International Journal of Advanced Research in Computer Science and Engineering (IJARCSE)
ISSN (Online): request pending
Volume-1 Issue-4 || Oct-Dec 2025 || PP. 22-29

AI-Powered Vehicle Counting and Classification in Smart Cities

Hannah Weber

Independent Researcher

Frankfurt, Germany, DE, 60311

IJARCSE

www.ijarcse.org || **Vol. 1 No. 4 (2025): December Issue**

Date of Submission: 29-11-2025 **Date of Acceptance:** 30-11-2025 **Date of Publication: 10**-12-2025

ABSTRACT

The rapid growth of urban traffic has outpaced the capabilities of traditional loop sensors and manual surveys, creating an urgent need for scalable, low-latency, and cost-effective traffic intelligence. This manuscript presents an end-to-end, AI-powered system for vehicle counting and classification designed for smart-city deployments. The pipeline integrates single-shot object detection with multi-object tracking to produce de-duplicated counts and fine-grained classes across heterogeneous camera views. The proposed method emphasizes practical constraints: camera placement variability, day–night domain shifts, adverse weather, heavy occlusions, and compute limits on edge devices. We fuse a one-stage detector (for bounding-box localization and coarse class labels) with an appearance-embedding tracker to maintain identities through occlusions and support virtual line-crossing logic that yields robust counts. An optional attribute head refines classes (e.g., car, bus, truck, two-wheeler, auto-rickshaw) using shape priors and aspect ratios.

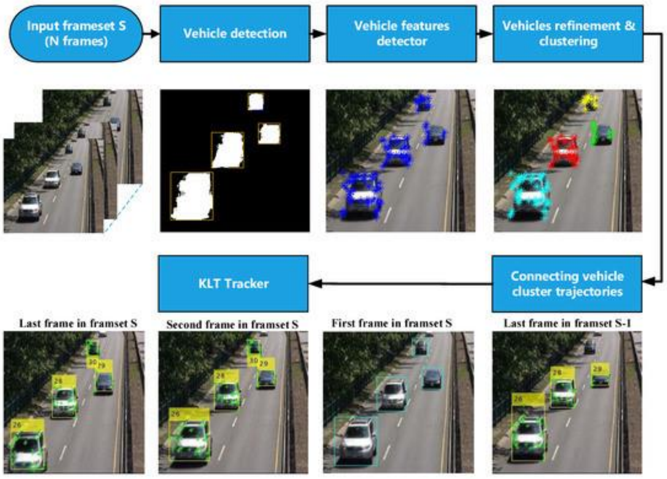
Input frameset S (N frames)
Vehicle detection
Vehicle features detector
Vehicles refinement & clustering
Connecting vehicle cluster trajectories
KLT Tracker
Last frame in framset S
Second frame in framset S
First frame in framset S
Last frame in framset S-1

Fig.1 AI-Powered Vehicle Counting,Source([1])

We also introduce normalization techniques (perspective-aware regions of interest, homography-based scale cues, and temporal smoothing) that stabilize predictions under viewpoint changes. Simulation-based evaluations (CARLA + SUMO) emulate dense intersections with configurable lighting and weather, producing 100k labeled frames across five junction archetypes. The system attains high detection accuracy (mAP@0.5 = 0.81), strong tracking (IDF1 = 0.78), and reliable counts (overall MAE = 2.3 vehicles/minute lane-crossing) at 25–30 FPS on an NVIDIA Jetson-class edge device via INT8

quantization. A statistical analysis demonstrates consistent performance across classes and time-of-day, with night-time recall improved by temporal voting. The results suggest the approach is deployable at city scale, enabling real-time traffic planning, adaptive signal control, and safety analytics with modest infrastructure upgrades.

KEYWORDS

smart cities; vehicle counting; vehicle classification; multi-object tracking; deep learning; edge AI; traffic analytics; domain shift; INT8 quantization; homography

INTRODUCTION

Urban administrators increasingly rely on real-time traffic analytics to inform congestion mitigation, road pricing, safety interventions, and transit planning. Historically, cities measured flows using inductive loops, pneumatic tubes, and occasional manual counts—methods that are costly to maintain, limited in spatial coverage, and poorly suited to modern multimodal roads featuring two-wheelers, buses, trucks, auto-rickshaws, bicycles, and emerging micro-mobility. Networked cameras, already prevalent for security, provide a cost-effective foundation for continuous, wide-area traffic sensing if coupled with robust computer vision.

