WHERE INNOVATION BEGINS
DESIGN AUTOMATION CONFERENCE

Team CapyNet

System Design Contest, GPU Track

Linyan Yang, Melina Soysal
Supervisor: Prof. Ulf Schlichtmann

https://github.com/ly-muc/CapyNet-SDC-DAC2023

Executive Summary

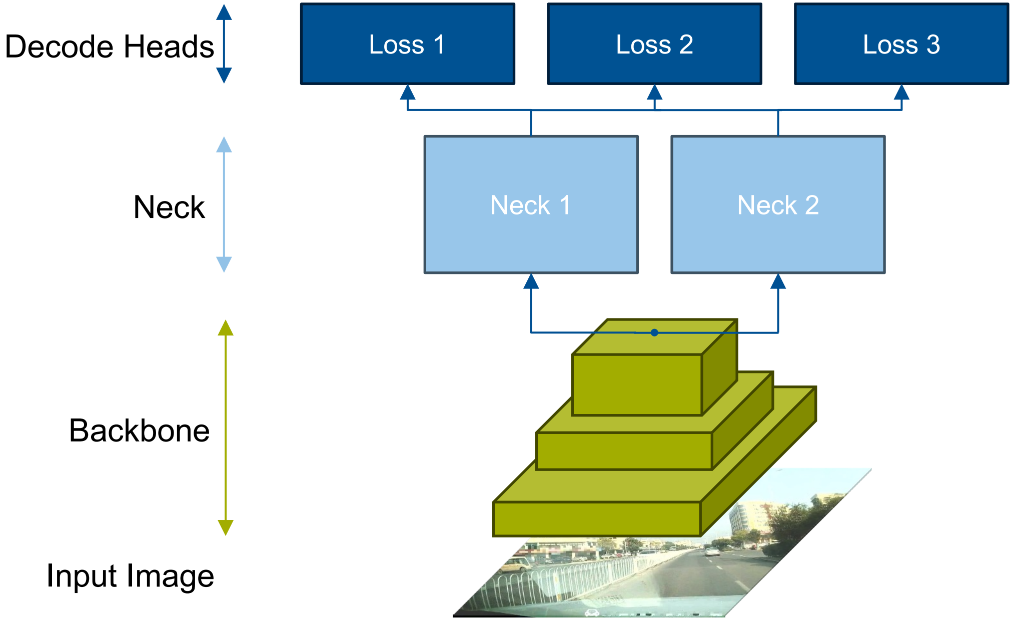
Problem Statement: Object Detection of seven distinct classes on an NVIDIA Jetson Nano platform

Key Idea: Focus on designing a small, robust, and fast network while co-optimizing for high recall and precision scores

Challenges:

1. Diverse dataset
2. Restricted time & compute during training
3. Limited hardware resources during inference

Object Detection Pipeline – Modular Approach



Decode Heads

Neck

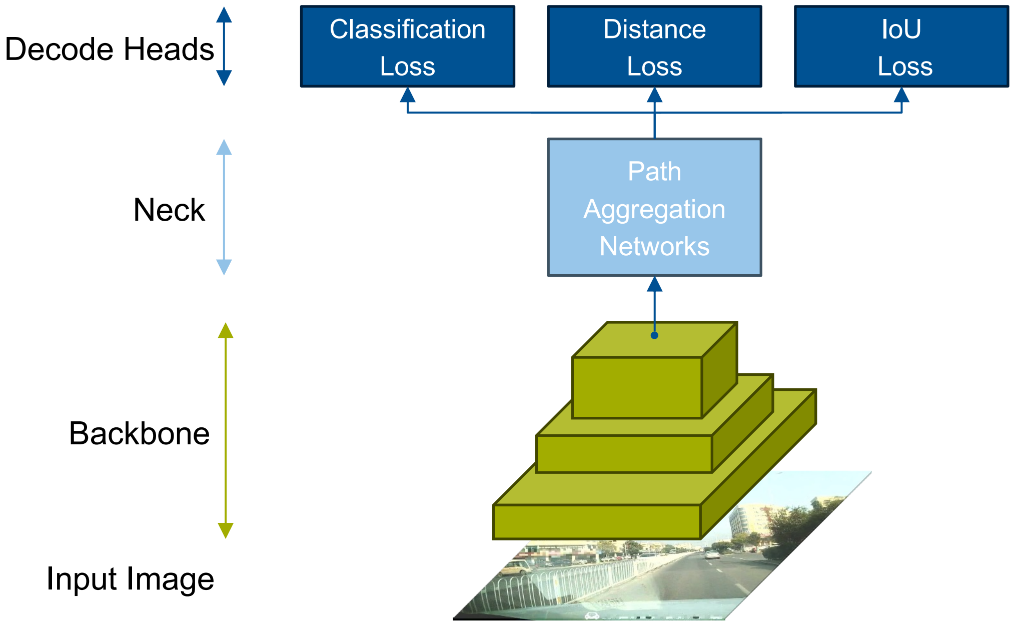
Backbone

Input Image

- generate predictions regarding the existence, location, and class of objects
- refine/combine features at different levels of abstraction
- capture spatial and semantic information

[1] Q. Zhao et al.: “M2Det: A Single-Shot Object Detector based on Multi-Level Feature Pyramid Network”, AAAI 2019

Object Detection Pipeline – Modular Approach



- **NanoDet Heads**: Loss functions suited for object detection
- **Path Aggregation Network:** preserves spatial information accurately
- **ShuffleNet**: tiny network with high extraction capability

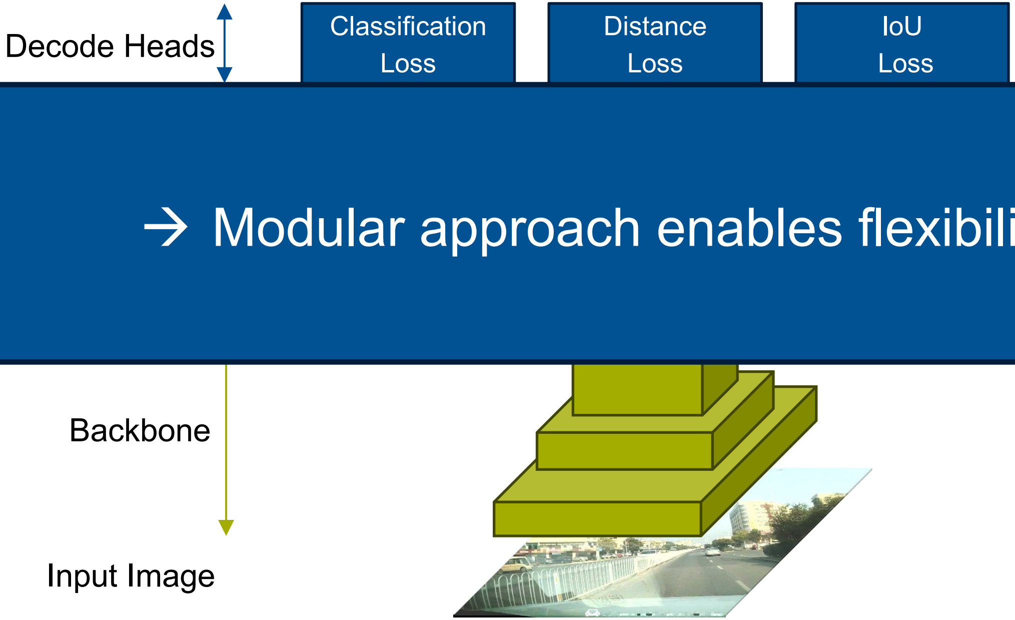
NanoDet

[2] R. Lyu: “NanoDet-Plus: Super fast and high accuracy lightweight anchor-free object detection model“, GitHub 2021

[3] S.Liu et al.: “Path Aggregation Network for Instance Segmentation”, CVPR 2018

[4] X. Zhang et al.:” ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices“, IEEE Access 2017

