AI 2025
SUMMER SCHOOL
ai.uni-jena.de

You Only Look Once: Real-Time Object Detection with Neural Networks

Jamie Klöser

Faculty of Mathematics and Computer Sciences
Friedrich Schiller University Jena

Why Real-Time Detection? The Promise of Looking Once

Goal: detect and localize multiple objects **in real time** on consumer hardware. [4]

Why YOLO? Single-shot detector: predicts bounding boxes & classes in **one forward pass**, enabling high FPS. [3]

Poster contributions: crisp intuition of YOLO's pipeline; **mini-benchmark** of FPS vs. input resolution; wins & fails of YOLO

Applications: assistive robotics, AR, safety monitoring, logistics.

Fig. 1 — Cityscapes (Hero / Urban Szene): *Urban street scene with instance annotations. Reproduced from* **Cordts et al.**, 2016, **Cityscapes examples page / Fig. 1 teaser**. [1]

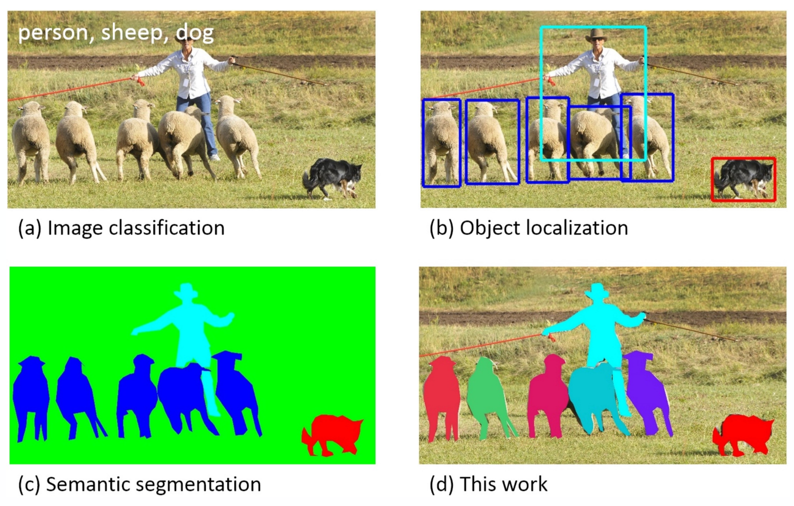

Fig. 2 *Examples from MS COCO illustrating objects in context. Reproduced from* **Lin et al.**, 2014, **Fig. 1**. [4]

How YOLO Sees: From Pixels to Predictions

The One-Pass Recipe: Backbone → Neck → Head

Backbone: convolutional feature extractor (downsampling; rich feature maps). [4]

Neck: multi-scale feature fusion (e.g., FPN/PAN-like) to detect small & large objects. [6]

Head: direct box regression + objectness + class scores per grid cell/anchor (anchor-free variants exist). [4]

Post-processing: NMS (Non-Maximum Suppression) removes duplicate boxes. [4]

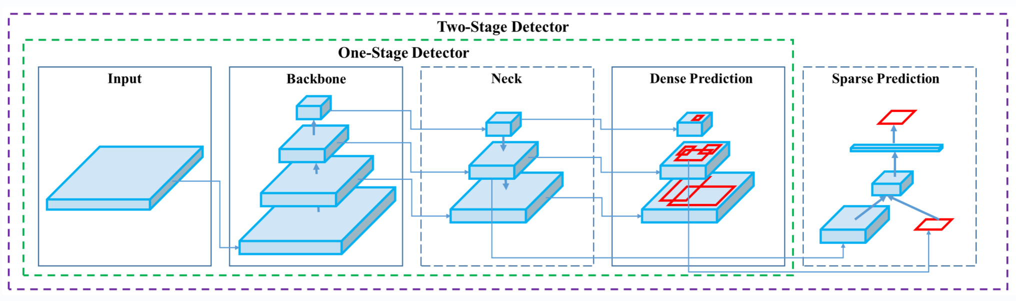

Fig. 3 — YOLOv4 Pipeline: *Single-pass detector, Backbone–Neck–Head with NMS. Reproduced from* **Bochkovskiy, Wang & Liao**, 2020, **Fig. 2**.

What the Network Optimizes: A Three-Part Loss

Box loss: distance between predicted and target boxes (IoU-family).

Objectness: is there an object?