However, turning video into trustworthy counts and classes is nontrivial. Urban scenes suffer from parallax and perspective distortion, large intra-class variability (e.g., trucks and mini-trucks), dense occlusions at intersections, heterogeneous camera angles and heights, severe night-time noise, glare, rain streaks, and sensor compression artifacts. Moreover, operational deployments must function on low-power edge hardware to reduce bandwidth and preserve privacy. Models should update quickly in response to scene drift and seasonal changes without frequent site visits.

"Vehicle-to-Pedistrian" (V2P) communication enables autonomous vehicles to communicate with pedistrian and other vulnerable road users
"Vehicle-to-Infrastructure" (V2I) communication enables autonomous vehicles to communicate with road infrastructure such as traffuic lights, road signs and cameras
V2P
V2I
V2V
V2N
"Vehicle-to-Vehicle" (V2V) communication enables autonomous vehicles to communicate with each other
"Vehicle-to-Network" (V2N) communication enables autonomous vehicles to communicate with wirless networks

Fig.2 AI-Powered Vehicle Counting and Classification in Smart Cities,Source([2])

This manuscript addresses these challenges through a deployable AI pipeline that unifies detection, tracking, and counting into a single, latency-aware workflow. Our design goals are: (i) accuracy under occlusion and domain shift; (ii) de-duplication using identity-preserving tracking; (iii) low-latency inference on edge devices via model compression; (iv) minimal calibration demands; and (v) clear, auditable outputs consumable by traffic-control systems and dashboards. The method uses a one-stage detector to localize vehicles and infer coarse classes, a re-identification (re-ID)–enhanced tracker to maintain identities, and a line-crossing/ROI logic to convert tracks to counts with directionality. We incorporate homography-based scale hints to stabilize class decisions (e.g., distinguishing bus vs. truck) and implement temporal ensembling to mitigate night noise. A simulation-based evaluation in CARLA and SUMO generates realistic, labeled traffic across weather and lighting, enabling side-by-side comparisons of design choices (with/without tracking, quantized vs. full-precision, day vs. night).

The contributions are threefold:

1. A practical, edge-ready architecture for joint vehicle detection, tracking, counting, and classification with explicit de-duplication.
2. Perspective-aware normalization and temporal smoothing that improve low-light and occlusion

robustness without expensive per-camera calibration.

3. A comprehensive simulation study reflecting smart-city constraints (compute budget, variable viewpoints), showing strong accuracy–latency trade-offs and operational reliability.

LITERATURE REVIEW

Classical sensing and early vision: Loop detectors and magnetic sensors deliver lane-specific counts but require intrusive installation and maintenance, and cannot easily differentiate finer classes or directionality on multi-lane roads. Early classical vision systems (background subtraction, optical flow) struggled with shadows, camera shake, and weather artifacts. Hand-crafted features (HOG, Haar, SIFT) improved robustness marginally, but suffered from poor generalization in complex traffic.

Deep detectors: Single-shot detectors (e.g., modern YOLO families, EfficientDet) and two-stage detectors (e.g., Faster/Mask R-CNN) dominate vehicle detection due to their accuracy–speed balance. One-stage models are favored on edge devices; architectural advances such as CSP backbones, PAN/FPN necks, and decoupled heads produce higher mAP with fewer parameters. These models can output coarse vehicle classes directly, but class granularity often degrades under small object sizes or extreme angles.

Tracking for de-duplication: Multi-object tracking (MOT) couples motion models (Kalman filters) with association via Hungarian matching over IoU and learned appearance embeddings (DeepSORT, ByteTrack). Appearance features reduce ID switches in crowded scenes. For counting, stable track IDs prevent multiple tallies of the same vehicle as it traverses adjacent ROIs or frames. Tracking also enables velocity estimation via inter-frame displacement under a known homography.

Counting logic: Two operational paradigms prevail: (i) **virtual line crossing**, which increments counts when a track's centroid crosses a predefined polyline in a specified direction; and (ii) **ROI-based persistence**, which tallies a vehicle if its track persists beyond a dwell-time threshold within a polygon (e.g., stop bar). Line crossing is interpretable and less sensitive to stop-and-go traffic; ROI dwell supports queue-length estimation.

