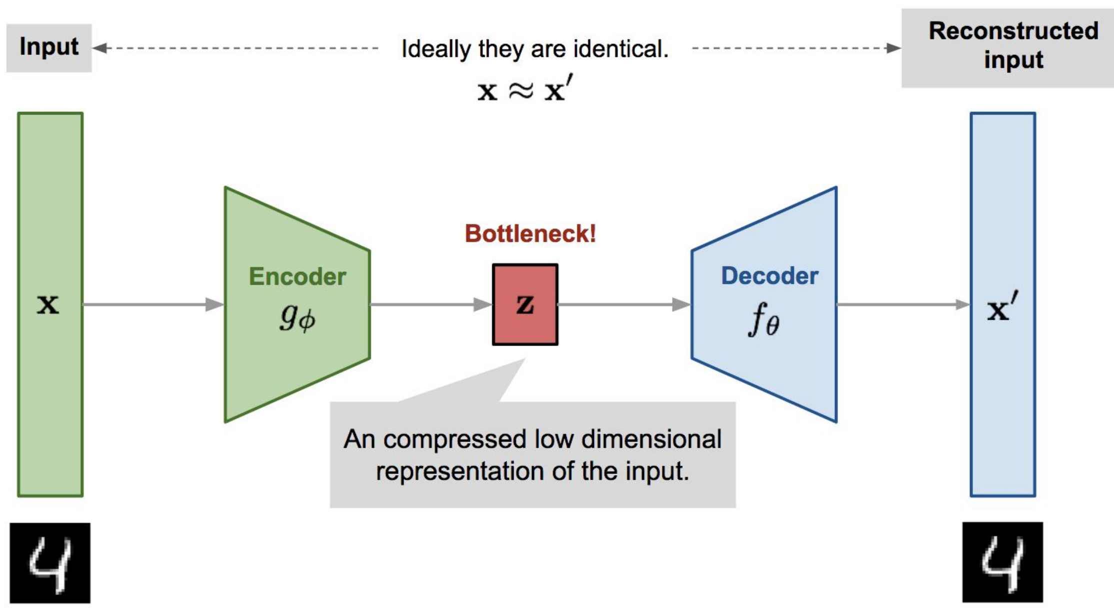
capacity of a Machine Learning classification model to recognize a class that it hasn't seen before but needs a secondary description

Exemple : Machine Learning model which classifies the species (impossible to have at least a training sample for each class)



handle low volumes of data

Variational Auto Encoder (VAE): generate new data



Main idea : have a powerful and compact representation of data in order to understand it

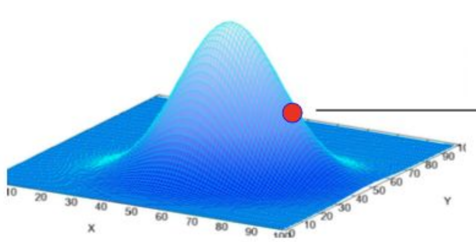
2 parts :

- Encoder : compress/encode data in a smaller space (latent space)

- Decoder : decode the latent to reconstruct original data

Latent space : compressed low dimensional representation of the input

By imposing a probabilistic law of the latent space, we can generate synthetic data.



Decoding

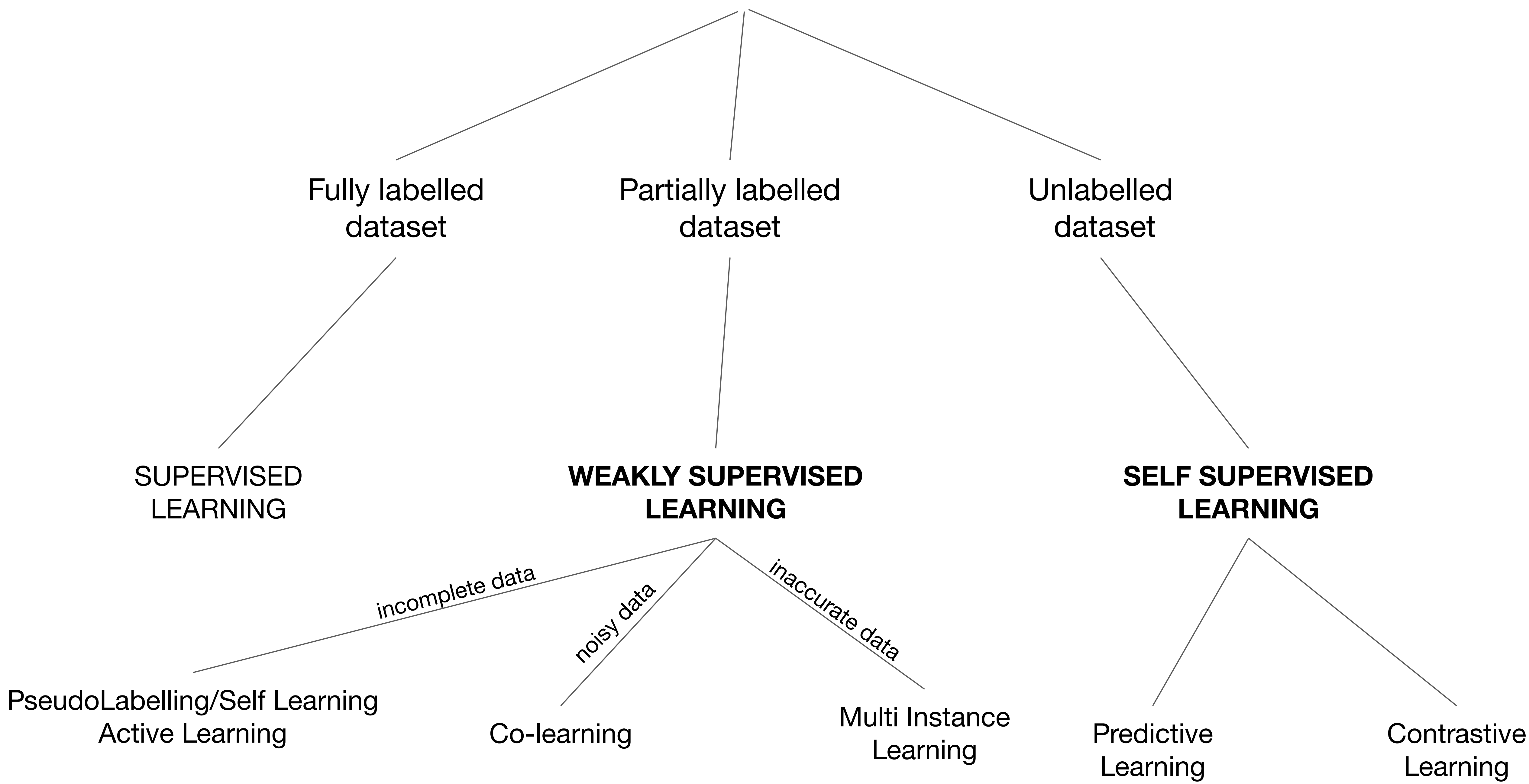
Probabilistic model in latent space

Synthesis of random image

[data augmentation in NLP]