Object Detection Pipeline – Modular Approach



- **NanoDet Detection Heads**: Loss functions suited for object detection

extraction capability

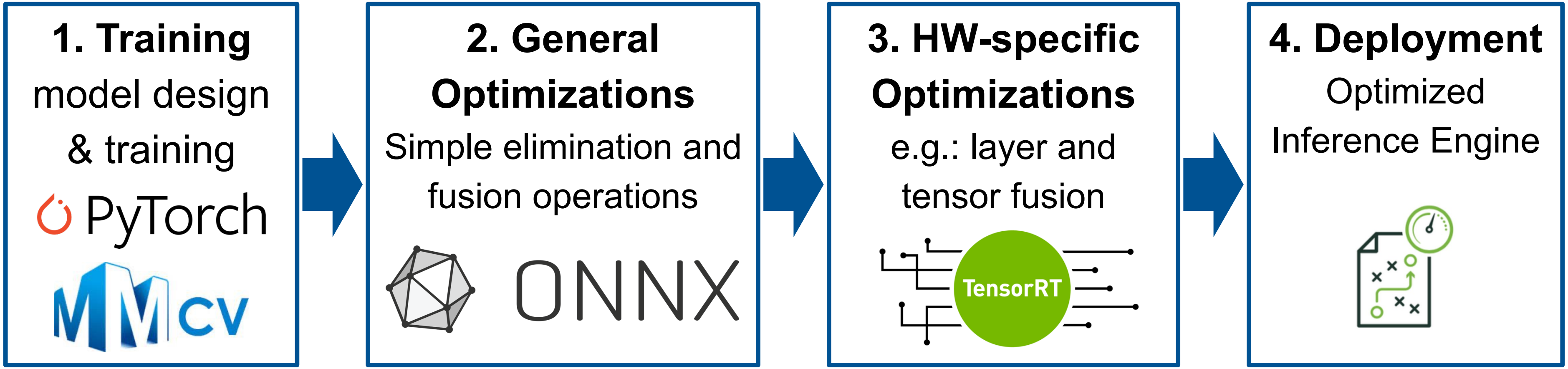
→ Modular approach enables flexibility, reusability, interchangeability

[2] R. Lyu: “NanoDet-Plus: Super fast and high accuracy lightweight anchor-free object detection model“, GitHub 2021

[3] S.Liu et al.: “Path Aggregation Network for Instance Segmentation”, CVPR 2018

[4] X. Zhang et al.:” ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices“, IEEE Access 2017

Training and Deployment Pipeline



1. Training
model design & training

2. General Optimizations
Simple elimination and fusion operations

3. HW-specific Optimizations
e.g.: layer and tensor fusion

4. Deployment
Optimized Inference Engine

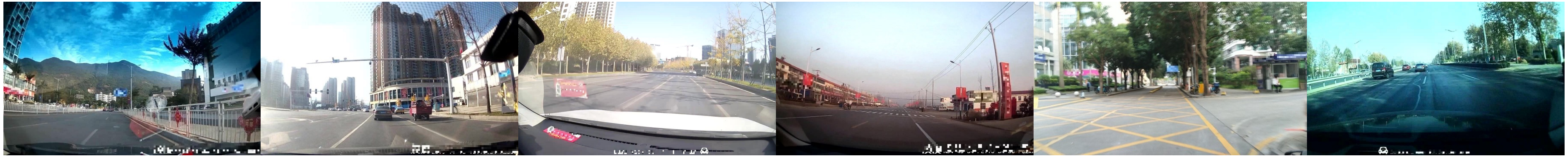
Challenges

1 **Diverse Dataset**

2 Restricted Time & Compute (Training)

3 Limited HW Resources (Inference)

Challenge 1: Diverse Dataset



- Different angles/viewpoints
- Different resolutions and color spectrums
- Class imbalance
- Inconsistent labels

→ Task requires a **robust** network

→ Choose a **keypoint-based detector** decode head as they are relatively more **robust to natural corruptions** than other detectors

Challenges

1 Diverse Dataset

2 **Restricted Time & Compute (Training)**

3 Limited HW Resources (Inference)

Challenge 2: Restricted Time & Compute (Training)