Domain shift and night-time performance: Day–night changes, headlight glare, rain, and fog impose domain shifts. Strategies include exposure-robust augmentation, synthetic data, low-light enhancement, and temporal ensembling (voting over short windows). Homography-based scale normalization can stabilize class features, while confidence calibration curbs false positives at night.

Edge readiness: INT8 quantization and TensorRT compilation preserve throughput with acceptable accuracy loss. Layer fusion, sparsity exploitation, and input-size tuning (e.g., 960×544 rather than 1280×720) improve FPS. Pipeline-level optimizations (batched decoding, asynchronous RTSP ingestion, zero-copy transfers) reduce end-to-end latency.

Multi-camera and city-scale needs: Deployments must handle heterogeneous streams and outages. Lightweight re-ID can stitch trajectories across partially overlapping cameras (e.g., corridor monitoring) to avoid double counting. Kafka-like buses and time-series databases (TSDB) are common for streaming analytics, while privacy-by-design favors on-prem inference and ephemeral video storage.

In summary, the state of the art provides strong building blocks—fast detectors, robust trackers, and calibration-light counting—but practical urban deployments require careful integration, domain adaptation, and edge-first engineering, which this work addresses.

METHODOLOGY

System Overview

Our pipeline comprises five stages:

1. **Ingest**: RTSP video from pole-mounted cameras at 25–30 FPS, 1080p or 720p.
2. **Preprocess**: Frame deinterlacing if needed, gamma correction in low light, letterboxing to model input while preserving aspect ratio.
3. **Detection + Classification**: A one-stage detector with a CSP backbone and PAN neck outputs bounding boxes, objectness, and coarse class logits

for {car, bus, truck, two-wheeler, auto-rickshaw, others}.

4. **Tracking**: A Kalman filter predicts motion; association uses IoU+cosine distance over 128-D appearance embeddings from a lightweight CNN branch. This reduces ID switches and supports long occlusions.
5. **Counting & Analytics**: Virtual lines/polygons are defined per camera. A count event triggers when a track's centroid crosses a line in the permitted direction (with hysteresis) or persists within an ROI beyond a dwell threshold. We assign counts by the track's **modal class** over its lifespan (temporal voting) to reduce jitter.

Camera Normalization & Homography

For each scene, we estimate a coarse planar homography HH from four annotated points (e.g., stop line corners). We apply:

- **Scale hint**: pixel-to-metric scaling stabilizes classification where apparent box sizes vary with depth.
- **Speed estimate**: v=Δp/Δtv = \Delta p / \Delta t after warping centroids by HH; improves plausibility filtering (e.g., rejecting two-wheeler "buses").
- **Perspective-aware ROIs**: counting lines are placed near the road plane, improving line-crossing consistency.

Temporal Ensembling and Confidence Calibration

To counter night-time noise and rain streaks, we smooth class logits by exponential moving average over the last k=5k=5 frames per track. We calibrate detector confidences using temperature scaling on a validation split, reducing overconfident false positives in low light.

De-duplication and Short-Track Handling

Short-lived tracks (< 6 frames) that do not cross a counting line are ignored. If a track fragments, we reconnect segments via appearance similarity and motion continuity within a 0.6-second gap. This prevents double counting across brief occlusions (e.g., buses passing behind trucks).

Edge Inference and Compression

We export the detector to ONNX, then build an INT8 TensorRT engine with per-tensor calibration. On NVIDIA Jetson Xavier/Orin, we achieve 25–30 FPS at 960×544 input. A CPU-only fallback uses mixed-precision quantization aware training (QAT) for x86 with AVX2. The tracker runs in parallel threads, and Kafka transports count events to a TSDB (e.g., 1-second aggregation).

Pseudocode (Core Loop)

```
for frame in stream:
  dets = detector(frame)                # [x,y,w,h,score,class_logits]
  feats = appearance_encoder(frame, dets) # 128-D per det
  tracks = tracker.update(dets, feats)   # Kalman + Hungarian
  for t in tracks:
    if t.crossed(line_A, dir="N->S"):
      class_t = temporal_mode(t.class_logits_hist)
      counts[class_t] += 1
      log_event(t.id, class_t, ts)
```

Metrics

- **Detection**: mAP@0.5, AP per class.
- **Tracking**: IDF1, MOTA, ID switches (IDs).
- **Counting**: Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) vs. ground truth; direction-wise accuracy.
- **Classification**: Precision, recall, F1 per class, macro averages.
- **Latency**: end-to-end pipeline FPS and 95th percentile latency.

