Politecnico di Torino

FONDAZIONE links
PASSION FOR INNOVATION

Deep Learning at the Edge:
Optimizations of object detection models for embedded devices

Supervisors:

Prof. Andrea Calimera

Dott. Antonio Defina

Candidate:

Ivan Murabito

Thesis goal

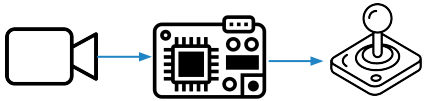
Analyze and evaluate the main **optimization and compression** methods of deep models,
with the purpose of perform **inference** of **object detections algorithms efficiently** on devices with **limited hardware resources.**

- improve inference speed
- reduce model size
- low accuracy trade-off

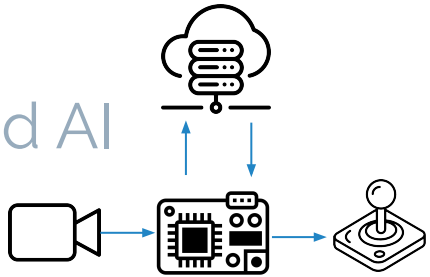
Inference at the edge: Motivations

Two ways:

Edge AI



Cloud AI



Edge AI Motivations

Low Latency	Privacy	Energy Consumption
Unlock the way for Real-Time Applications.	Does not exchange sensitive data over network.	Reduce load in datacenters and less cost and environmental impact.

EDGE AI Application Segments

AUTOMOTIVE

Autonomous Driving

Driving Assistance

Driver Monitor

INDUSTRY 4.0

Predictive Maintenance

Anomaly detection

DRONES

Autonomous Flying

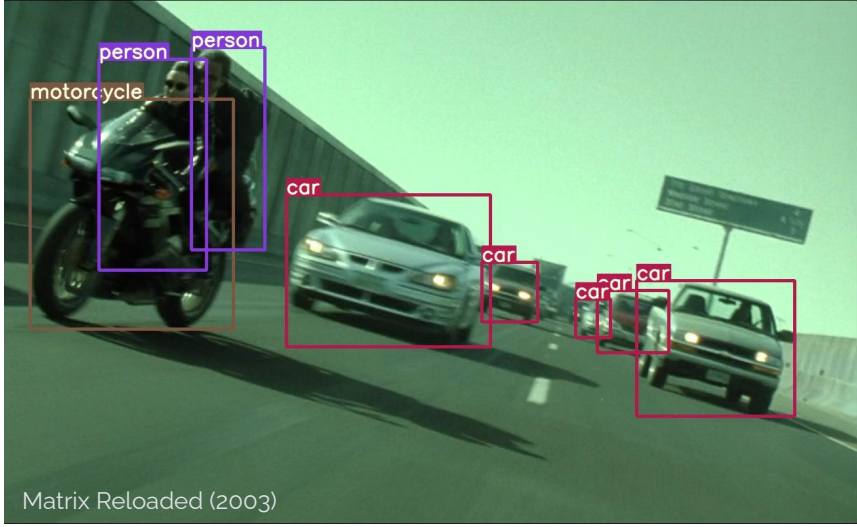
Obstacle Avoidance

And many more: Smart Home, Smart Farming, Wearable, Smartphone, Healthcare, ..,

Object Detection

Given an image, object detection models perform two main computer vision tasks

- **Object Recognition**
- **Image Classification**



Two Stage

- **FasterRCNN** ~100M params

Single Stage

- **SSD** ~ 4.2M params
- **Yolo v5** 2M up to 40M params

Output

- **Bounding Boxes with detected classes and confidence**

NXP iMX8mp

- Quad-Core 1.8 Ghz arm based Cortex-A53 CPU
- NPU (Neural processing unit)
- Dual image signal processors (ISP) and two camera inputs
- H.265 Encode/Decode
- Linux OS based on Yocto Project

