NEURAL CRITICALITY METRIC FOR OBJECT DETECTION DEEP NEURAL NETWORKS

AGENDA

WHY DO WE NEED CRITICALITY

Original idea: What is the equivalent of SoftWare Criticality Analysis (SWCA) in ML domain?
SWCA: Actuator-to-sensor analysis of SWC-interfaces and their dependencies/criticalities

We can quantify the contribution of a neuron (weights and biases) to the decision process

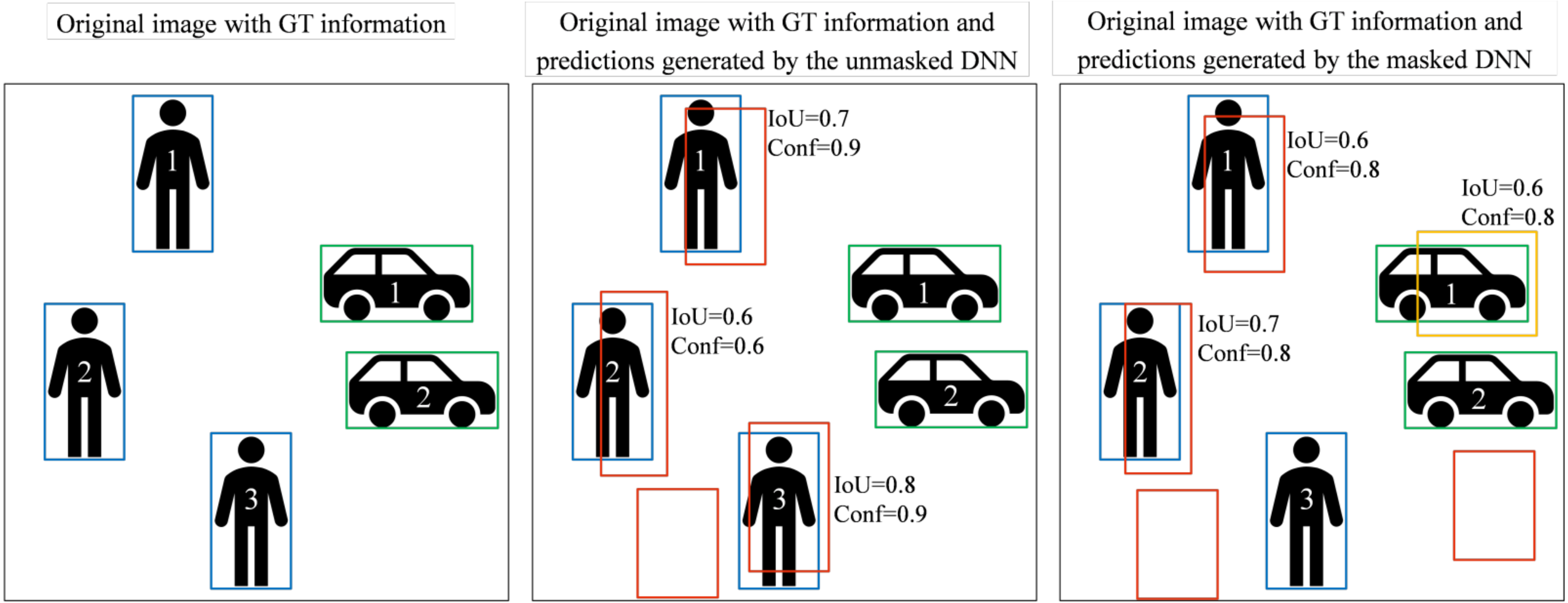
Analysis can discover critical feature extractors, the abilities of which can be re-distributed over several layers

The analysis, as part of the development process minimizes the uncertainty of unknown-unsafe scenario -> SOTIF compliant

Increase the understanding and reliability of the system and can be used for any other CV task

„To the best of our knowledge, there is no method, in the field of computer vision, that would evaluate and quantify the contribution of a neuron to the decision process of an arbitrary object detection DNN."

HOW TO CALCULATE CRITICALITY



Fußnote: ...
Quelle: ...