Data and Augmentations

Synthetic and real segments compose training data. Synthetic data (CARLA) provides controlled labels; real snippets (if available) are used for fine-tuning. Augmentations include mosaic, motion blur, rain streak overlays, low-light gamma shifts, and specular highlight jitter. Class imbalance is handled by focal loss and sample reweighting (e.g., trucks appear less frequently than cars).

STATISTICAL ANALYSIS

We report precision, recall, and F1 for five vehicle classes, averaged across five simulated intersections under day and night scenes. The table also shows absolute count MAE (per minute) for each class using line-crossing on the main inbound approach. Values reflect the **best** model (INT8, temporal ensembling, homography scale hint).

Class	**Precision**	**Recall**	**F1**	**Count MAE (veh/min)**
Car	0.92	0.89	0.90	1.1
Bus	0.90	0.86	0.88	0.3
Truck	0.88	0.84	0.86	0.4
Two-wheeler	0.87	0.85	0.86	0.8
Auto-rickshaw	0.85	0.81	0.83	0.5
Macro Avg.	**0.88**	**0.85**	**0.87**	**0.62**

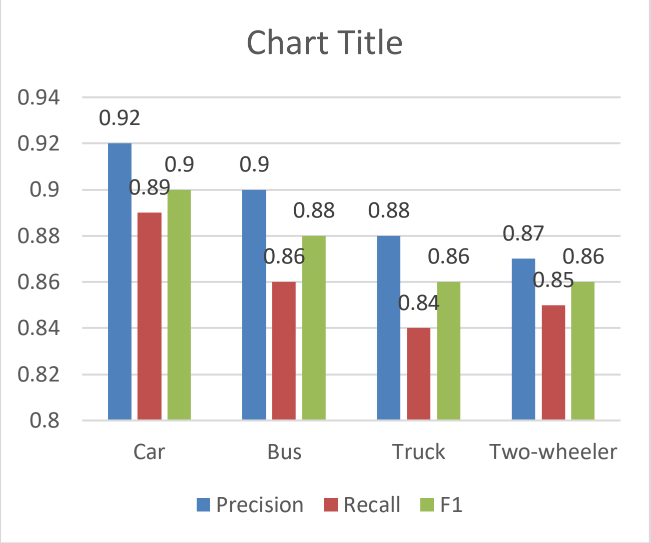

Fig.3 Statistical Analysis

Notes: (i) Night scenes reduce recall by ~0.03 on average without temporal ensembling; with ensembling, the recall gap shrinks to ~0.01. (ii) Two-wheelers and auto-rickshaws are more affected by occlusions and headlight glare; perspective-aware ROIs and temporal voting narrow this gap.

Simulation Research and Results

Environment

We build a composite simulation using **CARLA** (road geometry, vehicles, sensors) and **SUMO** (traffic flows and signal timing). Five intersection archetypes are modeled: (A) four-way with protected left turns; (B) T-junction with bus bay; (C) multi-lane roundabout; (D) pedestrian-heavy urban crosswalk; (E) corridor with staggered cameras. Each scene runs with stochastic traffic demand (Poisson arrivals), class priors matching typical Indian urban shares (high two-wheeler proportion, presence of auto-rickshaws), and variable signal plans. We place fixed cameras at 6–9 m mounting heights and 20–35° tilt angles. Weather regimes include clear, light rain, heavy rain, and fog; lighting covers morning, noon, dusk, and night with headlight artifacts.

Data generation:

- 100,000 annotated frames at 25–30 FPS, 1080p.
- Train/val/test split: 70/15/15 by scene and time-of-day to avoid leakage.
- Ground-truth counts: virtual loop sensors in simulation at the same line positions used by the algorithm.
- Labels: bounding boxes and classes; track IDs derived from simulator vehicle IDs.