MINIMIZE
REQUIRED VOLUMES
OF ANNOTATIONS

minimize required volume of annotations

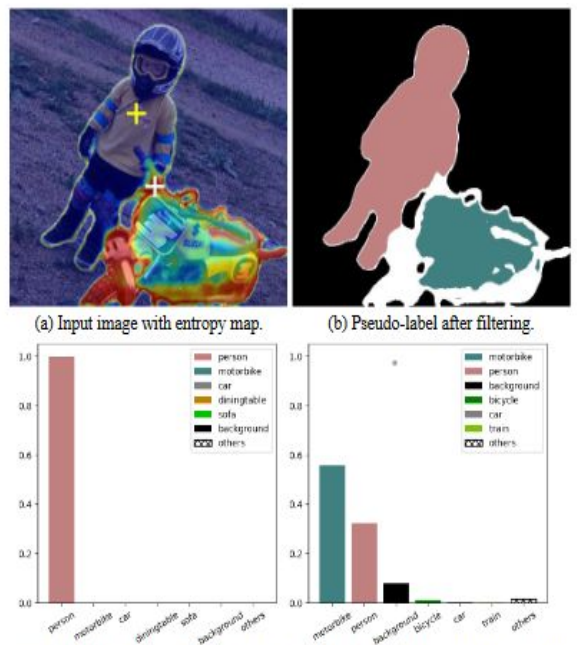


minimize required volume of annotations

Weakly-Supervised Learning

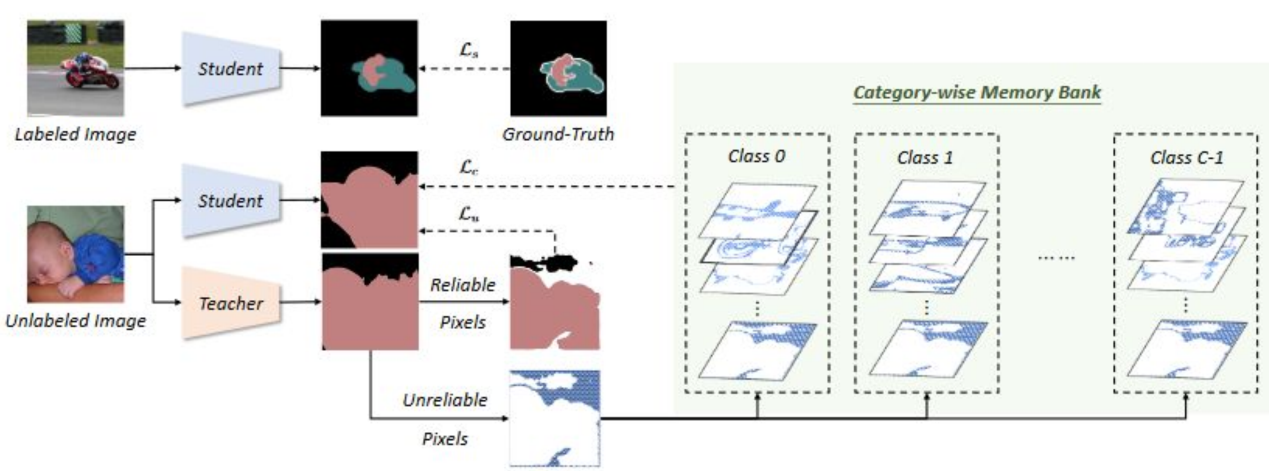
SELF LEARNING (Pseudo-Labelling)

General principle:
1. Train a first model on labelled data (supervised learning)
2. Use the trained model to infer pseudo-labels on unlabelled images
3. Select the most confident inferences
4. Use these as training labels and loop in 1



Pixel confidence evaluated through score entropy

U2PL* evolution: use also non confident pixels as negatives:

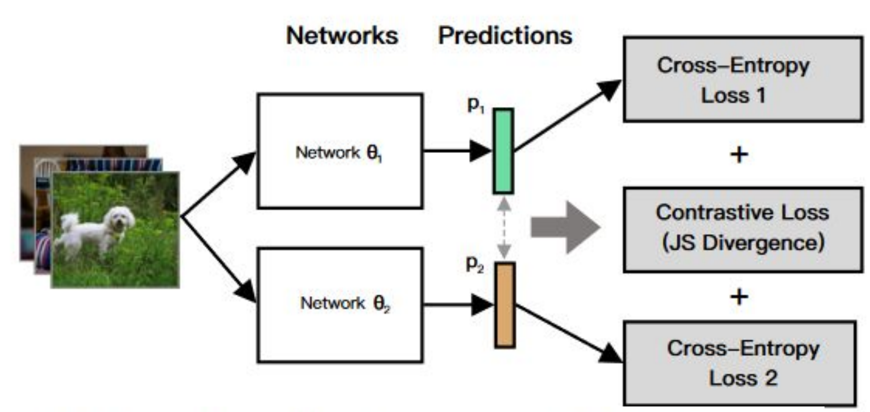


ACTIVE LEARNING

General principle:
1. Train a first model on labelled data (supervised learning)
2. Use the trained model to infer pseudo-labels on unlabelled images
3. Derive from these inferences the most valuable images to manually annotate next based on:
   a. Uncertainty sampling (entropy sampling)
   b. Diversity sampling (cluster based sampling)
   c. Representative sampling (margin sampling)
4. Annotate the selected images and loop in 1

CO TEACHING / CO LEARNING**

General principle: for noisy labels::
1. Train two models in parallel and asses their agreement/disagreement
2. Build a loss as weighted sum of cross entropy & mutual entropy



$$\ell(x_i) = (1-\lambda) * \ell_{\text{sup}}(x_i, y_i) + \lambda * \ell_{\text{con}}(x_i)$$