HOW TO CALCULATE CRITICALITY

	True Positive (TP)	False Negative (FN)	False positive (FP)
Reality	object present	object present	no object present
Prediction	object detected	no object detected	object detected

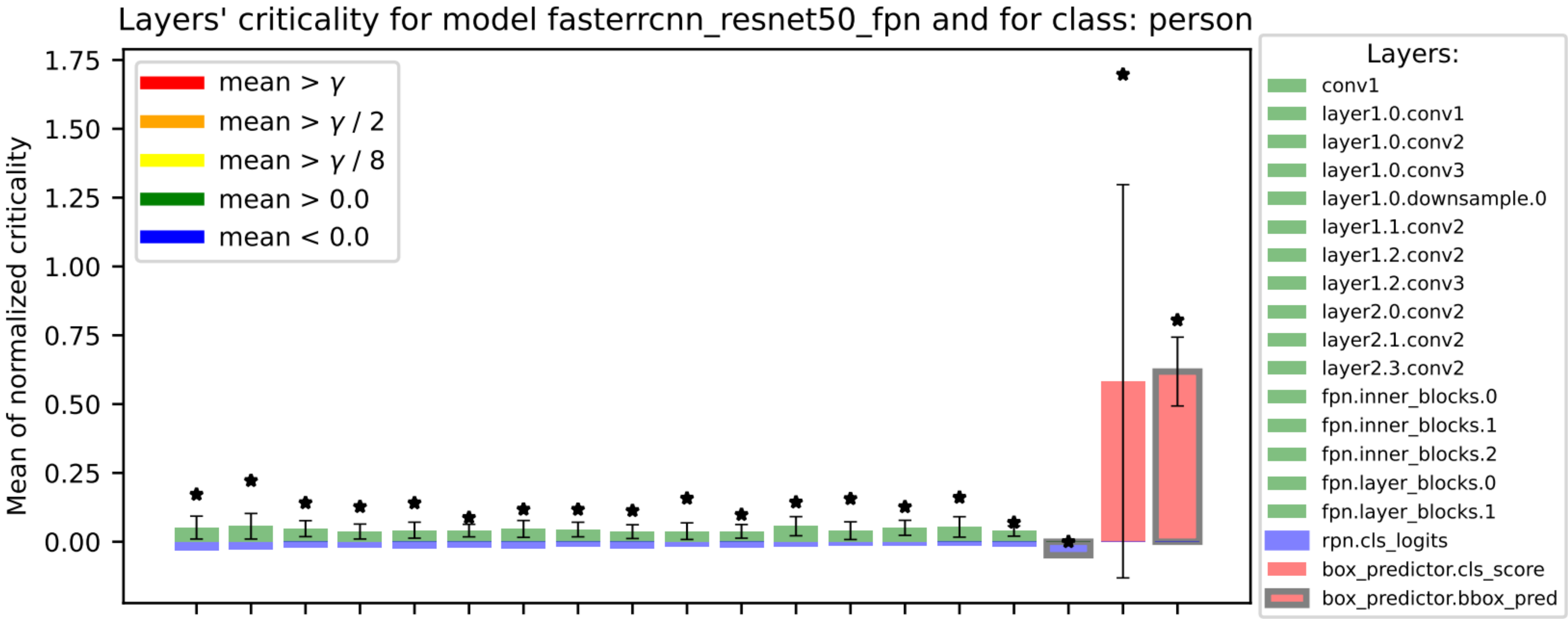
$$f_{cri,TP-FN,c_j}(CONF_{c_j}, IOU_{c_j}, CONF_{c_j,m}, IOU_{c_j,m}) = \frac{\sum_{i=1}^{n_{c_j}}((conf_i - conf_{i,m}) + (iou_i - iou_{i,m}))}{n_{c_j}}$$

$$f_{cri,FP,c_j}\left(n_{FP,c_j}, n_{FP,c_j,m}\right) = \min\left(\frac{n_{FP,c_j,m} - n_{FP,c_j}}{n_{FP,c_j} + \epsilon}, 2\right), \text{ where } \epsilon = 10^{-6}$$

$$f_{cri,total,c_j} = f_{cri,TP-FN,c_j} + f_{cri,FP,c_j}$$

Fußnote: ...
Quelle: ...