Evaluation Board

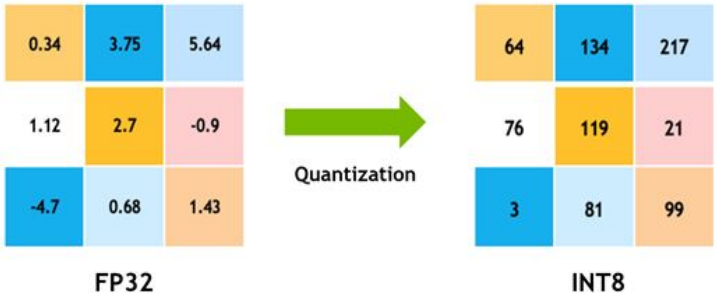
Optimizations: Quantization (Post Training)

After the training of a deep model, its weights are usually represented as 32 bit floating-point values.

Quantization consists in compressing a deep model, by mapping its weights to a lower precision value, **e.g FP32->INT8**

$$Q(x) = round(\frac{x}{scale\_factor}) - zero\_point$$

Scale factor: $\frac{Max(x) - Min(x)}{2^{nbit} - 1}$ **(Clipping Range)**



Clipping range calculation:

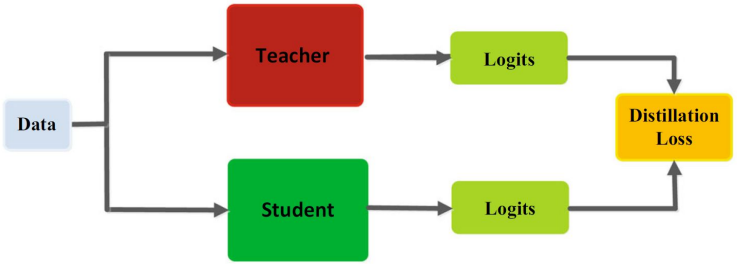
- Weights -> **static** clipping range calculation (weights are fixed after training)
- Activations -> **static** (with calibration data) or **dynamic** (online clipping range calculation during inference)

Optimizations: Knowledge Distillation

Transfer the knowledge from a larger (trained) model to a smaller one.
The goal is to improve student accuracy performance by replicating teacher behaviors during training phase.

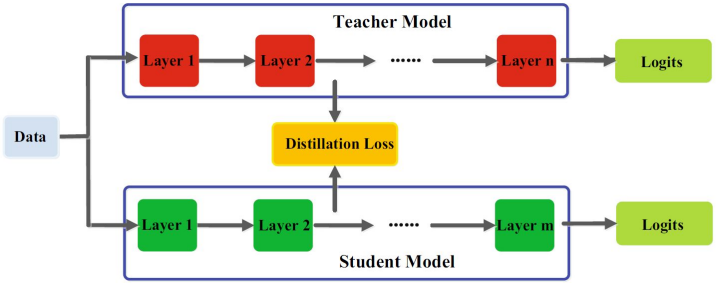
Output based

The student's training will only be affected by the output of the last layer of the teacher network.