Class: category probability. [8, 9]

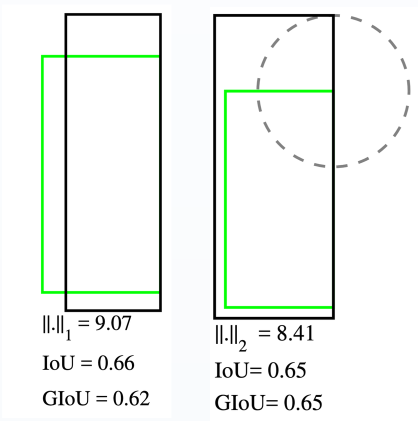

Fig. 4 & 5. — GIoU (IoU vs. GIoU Geometrie) *Bounding-box geometry: IoU vs. GIoU. Reproduced from* **Rezatofighi et al.**, 2019, **Fig. 1**.

Speed vs. Detail: The Resolution Trade-Off

Question: How does **input resolution** affect **speed** (FPS) and qualitative detection quality?

Models: YOLOv3/YOLOv4 families (single-shot detectors).

Resolutions: 320–640 px (square).

Observation: Higher input sizes generally **increase AP** (accuracy) but **reduce FPS**. For example:

YOLOv3 reports **22 ms at 320×320 (~45 FPS)** on Titan X (28.2 mAP); **YOLOv4** reports **~65 FPS** on Tesla V100 with higher AP than YOLOv3. Vendor docs also note **trades accuracy vs. speed** during inference. [3]

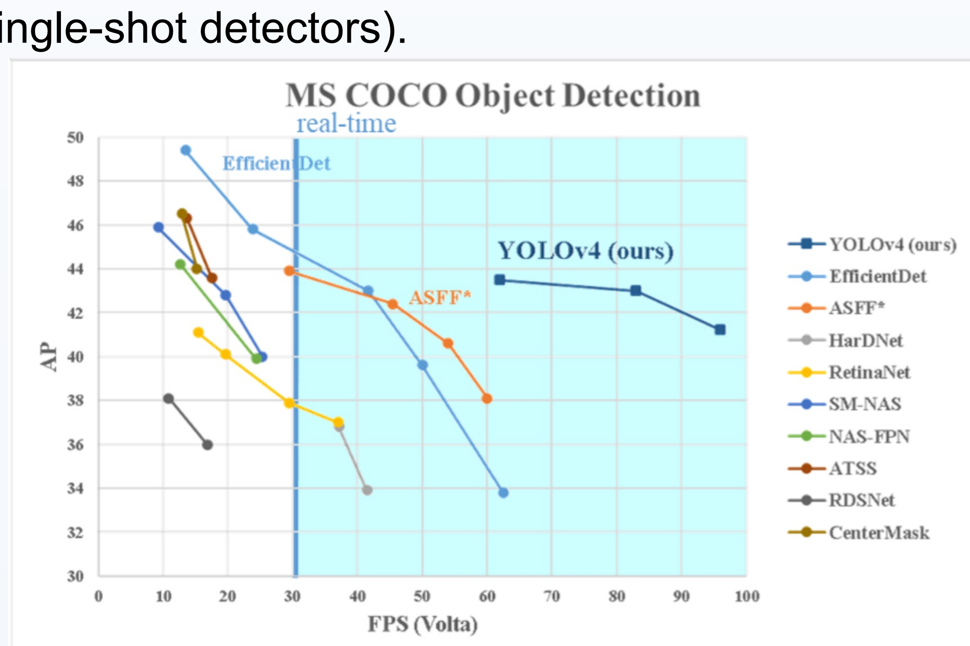

Fig. 6 — YOLOv4 Speed–Accuracy *Speed–accuracy trade-off on COCO (real-time region highlighted). Reproduced from* **Bochkovskiy, Wang & Liao**, 2020, **Fig. 1**.

What Others Observed: Reported FPS at Common Sizes

Speed trend (YOLOv4 example): ~54 FPS @416, ~43 FPS @512, ~33 FPS @608 on Pascal/Volta-class GPUs; accuracy (AP) increases with input size. *Hardware-dependent.*

Qualitative trend: higher resolution → better small-object detection; lower resolution → higher FPS.

Stability: lighting and motion blur impact detection consistency. [4, 10]

Model	#Param.	FLOPs	Size	FPS (V100)
YOLOv7-tiny-SiLU	6.2M	13.8G	640	273
YOLOv7	36.9M	104.7G	640	118
YOLOX-S	9.0M	26.8G	640	102
YOLOv7-W6	70.4M	360.0G	1280	80
YOLOv7-E6	97.2M	515.2G	1280	54

Table 1. *Excerpt from* ***YOLOv7, Table 9*** *on* ***V100****. Columns shown: Model, #Params, FLOPs, Input Size,* ***FPS (V100)****. Results are hardware-dependent.* **Reproduced from Wang et al., CVPR 2023, Table 9.**

Making It Work: Practical Choices & Trade-Offs

Trade-off: Throughput vs. detail. Choose resolution based on task needs (e.g., tiny objects require ≥480/640).

