Automated visual traffic surveillance system for traffic analysis

Convolve – EU Horizon 2020

Dick Scholte

R&D Engineer ViNotion BV / Doctoral Candidate TU/e

ViNotion

Overview

- Introduction
- Typical system description
- Dynamic neural network
- Improving object localization

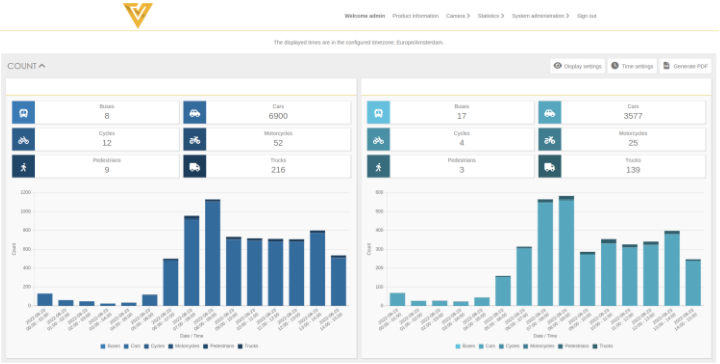
Introduction – Traffic surveillance

- Range of tasks -> road/traffic safety
  - Crowd management analysis
  - Congestion measurements
  - (Near-) accident observations
- Traffic surveillance cameras
  - Positioned couple of meters above ground
  - Neural networks extract relevant information
- Analyze behavior of participants
  - Detect, localize and follow
  - Real time -> computationally efficient + optimized embedded hardware

ViNotion

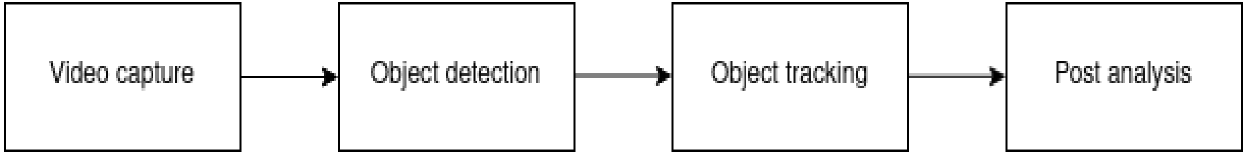
Typical system description

Video capture → Object detection → Object tracking → Post analysis



ViNotion

Typical system algorithms



- Data throughput, low latency, data transfer optimization, parallel processes
- Optimization
  - Software scheduling
  - profiling tools