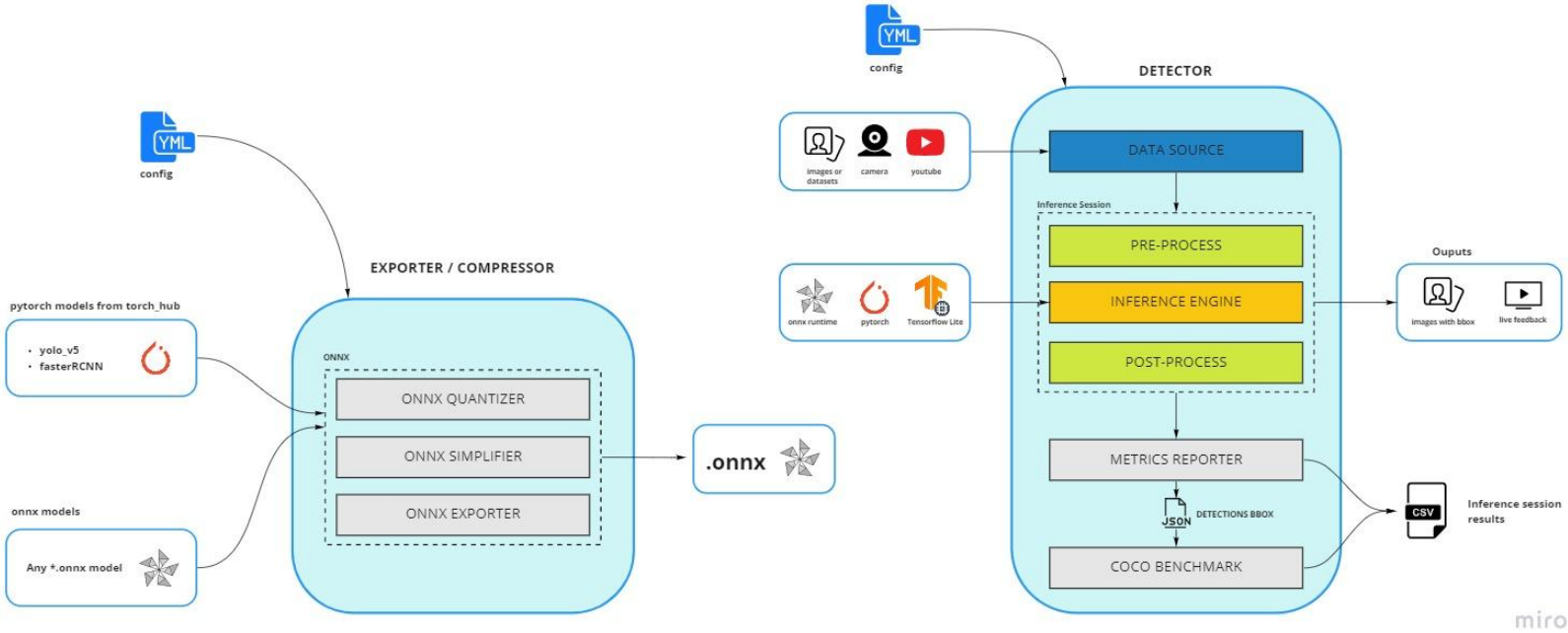
Feature based

The student will be influenced also by the behavior of the intermediate layers of the teacher.