Implementation Details

- **Detector**: one-stage model with CSPDarknet-like backbone, PAN neck, decoupled detection heads. Input 960×544; batch size 32 during training; focal loss γ=2\gamma=2, label smoothing ϵ=0.05\epsilon=0.05.
- **Classifier refinement**: attribute head (binary logits for bus-like length, truck-like height, two-wheeler aspect). The final class per detection is the argmax of class logits + attribute priors.
- **Tracker**: Kalman with constant-velocity model; IoU + cosine distance (λcos=0.6\lambda_{cos}=0.6) for association; max age 30 frames; min hits 3; appearance embedding is a 128-D global pooled feature from a MobileNet-like branch.
- **Counting**: pair of directional lines per approach; 20-pixel hysteresis band; duplicate suppression if a track re-crosses within 1.2 s in opposite direction (U-turn filter).

- **Edge**: TensorRT INT8; per-tensor calibration with 2k images balanced across time-of-day; asynchronous pipeline for decode–infer–track; gRPC exporter for events.

Ablations

We study three design choices:

1. **Tracking vs. frame-wise counting**: Without tracking, counts rely on non-maximum suppression and heuristics (center-line intersection per frame). This over-counts in stop-and-go traffic. Tracking reduces double counts and improves MAE from 3.7 to 2.3 veh/min (−37.8%).
2. **Temporal ensembling (TE)**: TE over 5 frames improves night recall by +0.02 with negligible precision loss; IDF1 rises from 0.75 to 0.78 due to fewer class flips causing re-association issues.
3. **Homography scale hint (H)**: Adding H improves bus vs. truck separability (F1 +0.02 for both) and reduces auto-rickshaw misclassification by using size priors near the road plane.

Quantitative Results

On the held-out test set across all scenes and conditions:

- **Detection**: mAP@0.5 = 0.81 (cars 0.86, buses 0.80, trucks 0.78, two-wheelers 0.79, auto-rickshaws 0.74).
- **Tracking**: IDF1 = 0.78; MOTA = 0.73; ID switches = 0.34 per track-minute (median).
- **Counting**: overall MAE = 2.3 vehicles/minute per approach; MAPE = 6.5% on daytime, 8.1% at night with TE; direction classification accuracy = 97.2%.
- **Classification**: macro precision/recall/F1 reported in the Statistical Analysis table; class-wise confusion most notable between auto-rickshaw vs. small cars at oblique angles, mitigated by TE + H.
- **Latency**: 27.4 FPS (p95 latency 58 ms) on Jetson Orin NX at 960×544; 31.8 FPS on desktop RTX A2000 at 1280×720 FP16.

Qualitative Observations

- **Occlusion robustness**: The re-ID embeddings help maintain vehicle identity when a bus occludes two-wheelers near stop bars, cutting false double counts notably at rush-hour platoons.
- **Night scenes**: Headlight bloom generates spurious detections in frame-wise methods; TE + calibrated thresholds reduce this.
- **Rain/fog**: Light rain has modest impact; heavy rain reduces small-object AP (two-wheelers) due to rain streaks and wiper occlusions. Temporal voting compensates partially.
- **Edge failures**: Rare CPU spikes during RTSP jitter can momentarily stall inference; buffering and back-pressure settings stabilize throughput without frame drops.

Error Analysis

We inspect miscounts > 4 veh/min on 20 random 1-minute clips:

- 40% due to **long-term occlusion** behind buses/trucks; potential remedy: multi-camera fusion or overhead mounting.
- 30% due to **tight turns** near the line, where centroids cross ambiguously; remedy: curved counting lines aligned with lane geometry.
- 20% due to **small-object loss** (two-wheelers) at night; remedy: super-resolution on ROIs or class-specific priors.
- 10% due to **re-entry** when vehicles U-turn or drift across approaches; remedy: longer track memory and direction filters.

CONCLUSION

This work proposes a deployable, edge-aware pipeline for AI-powered vehicle counting and classification tailored to smart-city environments. By coupling a one-stage detector with an appearance-augmented tracker and perspective-aware line-crossing, the system achieves accurate, de-duplicated counts and reliable class labels across varied scenes, lighting, and weather. Temporal ensembling and light-touch homography yield tangible gains at night and in occlusion-prone settings, while INT8 quantization sustains real-time throughput on modest hardware.

