Title

Setup Of A Deep Learning-Based System For Detecting Queues On Highways Using Z2 Board, Evaluation Of Performance And Power Consumption

Relatore:
Prof. Alessandro Mecocci
Prof. Sandro Bartolini

Candidato:
Vahid Farjood Chafi

Outline

With a complete harvesting device (Z2 board), we aim to determine the level of traffic on a particular highway:

Steps:

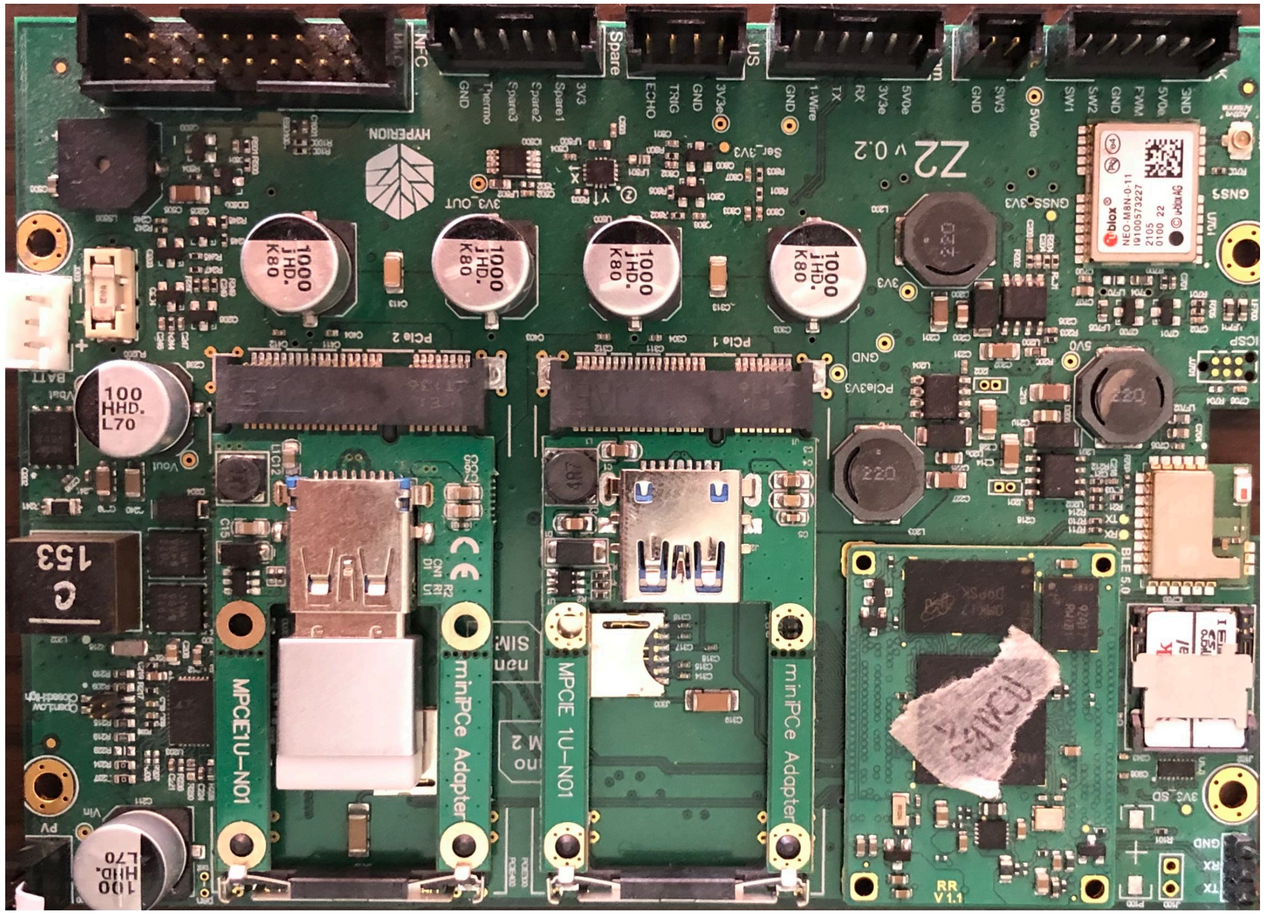
1. Capturing two sequential images.
2. Detecting all the existing vehicles.
3. Tracking those detected vehicles.
4. Collect some quantitative features.
5. Estimate the level of traffic (Low, Medium, High).
6. Transmit the result.