Ivan Murabito

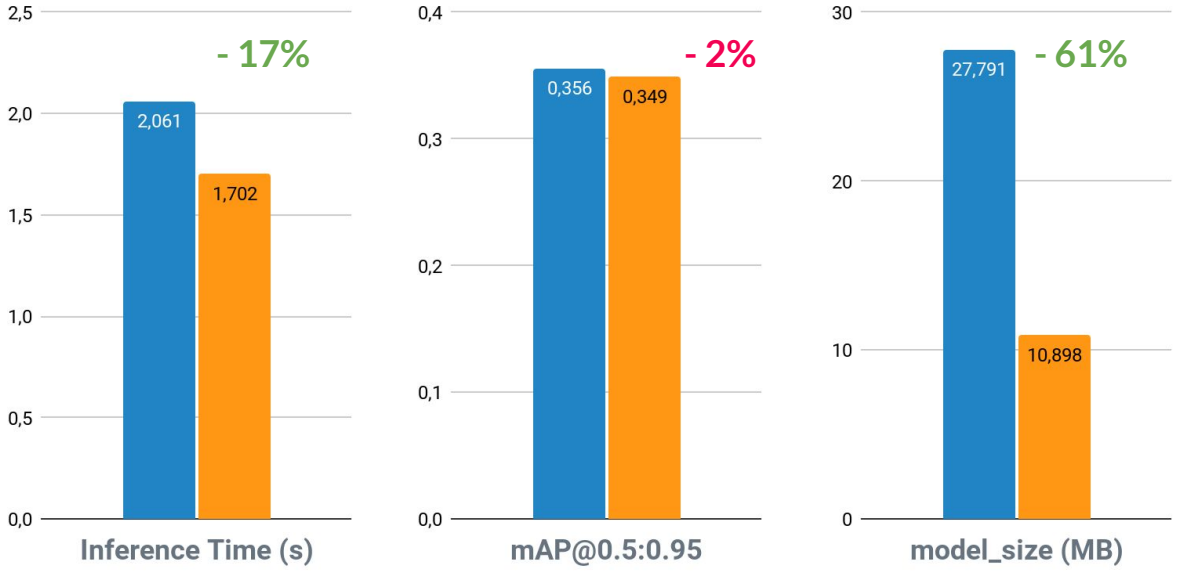
Model Compression Box



Ivan Murabito

Yolo v5s Quantization Results

Onnx post training dynamic quantization



FP32 Baseline

UINT8 Quantized

Results on target board (cpu)

- **430ms faster**
- From 27 to 11 MB
- **<0.01 mAP** degradation

mAP evaluated on coco dataset

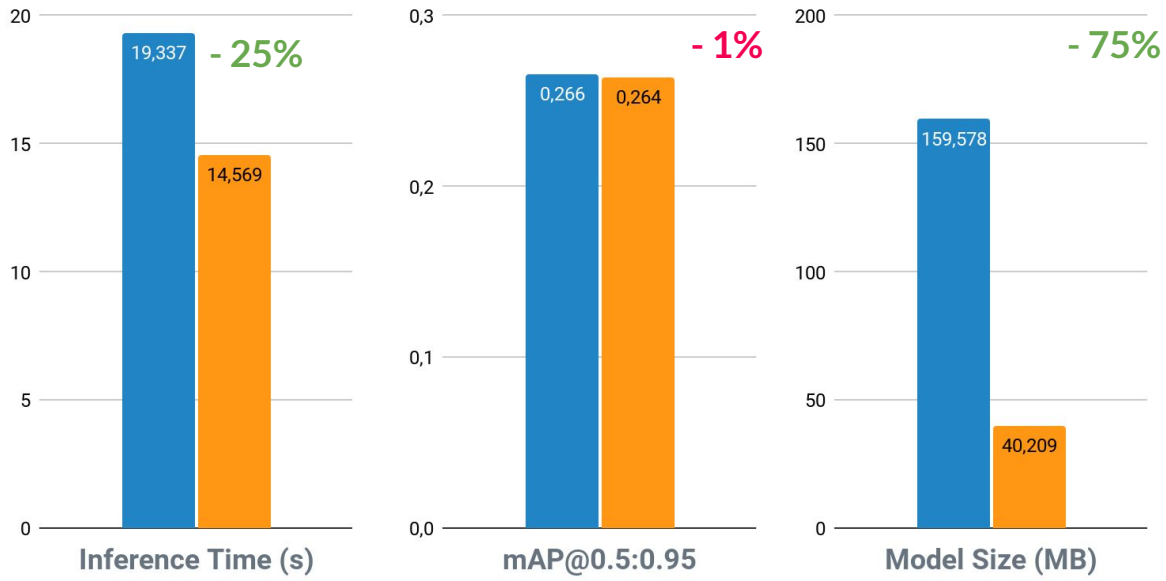
Ivan Murabito

Politecnico di Torino

FONDAZIONE links PASSION FOR INNOVATION

FasterRCNN Quantization Results

Onnx post training dynamic quantization



FP32 Baseline

UINT8 Quantized

Results on target board (cpu)

- **5s faster**
- **110MB Lighter**
- **<0.001 mAP** degradation

mAP evaluated on coco dataset

Ivan Murabito

Politecnico di Torino

FONDAZIONE links PASSION FOR INNOVATION

Knowledge Distillation Experiments

Teacher Networks

- Yolo v5 L (large)
- Yolo v5 M (medium)

Student Network

- Yolo v5 N (nano)

Dataset

- COCO (>200k images,80 classes)