Benchmarking GPU/CPU

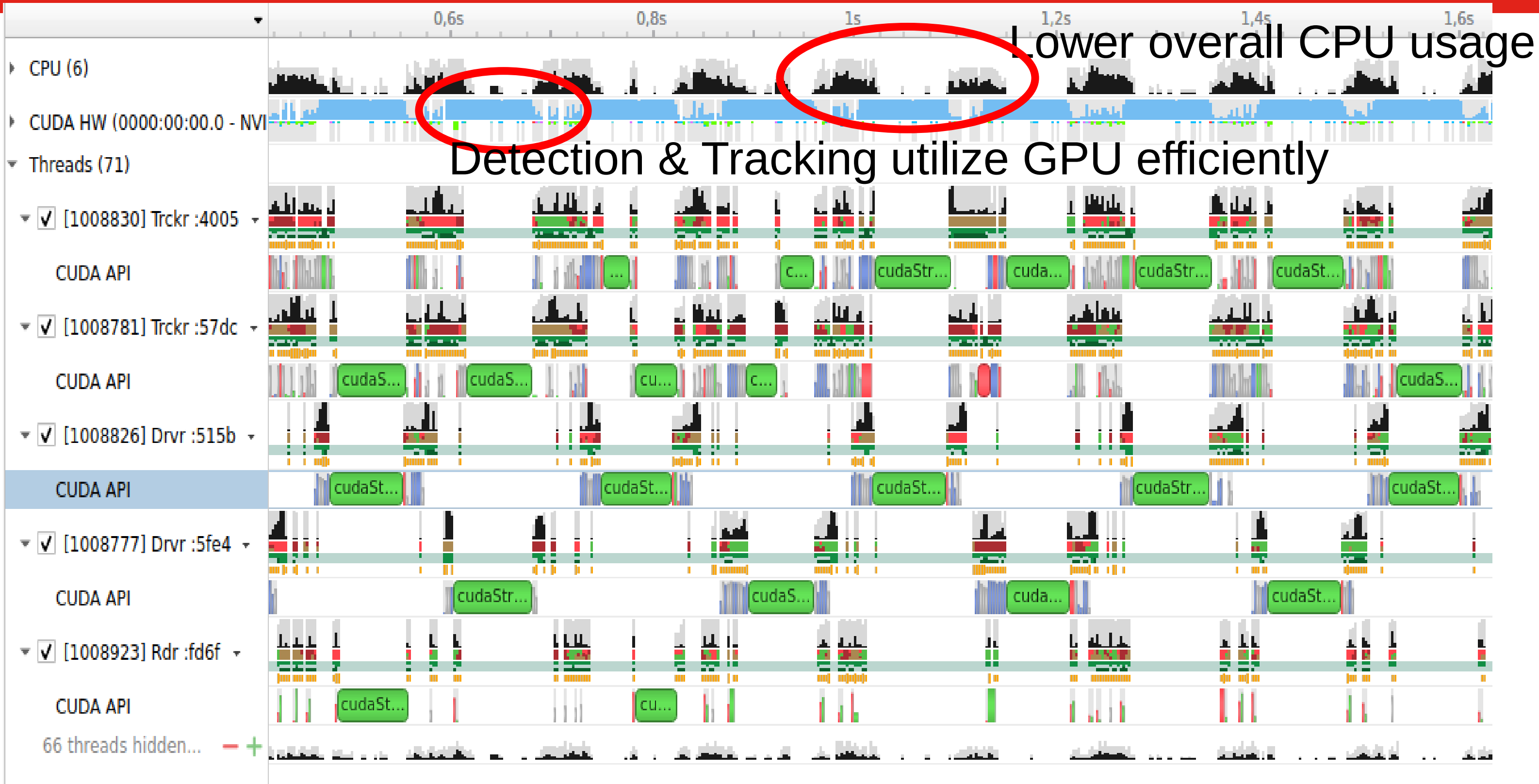
ViNotion

Benchmarking GPU/CPU

- GPU utilization (CUDA) is low
  - Tracking waits for detection (dependency)
  - CPU frequently waiting for GPU functions
  - CPU/GPU synchronization (data copies)
- Solution
  - Multi cuda stream processing
  - Minimize data synchronization
  - Decouple tracking and detection (@ 4 fps)

ViNotion

Benchmarking GPU/CPU



ViNotion

Power consumption

- After enhancements and optimizations:
  - Process 6 streams in parallel

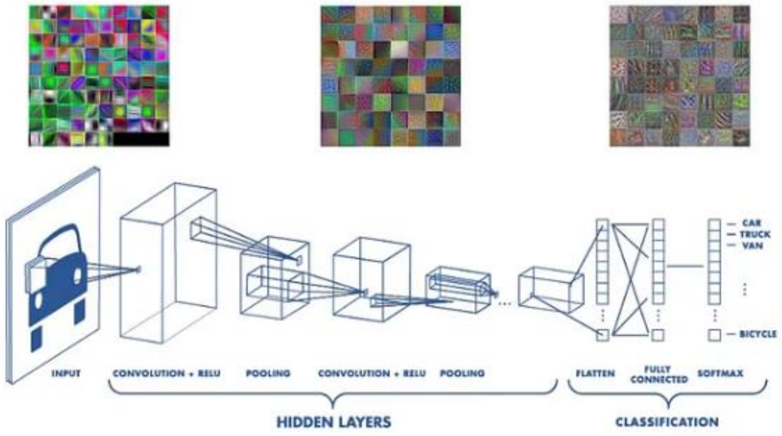
- **Jetson Xavier NX (21 Tops INT8, 15-20 Watt)**
- **Jetson Orin Nano (40 Tops, INT8, 7-15 Watt)**