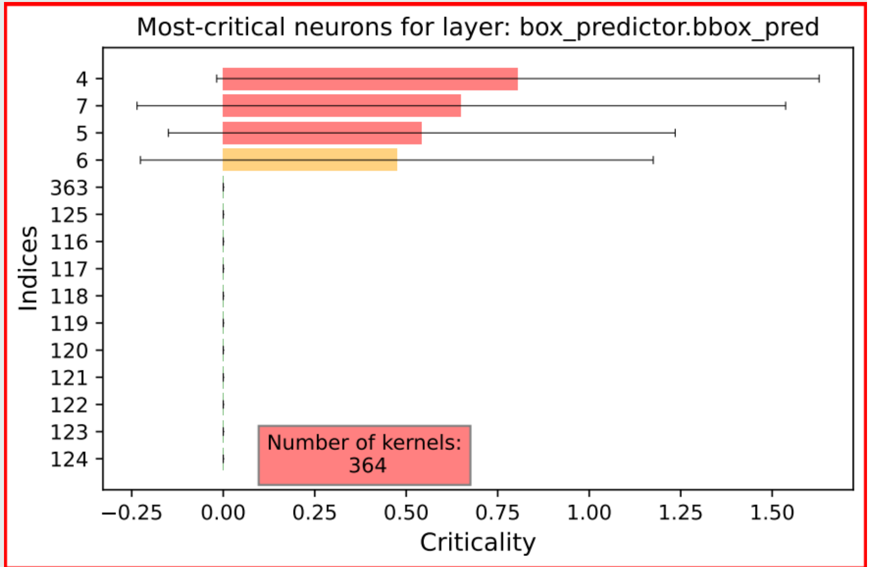
1) OVERALL MODELS' CRITICALITY



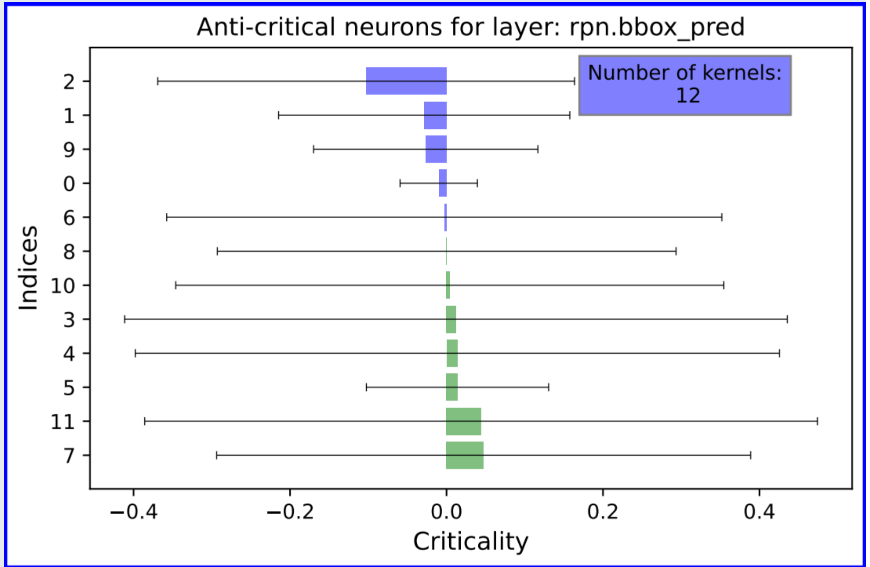
Fußnote: ...
Quelle: ...