Bottlenecks: pre/post-processing and NMS can dominate at high FPS; CPU-only runs are NMS-limited.

Generalization: pretrained weights perform well on common objects; domain shift (lighting, unusual classes) can degrade results.

Practical tips: fix exposure, avoid motion blur, and pin confidence/IoU thresholds for fair comparisons. [8, 9, 13]

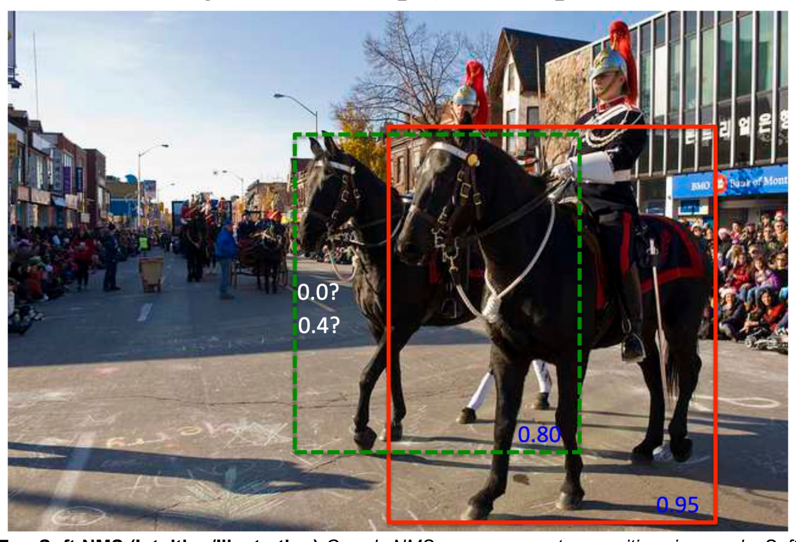

Fig. 7 — Soft-NMS (Intuition/Illustration) *Greedy NMS can suppress true positives in crowds; Soft-NMS mitigates this. Reproduced from* **Bodla et al.**, 2017, **Fig. 1**.

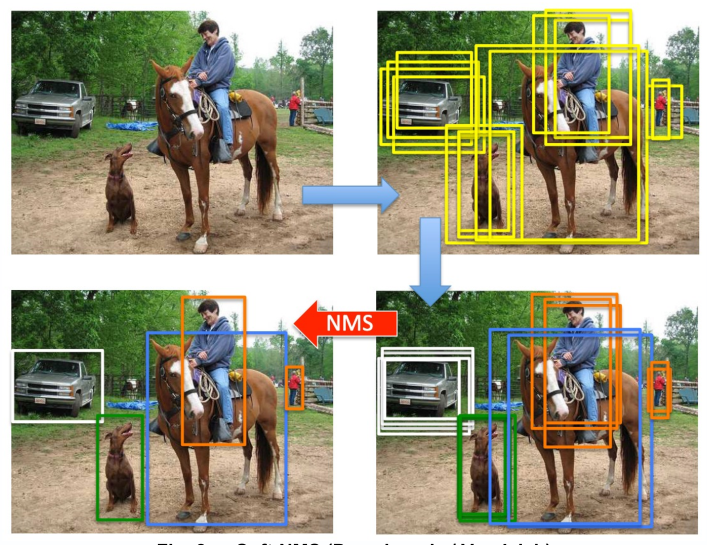

Fig. 8 — Soft-NMS (Pseudocode / Vergleich) *Soft-NMS algorithm vs. Greedy NMS. Reproduced from* **Bodla et al.**, 2017, **Fig. 3**.

Mind the Gaps: Limits Today, Easy Wins Tomorrow

Small/far objects remain hard at low input sizes.

Calibration: confidence ≠ probability; be cautious interpreting scores.

Future work: lightweight tracking (SORT/DeepSORT), model **quantization/ pruning** for CPU speedups, and **tiny finetune** on a custom 3-5-class desk dataset. [2, 11, 14]

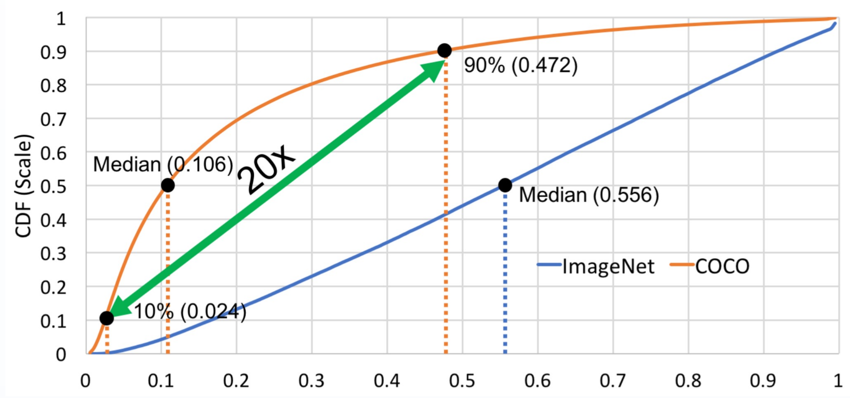

Fig. 9 — SNIP (kleine Objekte / Scale-Mismatch) *Distribution of RoI scales showing prevalence of tiny objects. Reproduced from* **Singh & Davis**, 2018, **Fig. 1**.