- High power consumption
  - not suited for battery-based systems

ViNotion

Overview

- Introduction
- Typical system description
- **Dynamic neural network**
- Improving object localization

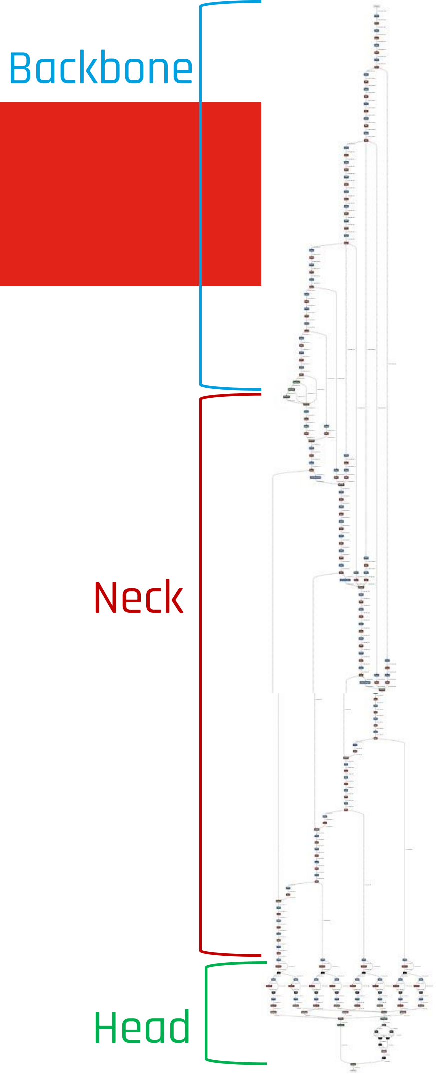


ViNotion

Dynamic neural network

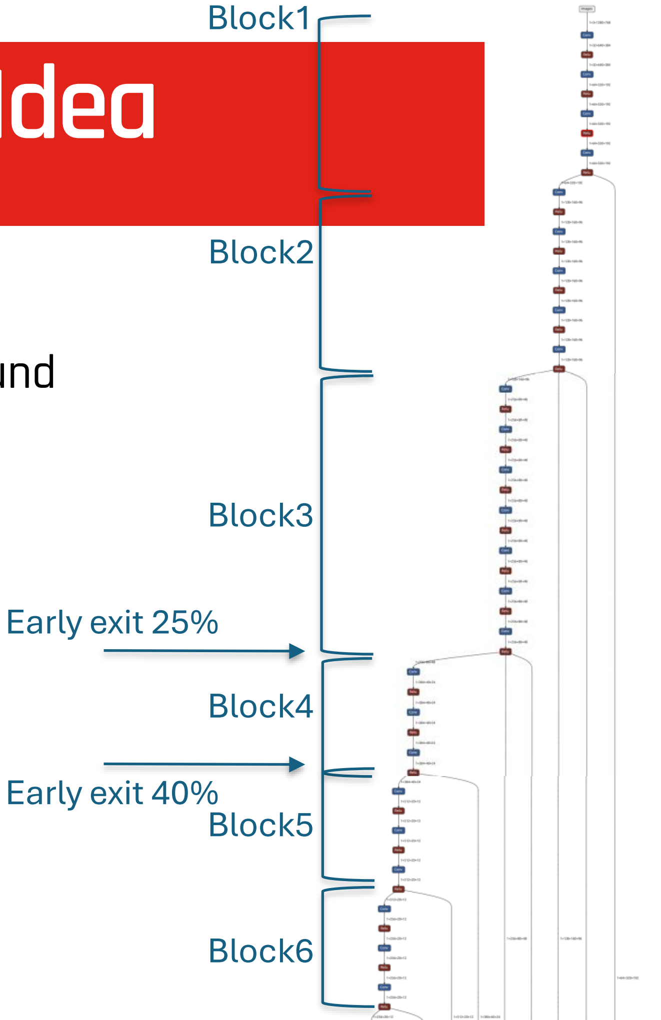
How to utilize YOLOv6 object detector?