2) MOST CRITICAL AND ANTI-CRITICAL LAYERS

CRITICAL NEURONS

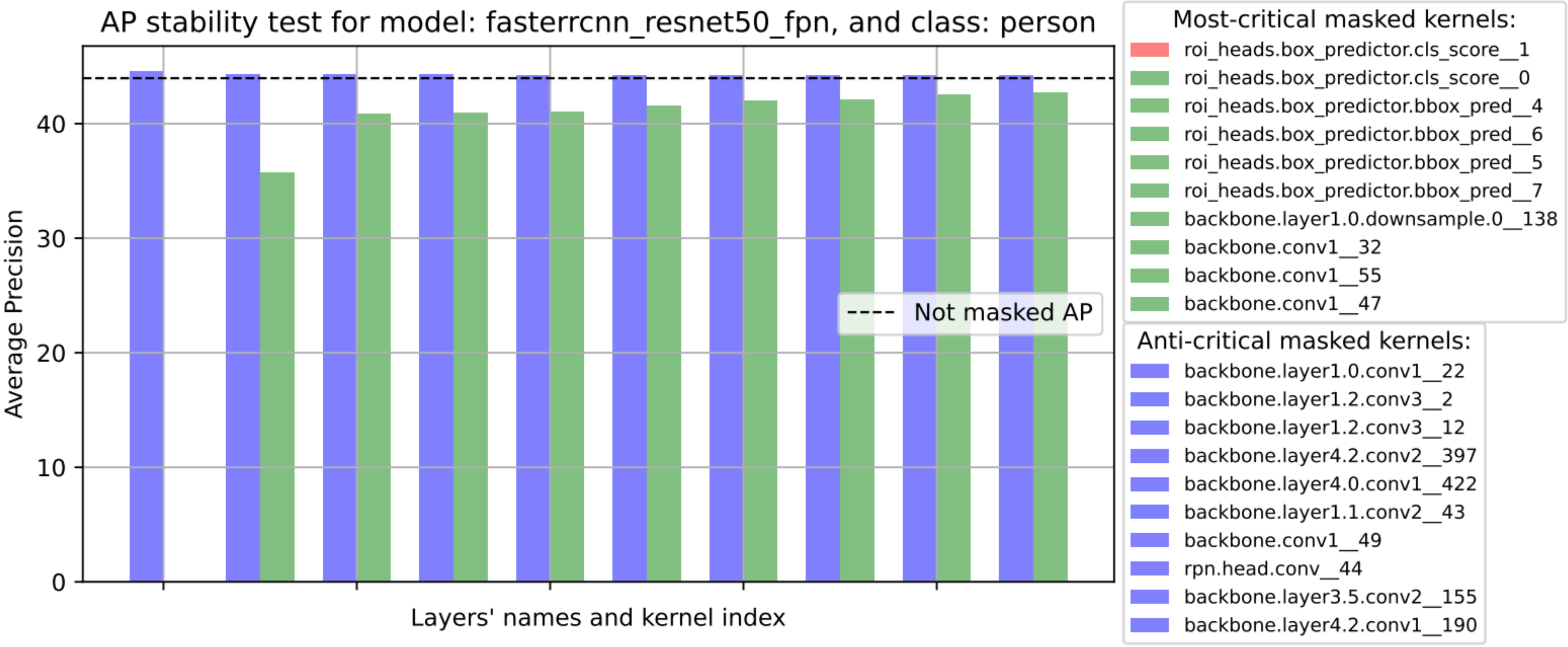


ANTI-CRITICAL NEURONS



Fußnote: ...
Quelle: ...