*Semi Supervised Semantic Segmentation Using Unreliable Pseudo Labels, 2022

**Combating Noisy Labels by Agreement: A Joint Training Method with Regularization, 2020

MINIMIZE
REQUIRED VOLUMES
OF ANNOTATIONS

minimize required volume of annotations

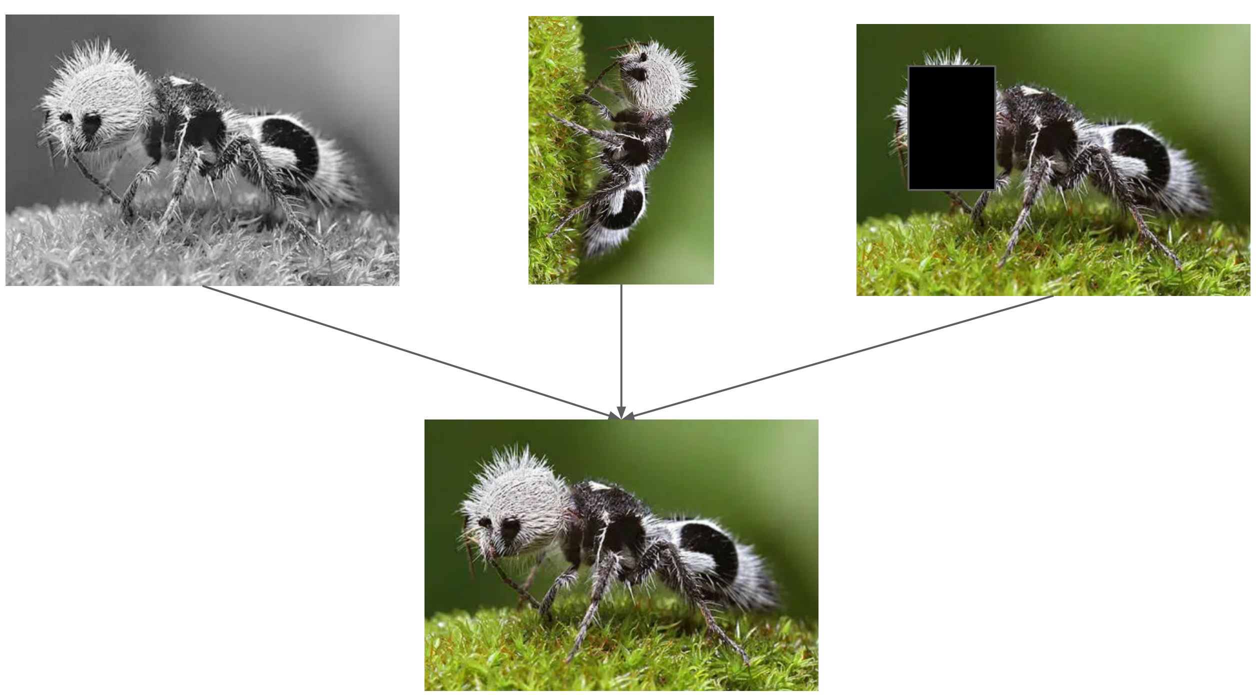
Self-Supervised Learning (unlabelled data)

PREDICTIVE LEARNING

<u>General principle:</u>

1. Use a “pretext task” whose ground truth can be extracted automatically from unlabelled data to pre-train a neural network
2. This pre-trained network is then fine tuned for the target task (with supervised or semi-supervised learning)

Examples of pretext tasks:



CONTRASTIVE LEARNING

<u>General principle:</u>

1. Learn a representation invariant to data augmentations performed on input unlabelled images
2. Build on this representation for the target task (e.g. linear classifier)