- System always processes
  - Detector always run
  - Waste of energy
- Early exit branch
  - Stops processing early
  - Lower the overall power consumption
- Backbone -> 50% of inference



YOLOv6 Object detection - Idea

- The metric used is a background score
  - Predict if the image contains objects vs background
- Any object present -> easier task
  - No localization and classification
  - Small network capacity
- Added early exits in the YOLOv6 network
  - Additional costs (~5%) for early exit metric
  - Large energy savings if no objects present
- Reduced processing time by 60 to 75%



YOLOv6 Object detection - Demo

Blurred for privacy reasons

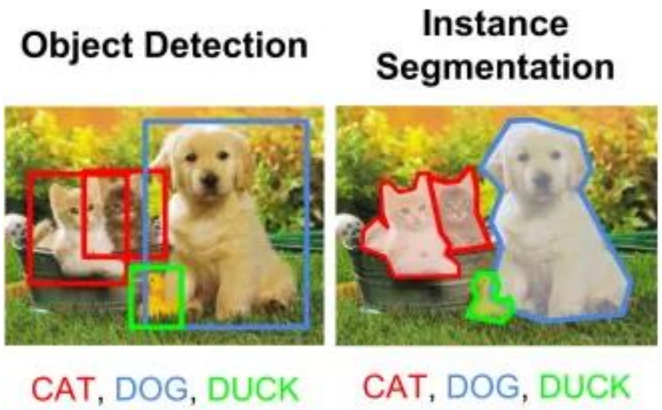
- Background score:
  - 1 (high) -> no objects
  - 0 (low) -> objects present

- **Conclusions**
  - Promising results
  - Efficient for low-traffic scenes

ViNotion

Overview

- Introduction
- Typical system description
- Dynamic neural network
- **Improving object localization**



ViNotion

Object locations

- 2D bounding boxes -> object location
- Participants have complex shape
- Contours -> more accurate representation
- Location large elongated truck
  - Bounding box
  - Instance segmentation

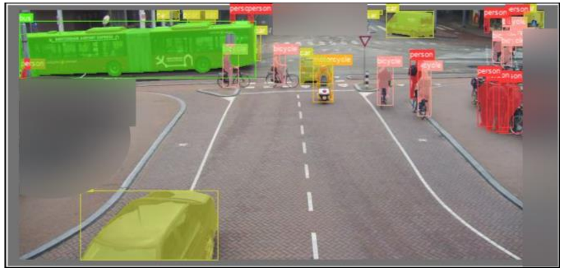
- **Instance masks -> better locations**