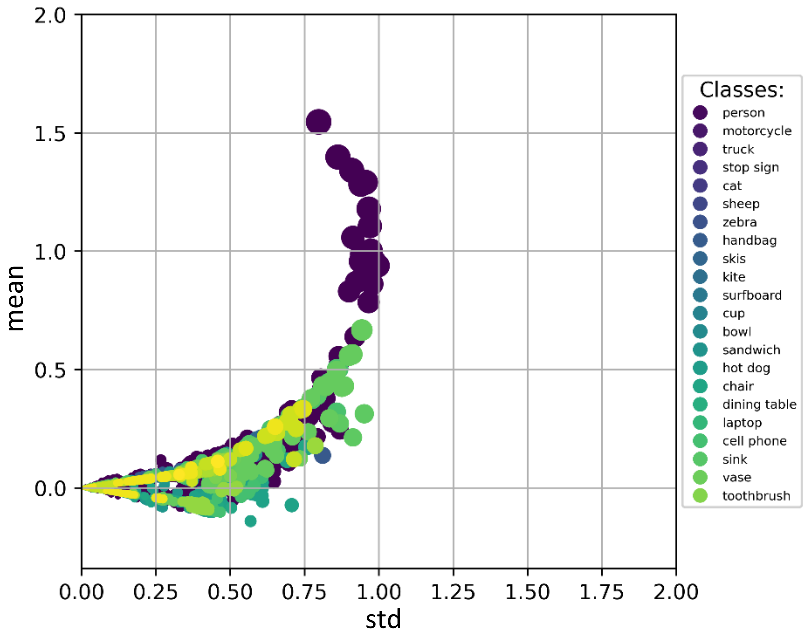
3) STABILITY TEST



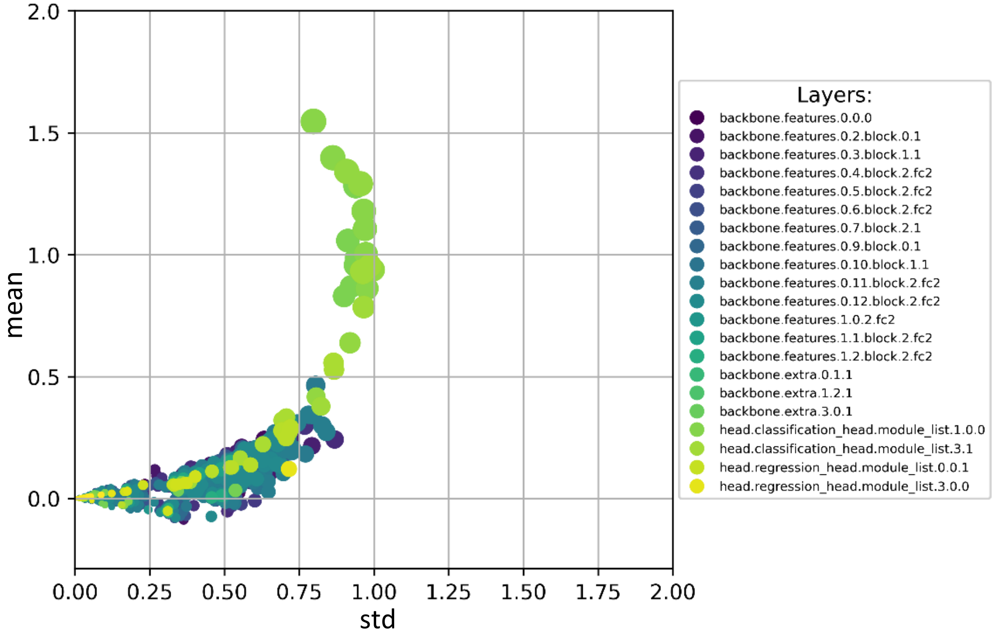
Fußnote: ...
Quelle: ...

4) FURTHER ANALYSIS – CLUSTERED NEURONS (CLASSES/LAYERS)

CRITICALITY - CLASSES CLUSTERED



CRITICALITY - LAYERS CLUSTERED – CLASS: PERSON



Fußnote: ...
Quelle: ...

CONCLUSION

	Number of neurons	Number of neurons $cri > 1.0$	$\|\|cr\|\|$ 'person'	AP 'person'	AP_{anti} 'person'	AP_{cri} 'person'
Faster R-CNN	31382	2	0.0295	43.97%	44.54%	**0.07%**
SSD	40244	**33**	0.0411	21.10%	21.16%	9.04%
YOLOv5	30781	1	0.0150	**44.41%**	**45.07%**	27.49%
DETR	**80576**	3	**0.0490**	20.34%	23.97%	0.65%

All analyzed models contain class-related critical neurons which can **prevent correct** object detection

The explanation of anti-critical neurons is more complex and would require a deeper dive into **cross-classes correlation**

Layers' MEAN criticality is a by-product of an inefficient or **non-present regularization** during the training

With this article we opened a space for a potential future work such as: evaluation and explanation of anti-critical neurons

From a functional safety point of view, the **optimal criticality** of the neurons within the layers should form a **uniform distribution**. How to achieve it?