Constraints:

- Hardware Resources.
- Power Consumption.

Device Specification

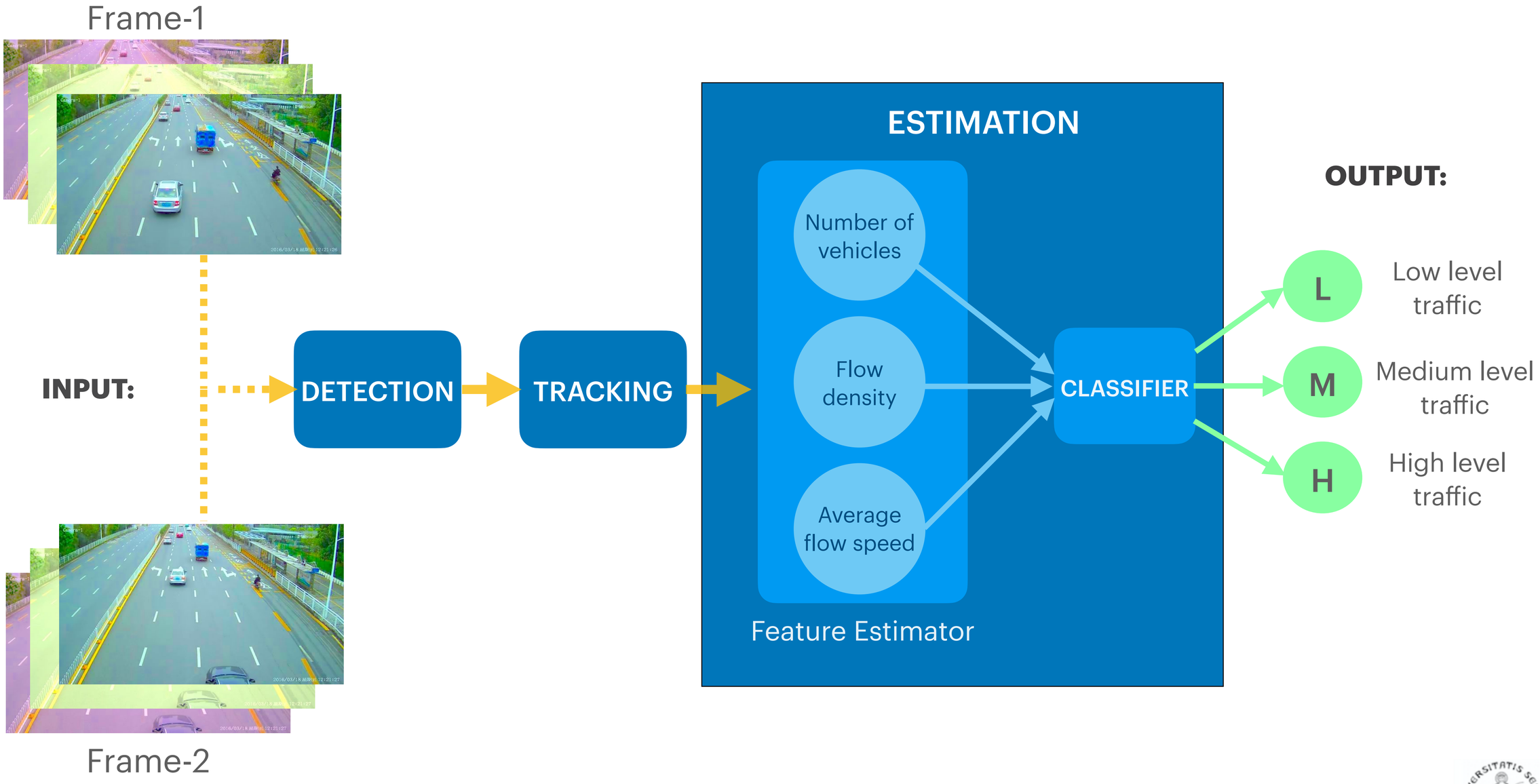
- **Hardware:**
  1. Z2 board
  2. Memory 256MB
  3. ARM processor
  4. USB Camera
  5. WiFi dongle
  6. Battery 16V
- **Software:**
  - Operating system:
    - Linux Debian 10
  - Programming Languages:
    - C++
    - Python
  - Build System:
    - CMake
    - Make
  - Packages:
    - OpenCV