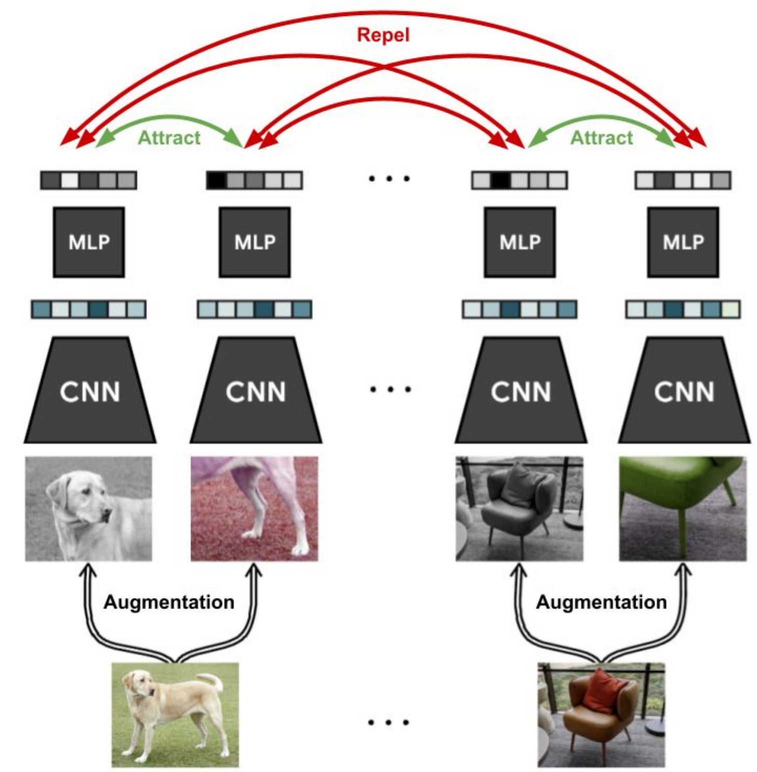


Image from SimCLR**

*Semi Supervised Semantic Segmentation Using Unreliable Pseudo Labels, 2022

**SimCLR: A Simple Framework for Contrastive Learning of Visual Representations, 2020

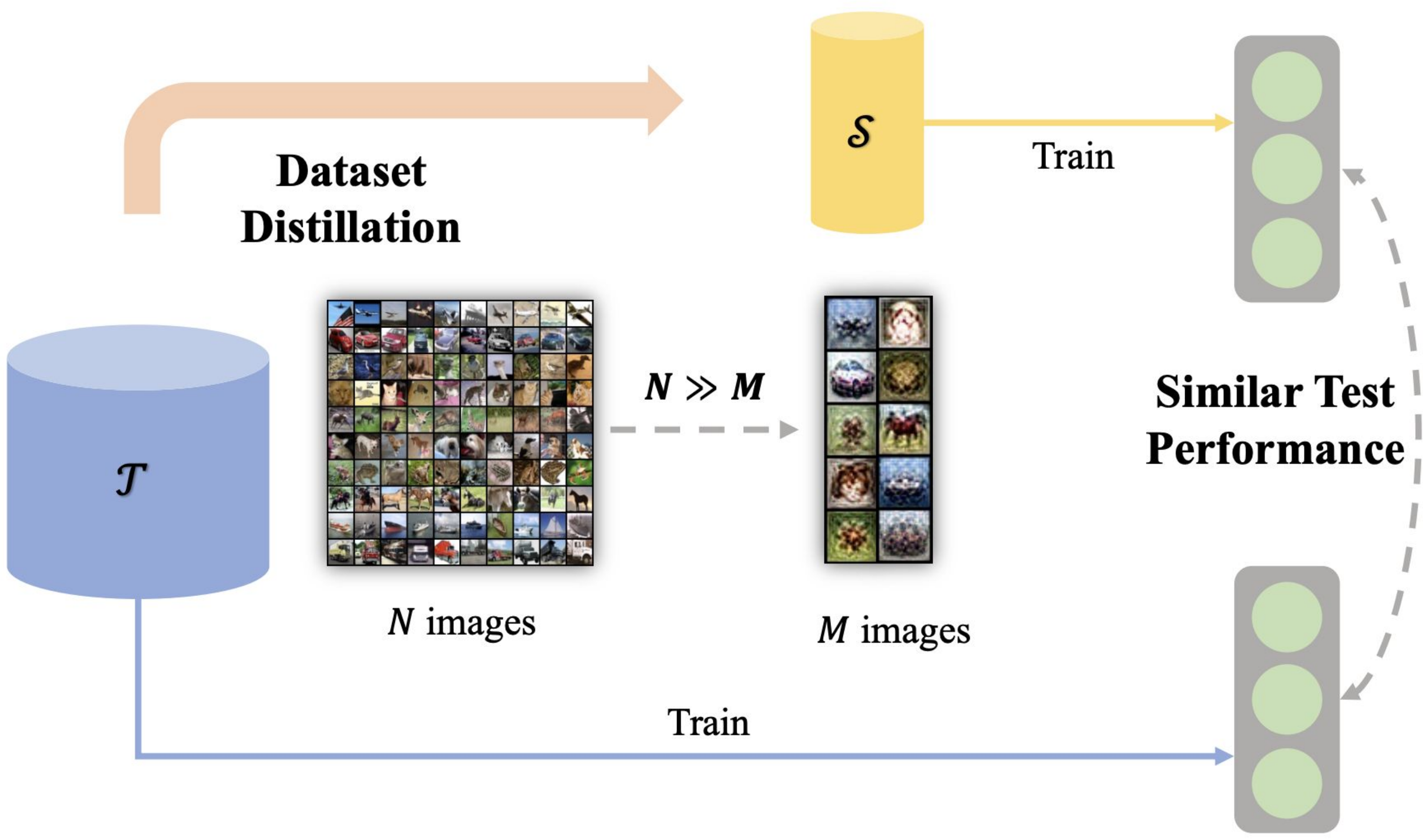
datacraft*

reduce environmental impact

Dataset distillation

task of synthesizing a large dataset into a smaller one in order to train a model which has the same performance compared to another one trained on the full dataset
useful when your computing capacities are limited or when you want to reduce your environmental impact

Example : Image classification model based on the CIFAR10 dataset



Main idea :

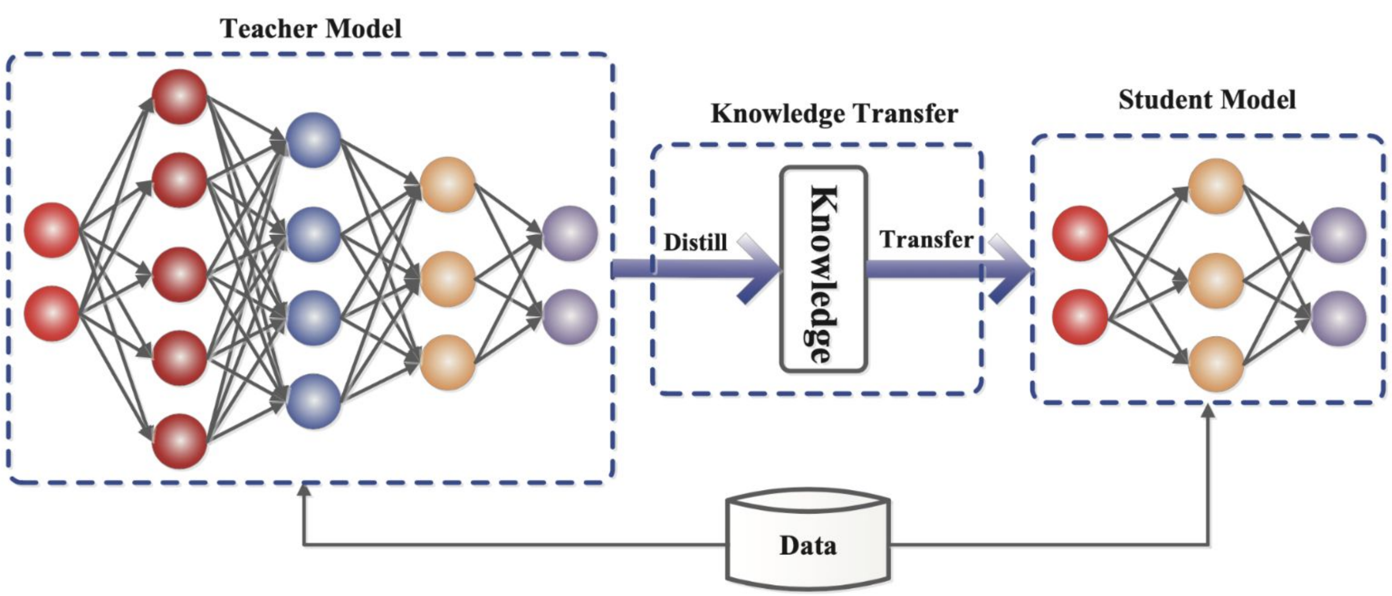
- Create a dataset of M synthetic images from N real images
- Train a model on these synthetic images