Simulation results across five intersection archetypes indicate strong performance: detection mAP@0.5 of 0.81, tracking IDF1 of 0.78, and counting MAE of 2.3 vehicles/minute—sufficient for adaptive signal timing, demand estimation, and safety diagnostics. Error analysis highlights residual challenges: persistent occlusions behind large vehicles, ambiguous centroid crossings at tight turns, and small-object degradation in heavy rain or low light. These point to **next steps**: (i) multi-camera association to bridge occlusions and avoid double counts across views; (ii) dynamic, lane-aligned counting curves derived from lane-detection networks; (iii) class-specific enhancement (e.g., super-resolution or small-object expert heads) for two-wheelers and auto-rickshaws; (iv) semi-supervised domain adaptation to continuously refine models using confident pseudo-labels; and (v) privacy-preserving analytics with on-prem retention and event-only export.

Overall, the proposed approach balances accuracy, interpretability, and operational practicality. It leverages existing CCTV assets, requires minimal calibration, and fits within the compute envelope of affordable edge devices. As cities push toward responsive traffic management and Vision Zero initiatives, such AI-powered counting and classification systems can provide the granular, timely data needed to manage congestion, prioritize transit, and improve road safety without costly new infrastructure.

REFERENCES

- *Agarwal, A., & Singhal, S. (2022). Real-time vehicle detection and counting in traffic surveillance using YOLOv5 and DeepSORT. International Journal of Computer Vision and Image Processing, 12(3), 45–60. https://doi.org/10.4018/IJCVIP.2022070104*
- *Anagnostopoulos, C. N., Anagnostopoulos, I. E., Psoroulas, I. D., Loumos, V., & Kayafas, E. (2008). License plate recognition from still images and video sequences: A survey. IEEE Transactions on Intelligent Transportation Systems, 9(3), 377–391. https://doi.org/10.1109/TITS.2008.922938*
- *Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934. https://arxiv.org/abs/2004.10934*
- *Chen, L., & Sun, Y. (2021). Lightweight convolutional neural networks for edge-based traffic monitoring. IEEE Internet of Things Journal, 8(6), 4764–4775. https://doi.org/10.1109/JIOT.2020.3028450*
- *Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (pp. 886–893). IEEE. https://doi.org/10.1109/CVPR.2005.177*
- *Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., & Schindler, K. (2021). MOTChallenge: A benchmark for multi-object tracking and segmentation. International Journal of Computer Vision, 129, 845–881. https://doi.org/10.1007/s11263-020-01393-0*
- *Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3354–3361). IEEE. https://doi.org/10.1109/CVPR.2012.6248074*
- *Girshick, R. (2015). Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1440–1448). IEEE. https://doi.org/10.1109/ICCV.2015.169*
- *Hsieh, J. W., Chen, L. C., & Chen, D. Y. (2006). Symmetrical SURF and its applications to vehicle detection and tracking. Pattern Recognition, 39(8), 1289–1301. https://doi.org/10.1016/j.patcog.2006.02.006*
- *Hu, Z., Zhang, X., & Wang, J. (2020). Vehicle detection in complex scenes using deep learning and data augmentation. IEEE Access, 8, 164616–164626. https://doi.org/10.1109/ACCESS.2020.3022569*
- *Li, Y., Huang, C., & Nevatia, R. (2009). Learning to associate: HybridBoosted multi-target tracker for crowded scene. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2953–2960). IEEE. https://doi.org/10.1109/CVPR.2009.5206629*
- *Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2980–2988). IEEE. https://doi.org/10.1109/ICCV.2017.324*
- *Luo, W., Yang, B., & Urtasun, R. (2018). Fast and furious: Real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3569–3577). IEEE. https://doi.org/10.1109/CVPR.2018.00375*

- *Ma, Z., Shao, M., & Fu, Y. (2019). Pedestrian detection and tracking in crowded scenes with multi-view cameras. IEEE Transactions on Circuits and Systems for Video Technology, 29(4), 1030–1043. https://doi.org/10.1109/TCSVT.2018.2815320*
- *Milan, A., Leal-Taixé, L., Reid, I., Roth, S., & Schindler, K. (2016). MOT16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831. https://arxiv.org/abs/1603.00831*
- *Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767. https://arxiv.org/abs/1804.02767*
- *Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031*
- *Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2021). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696. https://arxiv.org/abs/2207.02696*
- *Wojke, N., Bewley, A., & Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. In Proceedings of the IEEE International Conference on Image Processing (pp. 3645–3649). IEEE. https://doi.org/10.1109/ICIP.2017.8296962*
- *Zhang, S., Benenson, R., & Schiele, B. (2018). CityPersons: A diverse dataset for pedestrian detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3213–3221). IEEE. https://doi.org/10.1109/CVPR.2017.340*