Where YOLO Wins & Fails: Successes, Failure Modes, Quick Fixes

Quick fixes (no retraining):

Input size ↑ (e.g., 480 → 640) for small objects (accept lower FPS).

Thresholds: tune **confidence** & **IoU**; try **Soft-NMS/DIoU-NMS**.

Temporal smoothing: lightweight tracking (SORT/DeepSORT) to stabilize boxes.

Pre-processing: fix exposure/ISO; denoise or deblur lightly.

Mini-finetune: a few classes from your domain.

Works great when… large/near objects, clear contrast, moderate motion.

Struggles when… small/far objects, heavy occlusion, **low light/over-exposure**, motion blur, unusual viewpoints, cluttered scenes.

Why: fewer pixels per object, aliasing, weak features, NMS suppressing true boxes in crowds.

[4, 8, 9, 10, 14]

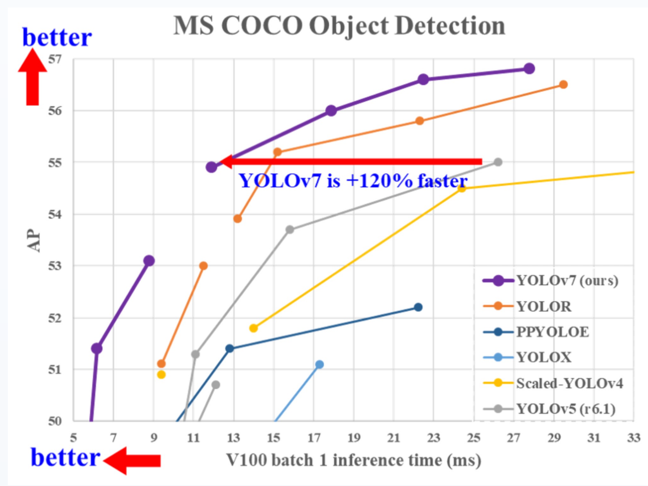

Fig. 10 — YOLOv7 (Real-time Detectors Comparison) *Comparison with other real-time object detectors. Reproduced from* **Wang, Bochkovskiy & Liao**, 2023, **Fig. 1**.

References

[1] M. Cordts *et al.*, "The Cityscapes Dataset for Semantic Urban Scene Understanding," **CVPR**, 2016. (incl. examples page/teaser)
[2] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft COCO: Common Objects in Context," **ECCV**, 2014.
[3] J. Redmon, A. Farhadi, "YOLOv3: An Incremental Improvement," **arXiv:1804.02767**, 2018.
[4] A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," **arXiv:2004.10934**, 2020.
[5] C.-Y. Wang, A. Bochkovskiy, H.-Y. M. Liao, "Scaled-YOLOv4: Scaling Cross-Stage-Partial Network," **arXiv:2011.08036**, 2020/2021.
[6] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, "Feature Pyramid Networks for Object Detection," **CVPR**, 2017.
[7] H. Rezatofighi *et al.*, "Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression," **CVPR**, 2019.
[8] Z. Zheng *et al.*, "Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression," **AAAI**, 2020. (incl. DIoU-NMS)
[9] N. Bodla, B. Singh, R. Chellappa, L. S. Davis, "Soft-NMS — Improving Object Detection With One Line of Code," **ICCV**, 2017.
[10] C.-Y. Wang, A. Bochkovskiy, H.-Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," **CVPR**, 2023.
[11] B. Singh, L. S. Davis, "An Analysis of Scale Invariance in Object Detection (SNIP)," **CVPR**, 2018.
[12] J. Huang *et al.*, "Speed/Accuracy Trade-Offs for Modern Object Detectors," **CVPR**, 2017.
[13] **Ultralytics Documentation**, *Inference settings* & imgsz *accuracy–speed trade-off*, accessed 2025.
[14] S. Hwang *et al.*, "Multispectral Pedestrian Detection Benchmark," **CVPR**, 2015. (KAIST)

Contact

Jamie Klöser
Friedrich Schiller University Jena, Faculty of Mathematics and Computer Sciences
Email: **jamie.kloeser@uni-jena.de**
Matrix: @jamiekloeser:matrix.org