Accuracy

Quantifies the detection performance of the model

MAC Operations

Design parameter determining model size and influencing inference speed

Parameters

Model size limited by Jetson Nano

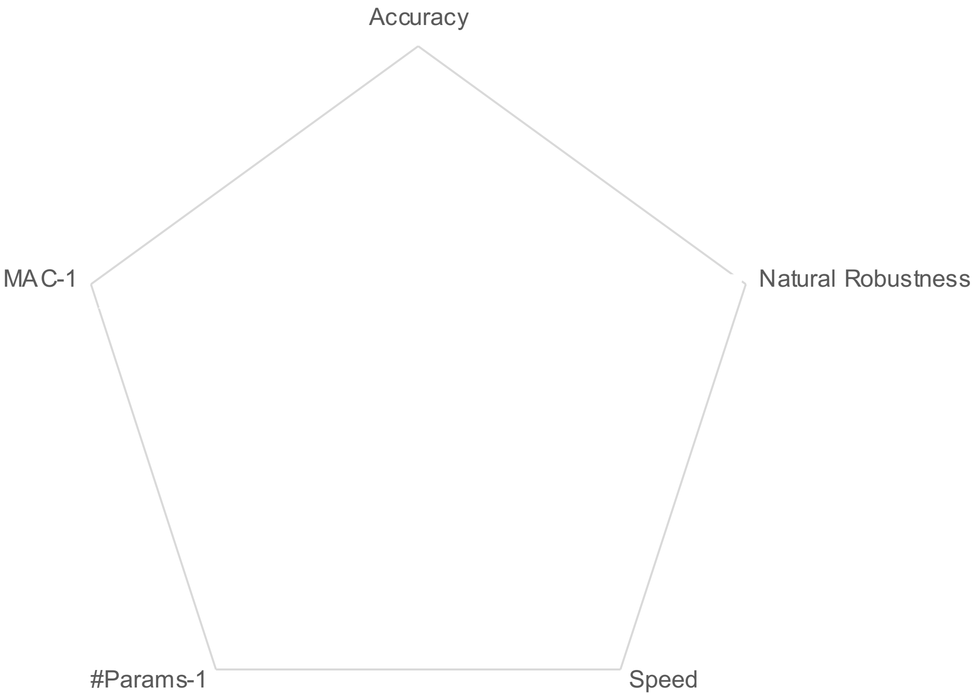
Speed

Inference Speed as important evaluation metric

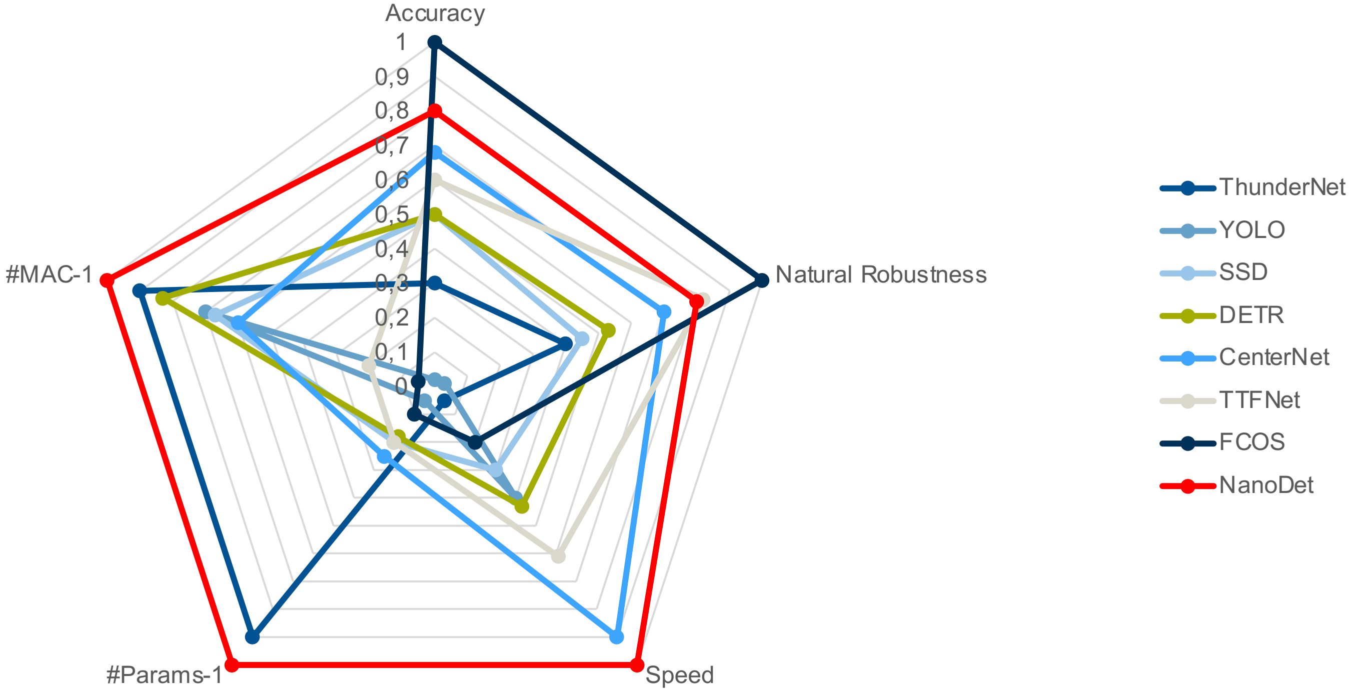
Natural Robustness

Robustness towards natural corruptions and variations

Challenge 2: Restricted Time & Compute (Training)