Vahid Farjood Chafi
M.S. Computer and Automation Engineering

University of Siena

Architecture



Vahid Farjood Chafi
M.S. Computer and Automation Engineering

University of Siena

Detection

- We need light and fast pre-trained object detection models:

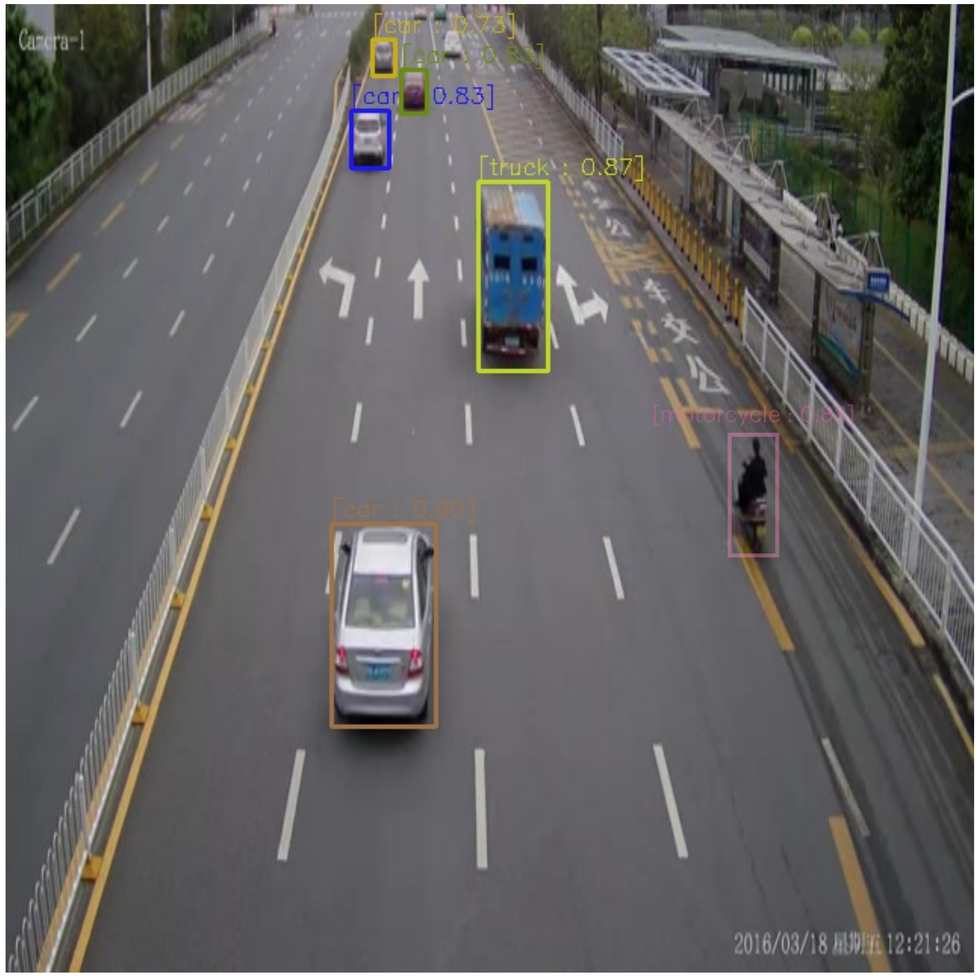
1. YOLOv5n.
2. MobileNet_SSDv2.