Blurred for privacy reasons

ViNotion

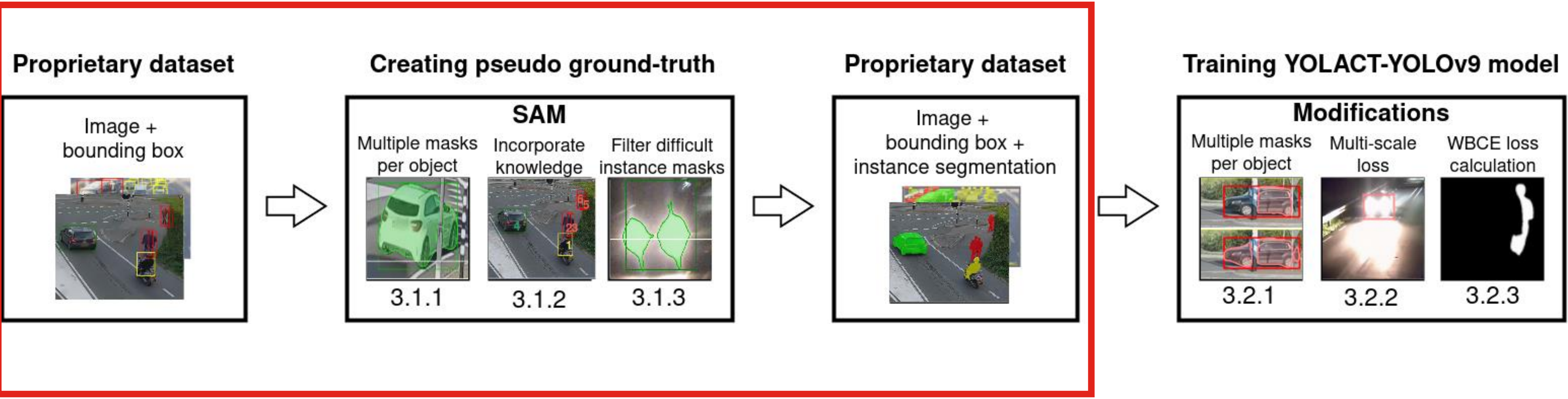
Problems

- Instance segmentation models
  - Most models cannot be utilized for real-time applications (25 fps)
  - Suitable for embedded GPU devices
  - YOLACT-YOLOv9 model is real-time
- Instance segmentation datasets
  - Publicly available datasets not suited for traffic surveillance
  - Proprietary dataset **only contains bounding boxes**
    - Manual annotation unfeasible -> 100 images manually annotated



ViNotion

System overview

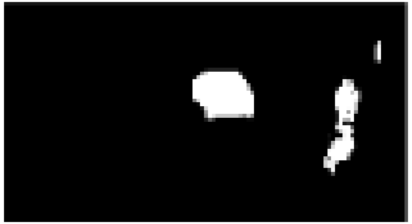


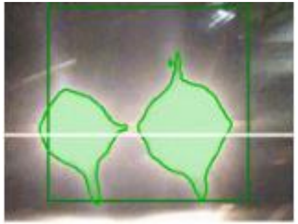
ViNotion

Generation Procedure

Utilizing Segment Anything Model

- Creating pseudo ground-truth data for instance segmentation
  - Prediction of multiple masks per object
  - Incorporating knowledge about foreground objects
  - Filtering of difficult instance masks

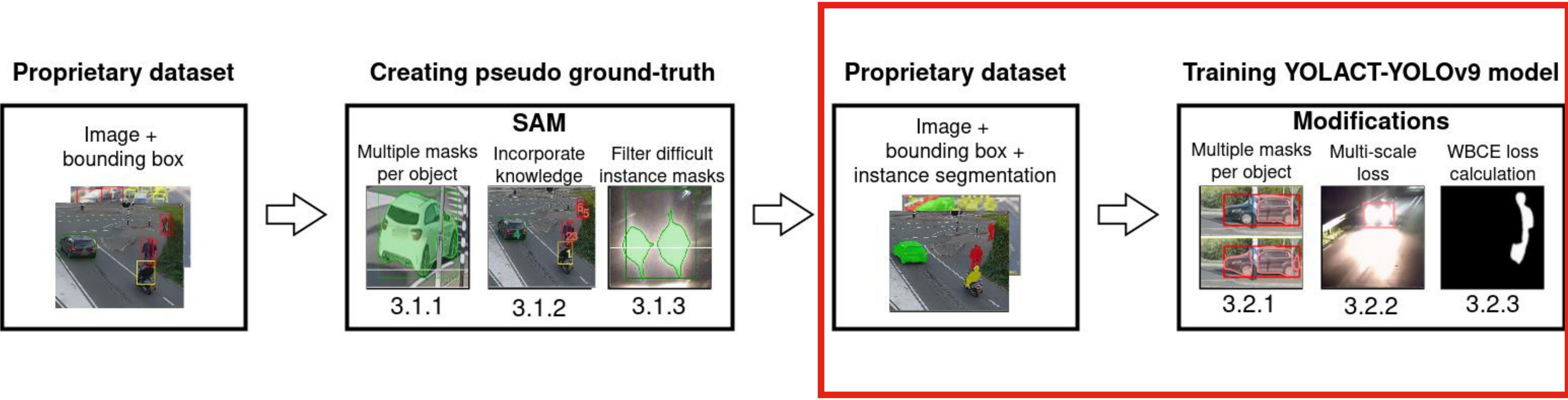




  - Segmentation performance mAP 0.5-0.95 increase from 77% -> 81.7%

ViNotion

System overview



ViNotion

Training YOLACT-YOLOv9 model

- Normally fully supervised trained
- Modifications:
  - Allowing multiple instance masks per object
    - Important for tracker
  - Implemented multi-scale loss
    - Uses missing instance masks
    - Based on scale consistency
  - Change of mask loss calculation
    - Highly occluded objects
    - Large background, small foreground -> negligible loss