OUTLOOK

INTEGRATION	• Incorporation of the criticality into training loss and backpropagation mechanism
FUNCTIONAL SAFETY	• Theoretical definition of a mapping to an ASIL
PRUNING CRITERION	✓ Neural Criticality as a potential network pruning metric
COMPUTATIONAL COST	• Minimize the computational costs and extend criticality to problem-specific parameters like distance to object or size of object

Fußnote: ...
Quelle: ...

CONTACT

WE MAKE **IDEAS** HAPPEN

ARRK ENGINEERING GMBH
Frankfurter Ring 160
80807 München
Deutschland

www.arrk-engineering.com

Ing. Václav Diviš
Senior Expert Machine Learning ADAS
ADAS Department, ARRK Engineering

+49 157 5010 0497
vaclav.divis@arrk-engineering.com

Tobias Schuster, M.Sc.
Junior Test Architect
Siemens Technology

+49 171 1187456
tobias.schuster@siemens.com

Ing. Marek Hrúz PhD.
Professor - UWB - Pilsen
DEPARTMENT OF CYBERNETICS

+420 37763 2555
mhruz@ntis.zcu.cz

BACKUP

Fußnote: ...
Quelle: ...

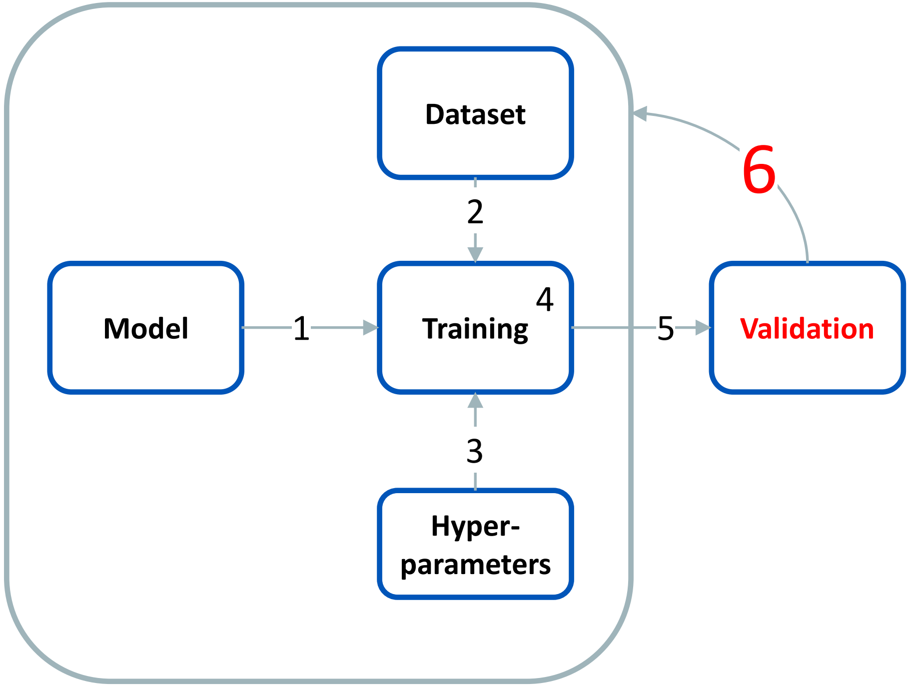
SIMPLIFIED ML TRAINING PROCESS

- **Data-driven supervised learning**

(object detection, classification, segmentation)

1. Define a model (SOTA Architecture)
2. Gather dataset (preprocess, data-augmentation)
3. Define initial training hyper-parameters
4. Setup a pipeline and start training
5. Validate the training results