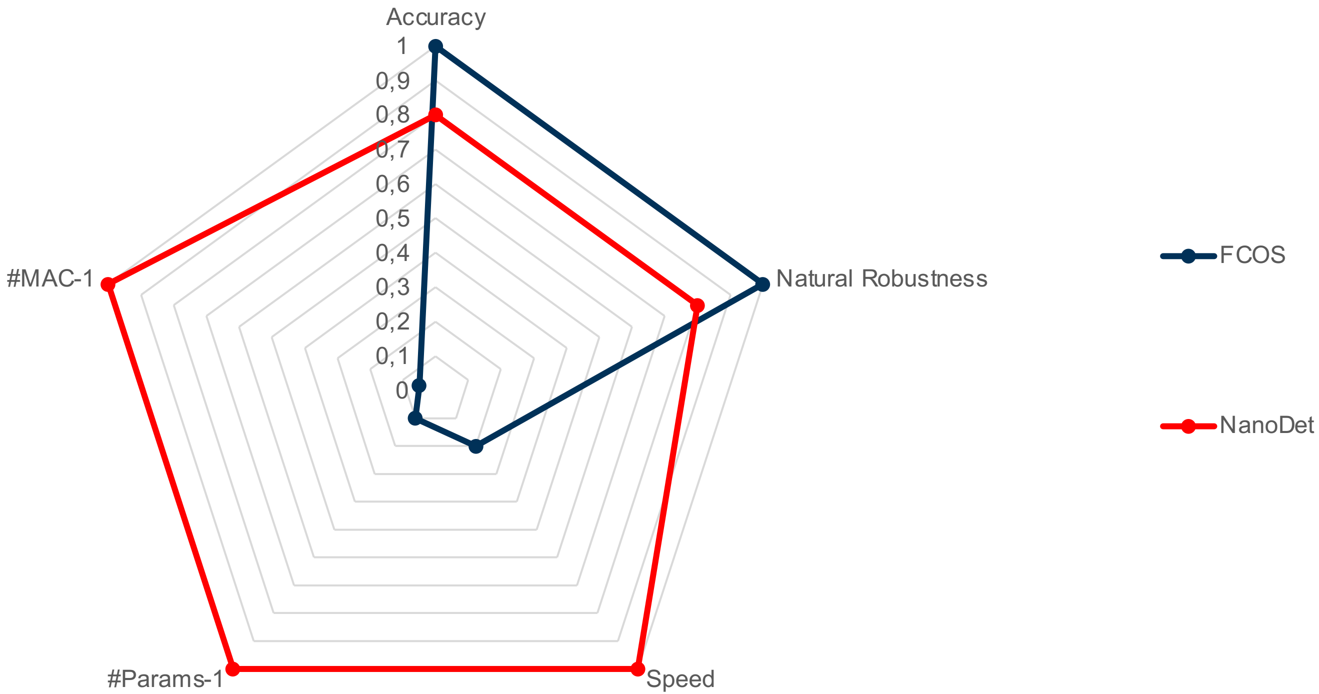


Challenge 2: Restricted Time & Compute (Training)



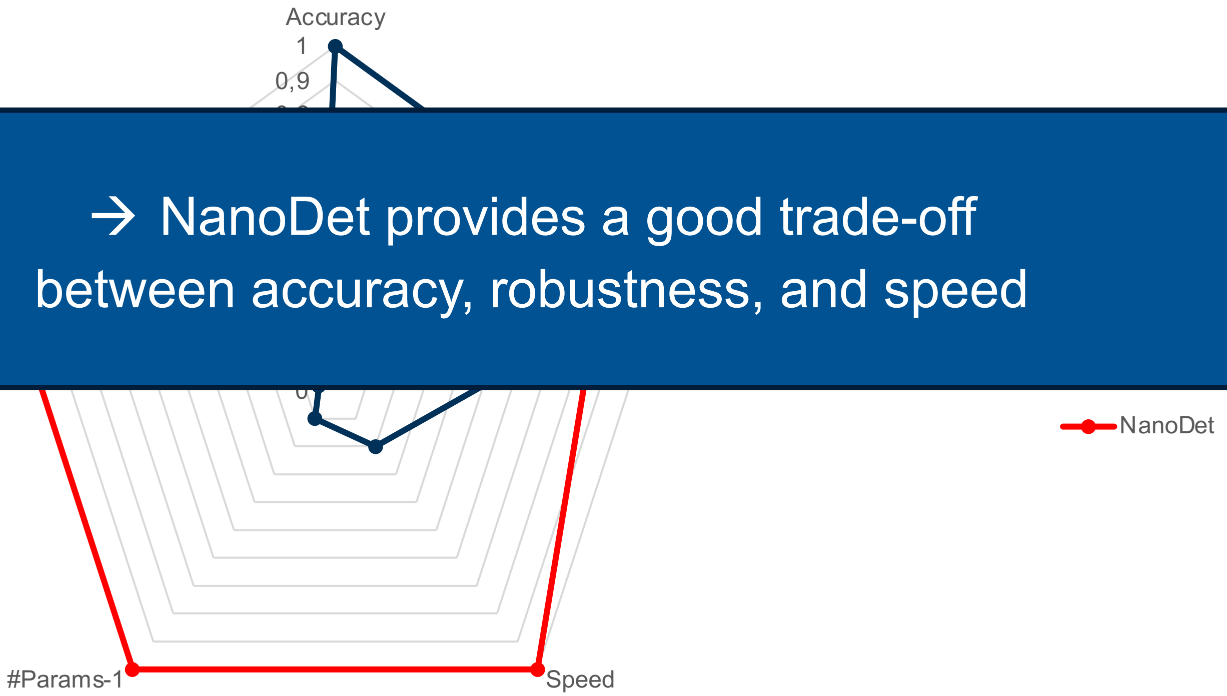
[5] E. Arani et al.: “A Comprehensive Study of Real-Time Object Detection Networks across Multiple Domains: A Survey”, TMLR 08/2022

Challenge 2: Restricted Time & Compute (Training)



[5] E. Arani et al.: “A Comprehensive Study of Real-Time Object Detection Networks across Multiple Domains: A Survey”, TMLR 08/2022

Challenge 2: Restricted Time & Compute (Training)



→ NanoDet provides a good trade-off between accuracy, robustness, and speed

[5] E. Arani et al.: “A Comprehensive Study of Real-Time Object Detection Networks across Multiple Domains: A Survey”, TMLR 08/2022