[Dataset Distillation for Text Classification]

reduce environmental impact

Model distillation

task of transferring knowledge from a teacher model to a simpler student model
both models share the same data
useful when your computing capacities are limited or when you want to reduce your environmental impact



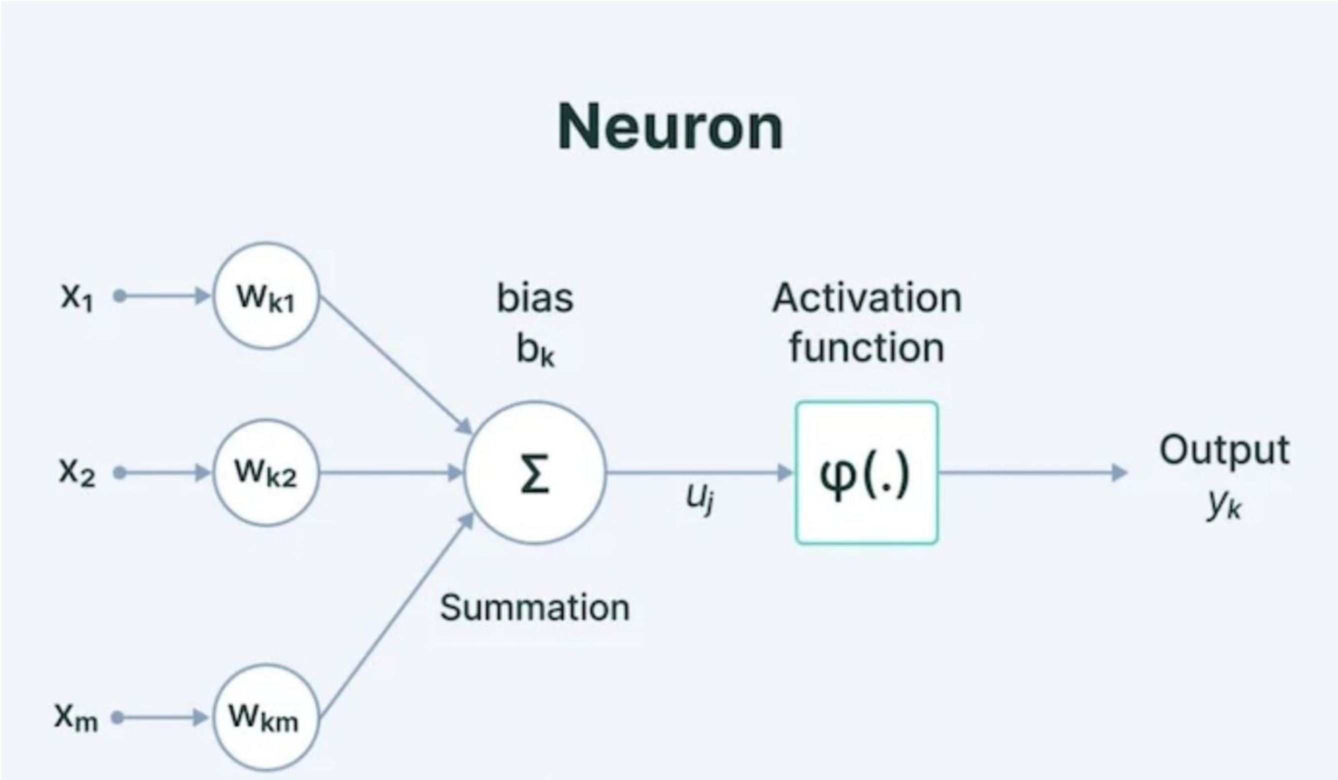
3 different types of knowledge transfer :

- Response-based knowledge : focuses on the final output layer to learn to mimic the predictions
- Feature-based knowledge : focuses of the features layers learned by the teacher model to learn the feature activations to the student model
- Relation-based knowledge : focuses on the relations (similarity matrix, feature embeddings, probabilistic distributions, …) between feature maps which are learned by the teacher model

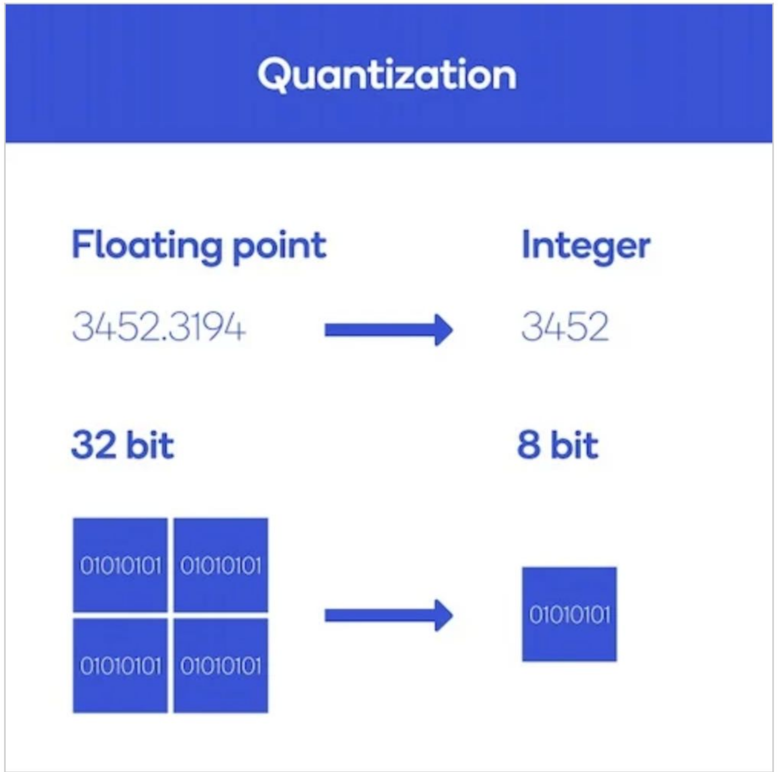
reduce environmental impact

Quantization

process of reducing the precision of the weights, biases and activations in au Neural Network by approximate them
generally, we reduce the precision from 32 bits to 8 bits