- We need to fine-tune those models based on a brand new dataset:

- Dataset:
  - 2K annotated traffic images.
  - Images were extracted from traffic videos (freely accessible online).
  - 4 classes (car, truck, bus, motorcycle).

- Inferencing:

- OpenCV DNN module were used for inferencing.



Vahid Farjood Chafi
M.S. Computer and Automation Engineering

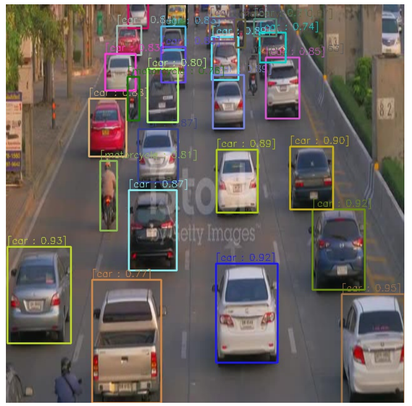
University of Siena

Detection

Frame-1

Object Detection Model

=

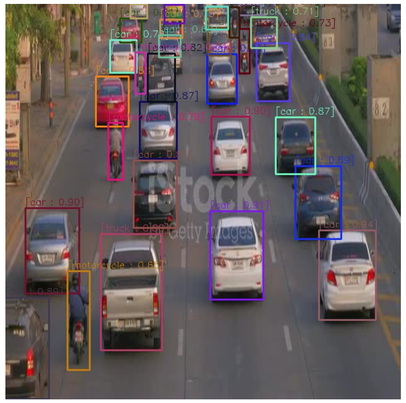


Buf-1 = all the detected vehicles

Frame-2

Object Detection Model

=



Buf-2 = all the detected vehicles

Total-Buff= [Buf-1, Buf-2]

Vahid Farjood Chafi
M.S. Computer and Automation Engineering

University of Siena

Tracking

Tracking is accomplished based on two assumptions:

1. No vehicle could cross from one lane to another during 1 second.
2. The area of the bounding box should not increase in the following frames due to the selected camera view.

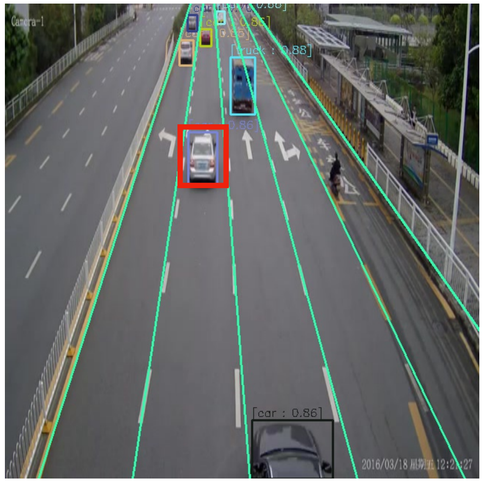
- Tracking procedure:

- Determine road's lanes on the image space.
- Find and assign which lane number vehicles belong to.
- For each vehicle on the same lane, check if:
  - The area of the boxes in the previous frame is larger or equal.
  - Find the least distance among them.

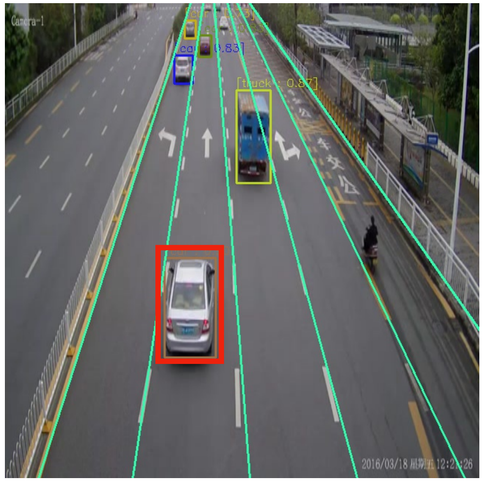
- Deep Learning approach?