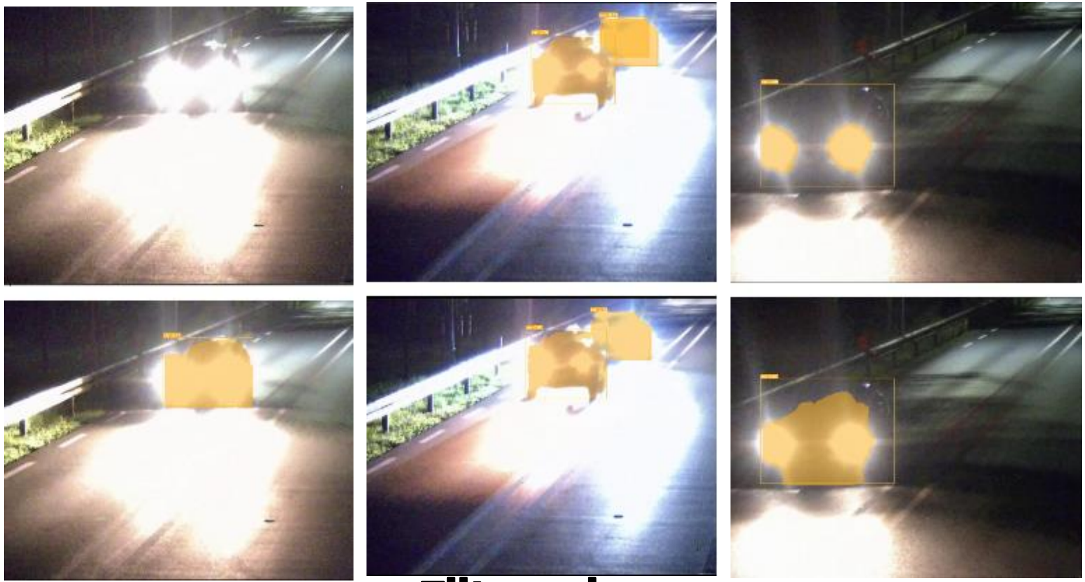
ViNotion

Filtering the over-saturated images

YOLOv9 trained with (un)filtered dataset

- Reduced false negatives
- Reduced false positives
- Prevent overfitting on headlights

Unfiltered



Filtered

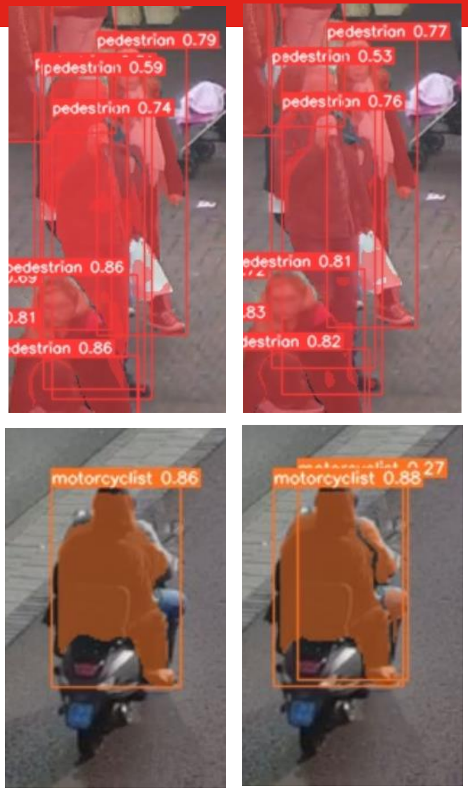
ViNotion

Results different loss calculation

BCE-WBCE

- Reduce overlapping instance masks
- Reduce false negatives

- Detection performance 83.5%
- Segmentation performance 61.5%
  - Pedestrians increase with 0.8%
  - Cyclists increase with 2.9%



ViNotion

Questions?

Blurred for privacy reasons

ViNotion