Artificial neuron reminder



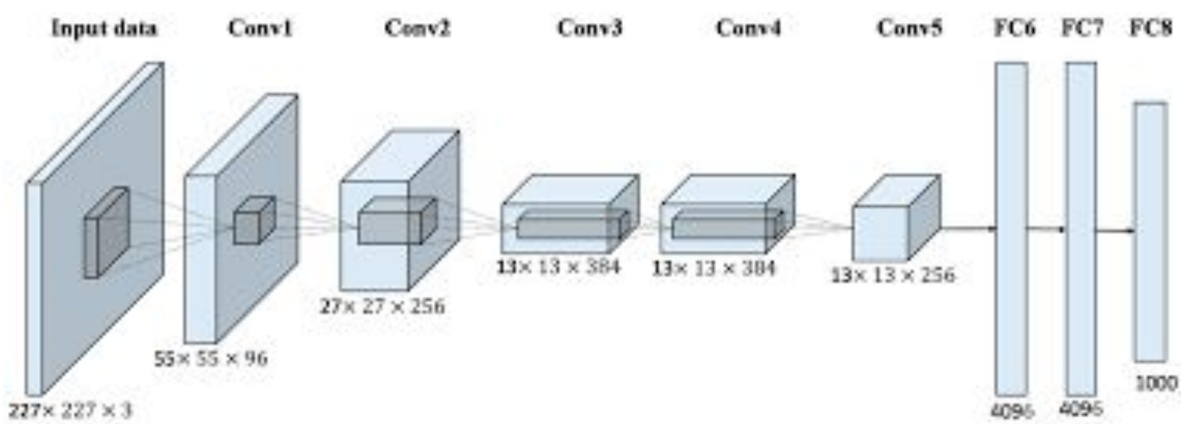
Quantization principle

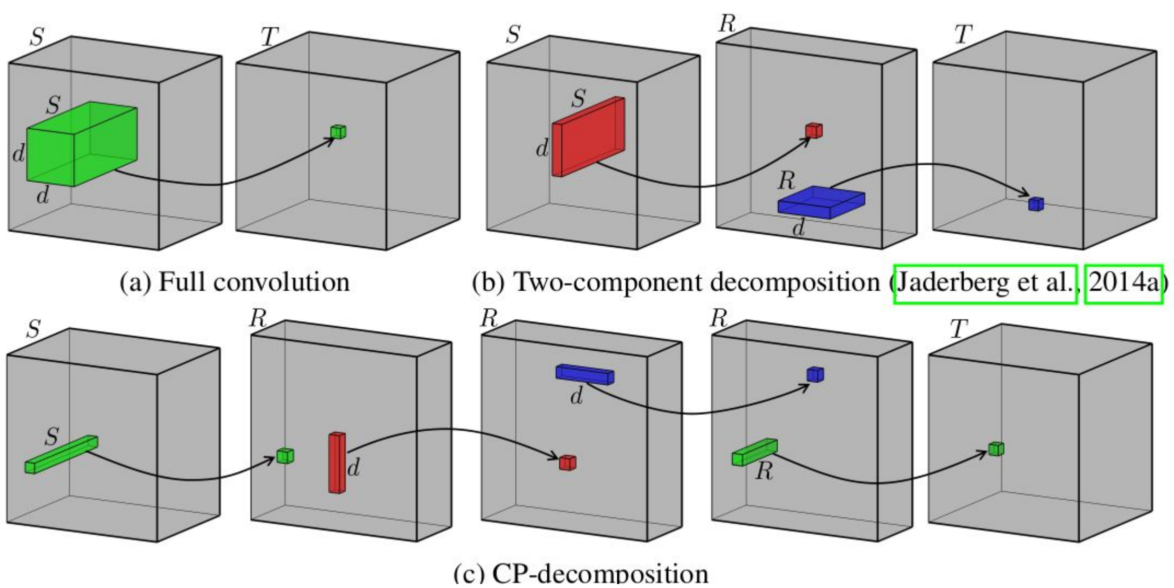
REDUCE ENVIRONMENTAL IMPACT

reduce environmental impact

Tensor compression

Reduce the inference time compressing the convolutional layers of a CNN which are tensors
We need to choose a way to compress the tensors
Several methods proposed in the literature : CP decomposition, Tensor Train....





Main benefits :

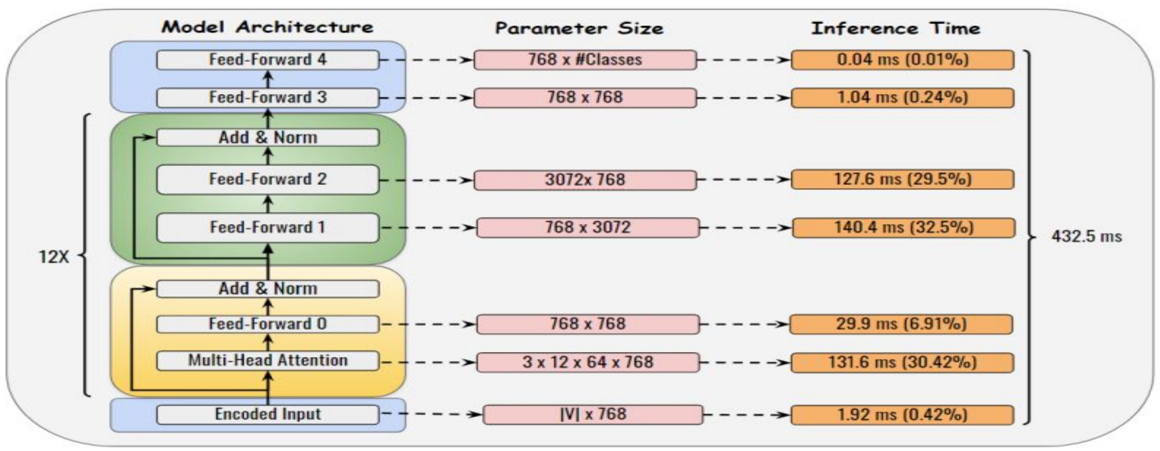
- **Gain in terms of inference time**
- **Small loss in accuracy**
- **Interpretability of the reduced layers**