- Constraints:
  1. Hardware Resources.
  2. Power Consumption.

Frame 2



Frame 1



Vahid Farjood Chafi
M.S. Computer and Automation Engineering

University of Siena

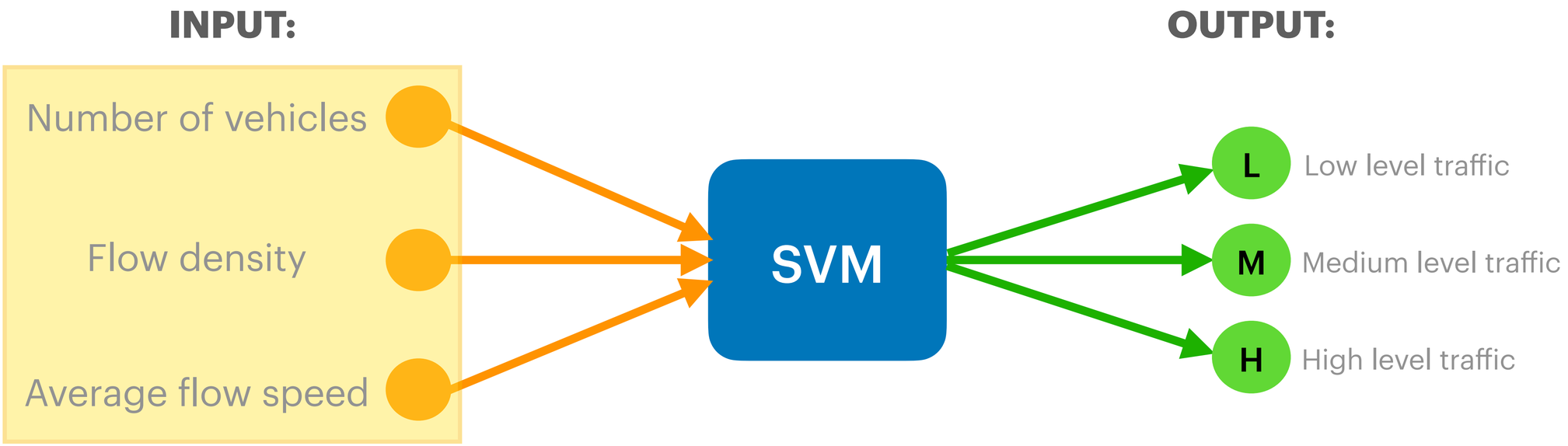
Estimating Features

- **Number of vehicle:**
  - Number of tracked vehicles in two consecutive images.

- **Flow density:**
  - Total area of the road.
  - Total area of the all vehicles.

- **Average flow speed:**
  - Estimate the speed of each vehicle.
    - Based on the distance travelled between two frames.
  - Sum them up and get the flow average speed.

Estimation



Vahid Farjood Chafi
M.S. Computer and Automation Engineering

University of Siena

Estimation

- **Why Linear SVM:**
  - We need a light and fast classifier.
  - More sophisticated model might be used in the future.
- **Training:**
  - Based on a brand new dataset:
    - 100 samples.
    - Samples with the same features (number of vehicles, flow density, flow speed).
    - By observing real evidence on the traffic videos.
- **Prediction:**
  - Capable of satisfying most of the practical applications, e.g. for highroad analysis.
  - Based on 3 levels (Low, Medium, High).
  - Future work could also include more precise classes.