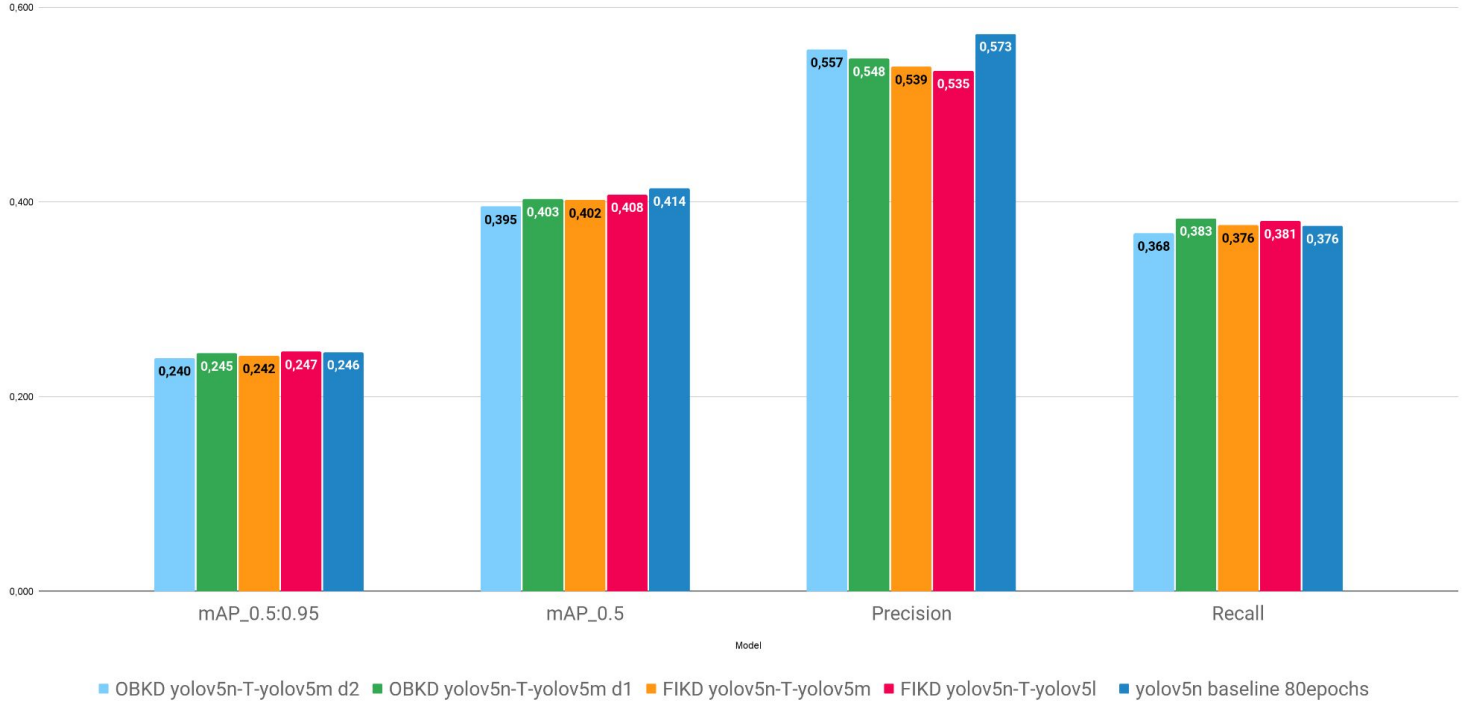
Knowledge Distillation types

- Output Based
- Feature Based

Approaches:

- Train student from scratch with knowledge distillation (80 epochs) of different combinations of networks and params
- Transfer Learning (pre-trained student with freezed backbone) and then small training with knowledge distillation

Knowledge Distillation Results



Ivan Murabito

Politecnico di Torino

FONDAZIONE links PASSION FOR INNOVATION

Final Consideration

Quantization:

- Easy to deploy, no need to retrain
- Model unaware, **plug and play optimization**
- OOB Excellent Results, low accuracy trade-off

Knowledge Distillation:

- can improve student accuracy performance, but not too much, it is a extreme optimization .
- need effort to find optimal parameters
- need to re-train the network

Future Works:

- Test Quantization-Aware-Training
- Test other combinations of teacher-student architectures,parameters and more epochs
- Test together knowledge distillation and quantization
- Test other embedded devices

Thanks!

Any questions?