REDUCE
ENVIRONMENTAL
IMPACT

reduce environmental impact

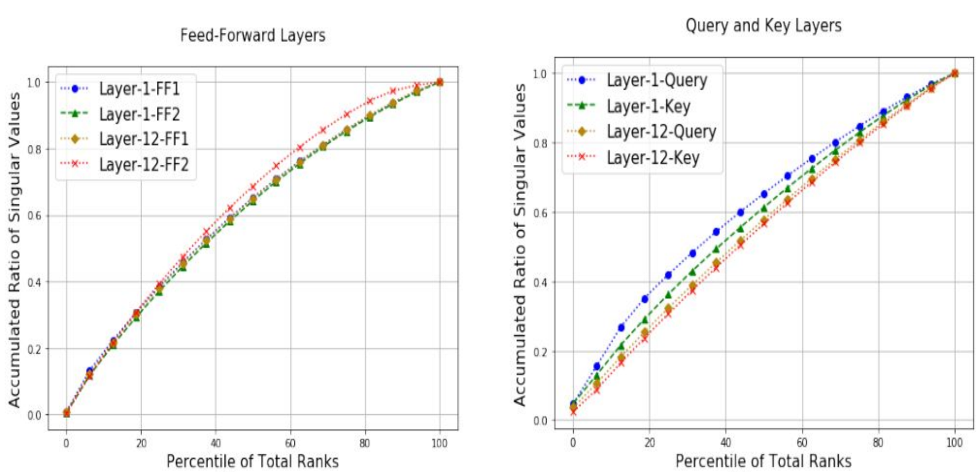
Applications to several tasks and several architecture

Initially proposed in the Computer Vision community
Approach that can be extended to NLP tasks



Facts

- Initial structure is not low rank
- Room for improvement in terms of compression of the layers



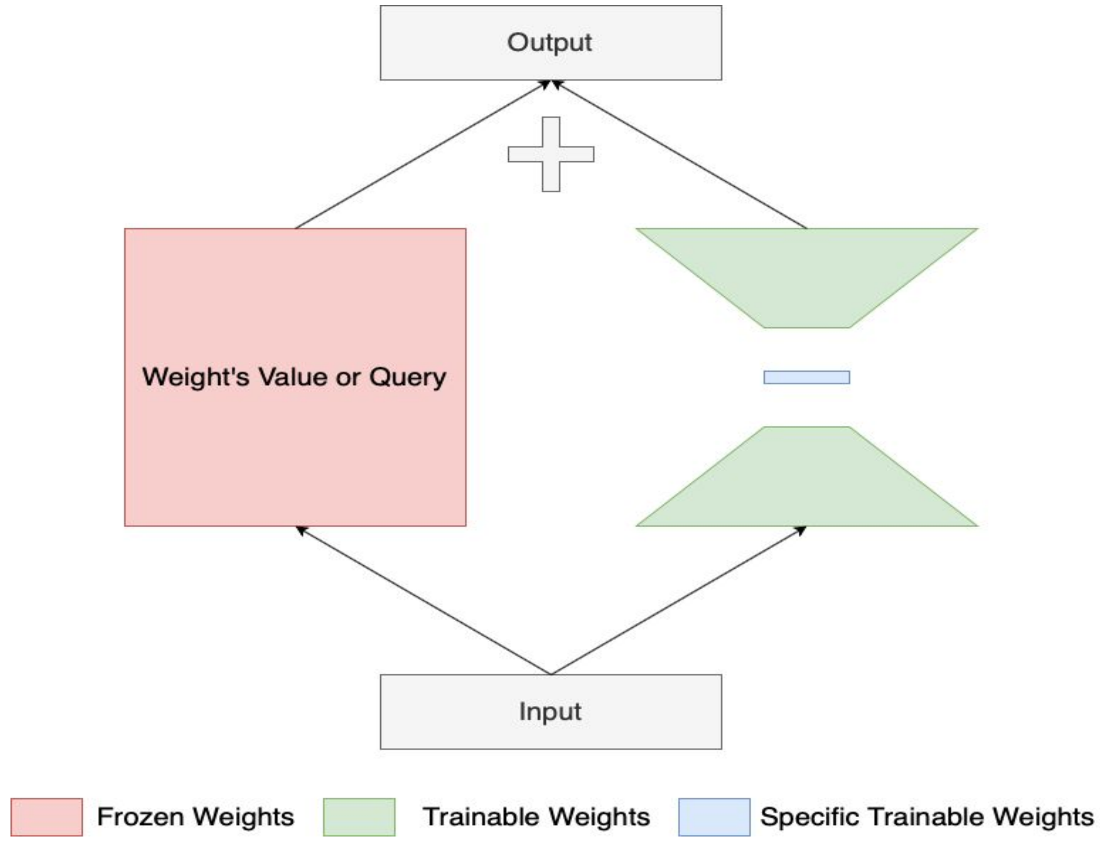
(a) (b)

reduce environmental impact

Applications to several tasks and several architecture

Initially proposed in the single task setting

Approach that can be extended to Multi-Task learning



Focus on measuring tools

Need of tools to measure the impact of the different mentioned methods on the environment

Cloud providers measuring tools

- AWS
- GCP
- Azure

Implemented as services and easy to implement in the different projects on the cloud
Not everyone has access to these tools (costs, on premise servers, ...) and lack of transparency

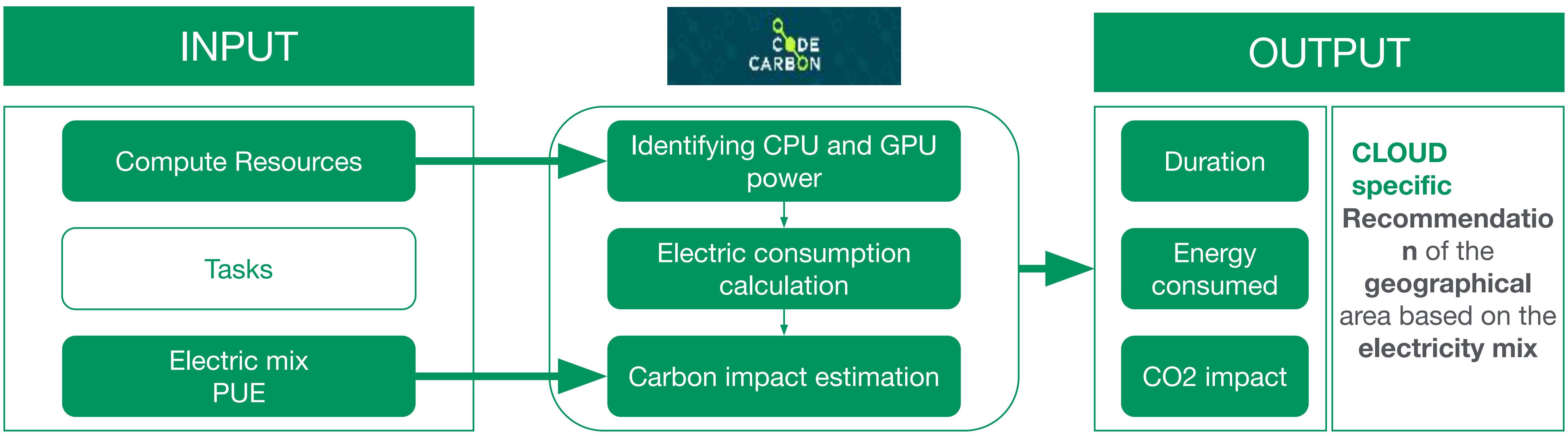
Focus on open sources Python package, easy to install and usable by all