6. Evaluate results, update training setup, dataset and repeat the training



ARRK ENGINEERING

ANALYSIS METHODOLOGY

- Dataset: **COCO Dataset**
- Analysed Networks:
  - Two stage: **Faster R-CNN**
  - One stage: **YOLO, SSD**
  - Attention: **DETR**

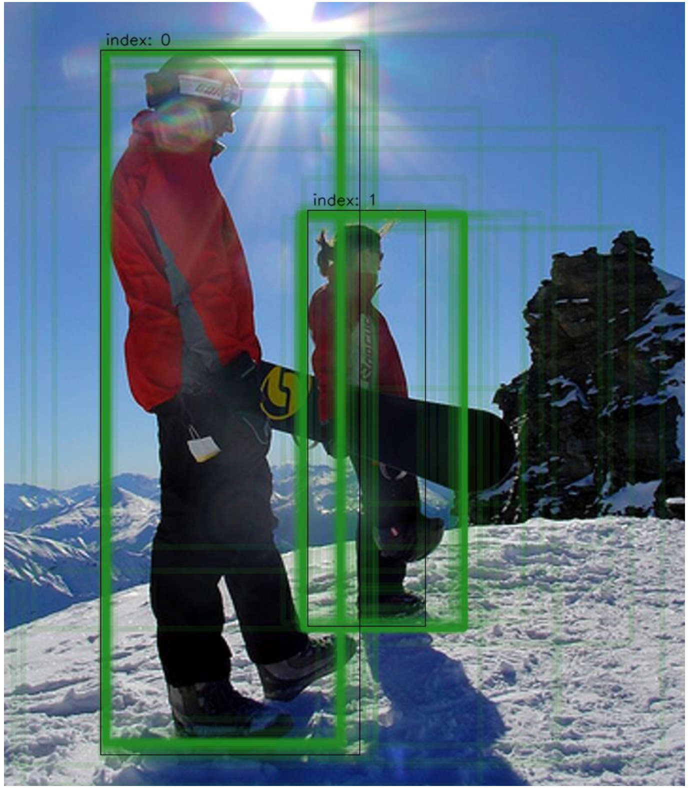
- Process:
  - For each image and each neuron calculate criticality
  - Conduct a stability test to verify

for *image* $d \in \mathcal{D}$ **do**
  $\mathcal{R}_{PR}, \mathcal{C}_{PR}, \mathcal{CONF}_{PR} = predict(DNN, d, \tau, \delta)$
  for *every* L *in* DNN **do**
    for *every* k *in layer* L **do**
      $DNN_{masked} = mask\_neuron(k)$
      $\mathcal{R}_{PR,m}, \mathcal{C}_{PR,m}, \mathcal{CONF}_{PR,m} = predict(DNN_{masked}, d, \tau, \delta)$
      $CONF, IOU, N_{FP} = match(\mathcal{R}_{PR}, \mathcal{C}_{PR}, \mathcal{CONF}_{PR}, \mathcal{R}_{GT}, \mathcal{C}_{GT}, \alpha)$
      $CONF_m, IOU_m, N_{FP,m} =$
      $match(\mathcal{R}_{PR,m}, \mathcal{C}_{PR,m}, \mathcal{CONF}_{PR,m}, \mathcal{R}_{GT}, \mathcal{C}_{GT}, \alpha)$
      for *every* c_j *in* $\mathcal{C}$ **do**
        $f_{cri,total,c_j} =$
        $f_{cri,TP-FN,c_j}(CONF_{c_j}, IOU_{c_j}, CONF_{c_j,m}, IOU_{c_j,m}) +$
        $f_{cri,FP,c}(n_{FP,c_j}, n_{FP,c_j,m})$
      $DNN_{masked} = unmask\_neuron(k)$

Image source: Taken from AUDI a2d2 dataset - Link

ARRK ENGINEERING

4) FURTHER ANALYSIS - BB HEATMAP



Fußnote: ...
Quelle: ...