Challenges

1. Diverse Dataset
2. Restricted Time & Compute (Training)
3. **Limited HW Resources (Inference)**

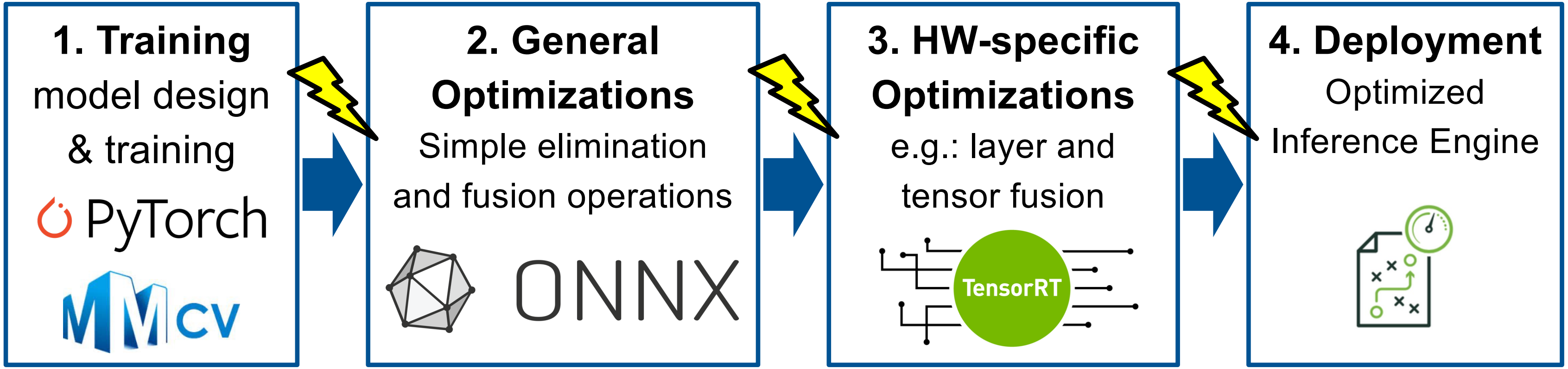
Challenge 3: Limited Hardware Resources (Inference)

During Model Selection and Training, we already considered:

- Model Size
- Inference Speed
- Computation Complexity

→ This restriction has been part of our design constraints

Challenge 3: Limited Hardware Resources (Inference)



1. Training
model design & training

2. General Optimizations
Simple elimination and fusion operations

3. HW-specific Optimizations
e.g.: layer and tensor fusion

4. Deployment
Optimized Inference Engine

Main challenges:

- Deployment pipeline compatibility issues
- Only smaller board compared to the target board available
  (Jetson Nano 2GB vs. Jetson Nano 4GB)

Conclusion & Takeaways

1. **Bootstrap**: leveraging work that is available open source and build on what others have done

2. **Minimum Viable Product**: prioritizing a running pipeline from training to deployment before applying any additional and detailed ideas during training

3. **Design Goals**: understanding and prioritizing different design parameters while being aware of possible trade-offs

Team CapyNet

System Design Contest, GPU Track
Linyan Yang, Melina Soysal
Supervisor: Prof. Ulf Schlichtmann

https://github.com/ly-muc/CapyNet-SDC-DAC2023

References

[1] Q. Zhao et al.: “M2Det: A Single-Shot Object Detector based on Multi-Level Feature Pyramid Network”, AAAI 2019

[2] R. Lyu: “NanoDet-Plus: Super fast and high accuracy lightweight anchor-free object detection model“, https://github.com/RangiLyu/nanodet, 2021

[3] S.Liu et al.: “Path Aggregation Network for Instance Segmentation”, CVPR 2018

[4] X. Zhang et al.:” ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices“, IEEE Access 2017

[5] E. Arani et al.: “A Comprehensive Study of Real-Time Object Detection Networks across Multiple Domains: A Survey”, TMLR 08/2022

[6] K. Chen et al.: “MMDetection: Open MMLab Detection Toolbox and Benchmark”, arXiv 2019