Python libraries

- CodeCarbon
- carbonai
- CarbonTracker

measuring tools : CodeCarbon Library

CodeCarbon **estimates** the **amount of carbon dioxide** (CO2) produced by the **cloud** or **personal computing resources** used to execute the code



- **A lightweight and easy-to-use Python pip package**
- **Emissions tracked based on your power consumption & location-dependent carbon intensity**
- **Effective visualization of outputs in an integrated dashboard**
- **Open-source, free, and driven by the community**

measuring tools : CodeCarbon Library

Data Sample on a VM with 62 GB of RAM and 4 CPU cores

Phase	Duration	Emissions grams of CO_2	Consumption kWh
Training	30 min	9.08	0.021
Predicting	17 sec	0.07	0.001
TOTAL	30.5 min	9.15	0.022

Result sample

100 Training = 1 Home-to-work trip

Code sample for ML model fitting

```
loss_fn = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)

model.compile(optimizer="adam", loss=loss_fn, metrics=["accuracy"])

tracker = EmissionsTracker()
tracker.start()
model.fit(x_train, y_train, epochs=10)
emissions: float = tracker.stop()
print(f"Emissions: {emissions} kg")
```

- **Integrated into our internal AutoML solution**
- **Best choice between model quality and CO_2 emissions**

datacraft*

measuring tools : carbonai Library

Library developed by Capgemini that measures the power consumed by a Python function

Compatible with classical Machine Learning models (scikit-learn)

Initialization with project information in JSON format (project name, CPU/GPU characteristics, country, ...)

Each time a project function is called, a CSV is updated with information like :

- date and time
- country
- name of the program
- CPU and GPU usage time
- cumulative energy of the program

```
from carbonai import PowerMeter
power_meter = PowerMeter(project_name="MNIST classifier", is_online=False, location="FR")

@power_meter.measure_power(
    package="sklearn",
    algorithm="RandomForestClassifier",
    data_type="tabular",
    data_shape=<your_data>.shape,
    algorithm_params="n_estimators=300, max_depth=15",
    comments="Classifier trained on the MNIST dataset, 3rd test"
)
def my_func(arg1, arg2, ...):
    # Do something
```

Code example

measuring tools : CarbonTracker Library

Tool for tracking and predicting the energy consumption and carbon footprint of training deep learning models

Initialization with information such that number of epochs, components to monitor (CPU, GPU), the number of epochs to monitor, the quantity of information displayed

At the end of the training, information about it are displayed like :

- time
- energy consumption
- CO2 equivalent
- comparison with the distance travelled by car

```
from carbontracker.tracker import CarbonTracker

tracker = CarbonTracker(epochs=max_epochs)

# Training loop.
for epoch in range(max_epochs):
    tracker.epoch_start()

    # Your model training.

    tracker.epoch_end()

# Optional: Add a stop in case of early termination before all monitor_epochs has
# been monitored to ensure that actual consumption is reported.
tracker.stop()
```

Code example

```
CarbonTracker: The following components
 ↪  were found: GPU with device(s) TITAN
 ↪  RTX. CPU with device(s) cpu:0, cpu:1.
CarbonTracker: Carbon intensity
 ↪  for the next 1:54:54 is predicted to
 ↪  be 54.09 gCO2/kWh at detected location:
 ↪  Copenhagen, Capital Region, DK.
CarbonTracker:
Predicted consumption for 100 epoch(s):
        Time:   1:54:54
        Energy: 1.159974 kWh
        CO2eq:  62.744032 g
        This is equivalent to:
        0.521130 km travelled by car
CarbonTracker: Average
 ↪  carbon intensity during training
 ↪  was 58.25 gCO2/kWh at detected location:
 ↪  Copenhagen, Capital Region, DK.
CarbonTracker:
Actual consumption for 100 epoch(s):
        Time:   1:55:55
        Energy: 1.334319 kWh
        CO2eq:  77.724065 g
        This is equivalent to:
        0.645549 km travelled by car
CarbonTracker: Finished monitoring.
```

Output example

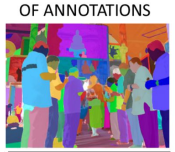
datacraft*

Next steps!

Upcoming workshops

MANAGE
LOW VOLUMES
OF DATA

MINIMIZE
REQUIRED VOLUMES
OF ANNOTATIONS



WORKSHOPS - End of June (29th tbc)

- Presentation of use cases and data
- state of the art on the methods allowing to treat these use cases
- identification of easily implementable methods (paper with code)

REDUCE
ENVIRONMENTAL
IMPACT

BENCHATHON ON MEASURING TOOLS - In July (tbc)

- assessment of measuring tools to evaluate AI models environmental impacts
- framework assessment: define evaluation criteria, metrics and a standard use-case

US PARTNERSHIP

Inviting interested U.S. companies and universities to participating to events

In Autumn....

- Workshops on low volumes of data and minimize volumes of annotations
- Workshops on benchmarking of compression methods w.r.t. environmental impacts
- Organize joint workshops with the US and online discussion to share results
- Plan a learning expedition in US

How to contribute?

- **Join the core team** and have a leading role on the scope, practical use cases, data provision
- Help in the **preparation of workshops**
- Actively participate in the workshops and help **synthesize the content**
- **Share appropriate contents**
- any other way!