Performance

1. Accuracy
2. Computation Time
3. Power Consumption

Performance

1. **Accuracy**:
   - Dataset:
     - 135 input samples
     - Extracted from traffic videos (19 highways)
     - Based on observing real evidence
   - Evaluation:
     - Detection and Tracking:
       - Based on discrepancy between the prediction and ground truth
     - Estimation:
       - Using Confusion Matrix

Total Accuracy

Pipeline	YOLO	MobileNet_SSD
Detection	0,89	0,65
Tracking	0,92	0,73
Estimation	0,93	0,47
Total	**0,91**	**0,62**

	Prediction		
[135]	Low	Medium	High
Low	45	4	0
Medium	2	38	1
High	0	2	43
Recall	**0,92**	**0,93**	**0,96**
Precision	**0,96**	**0,86**	**0,98**
Total Acc	**0,93**		

YOLOv5

	Prediction		
[135]	Low	Medium	High
Low	44	5	0
Medium	21	19	1
High	12	33	0
Recall	**0,90**	**0,46**	**0,00**
Precision	**0,57**	**0,33**	**0,00**
Total Acc	**0,47**		

MobileNet SSD

Vahid Farjood Chafi
M.S. Computer and Automation Engineering

University of Siena

Example

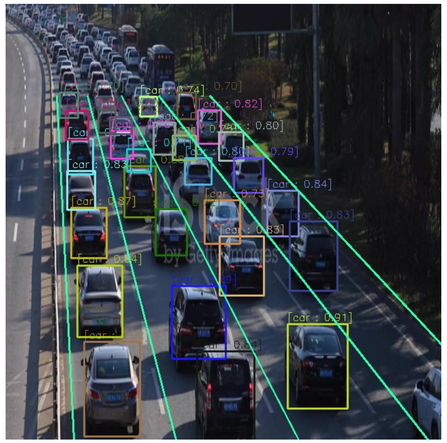
Input:

- Two images (one sec sample rate)

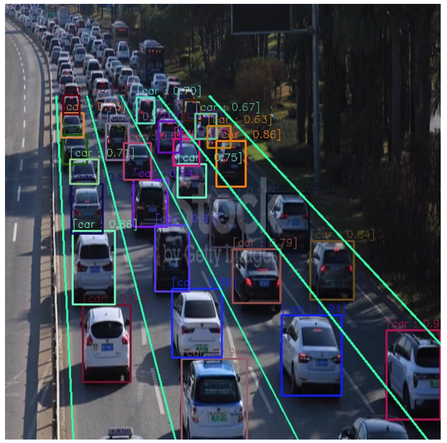
Label:

- Estimation = High
- Detection = 60
- Tracking = 37

Frame 1



Frame 2

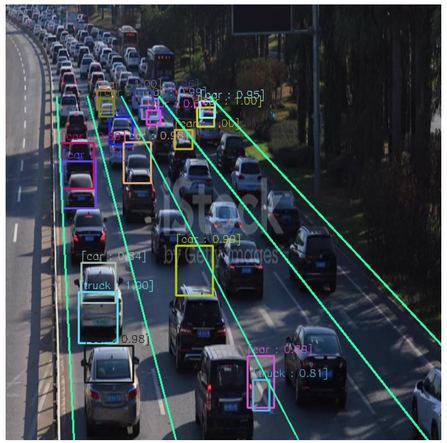


Result:

Output for **YOLO**:

- Traffic Status: High
- Number of detected: 55
- Number of tracked: 31
- Flow density: 57%
- Flow average speed: 21
- Highest flow lane number: 2

Frame 1



Frame 2

Result:

Output for **MobileNet_SSD**:

- Traffic Status: Medium
- Number of detected: 35
- Number of cars: 22
- Flow density: 20%
- Flow average speed: 42
- Highest flow lane number: 2

Vahid Farjood Chafi
M.S. Computer and Automation Engineering

University of Siena

Performance

2. Computation Time:

- Based on the two different detection models.
- For each task separately
- Used STL std::*chrono*

Pipeline	YOLO	MobileNet_SSD
Capturing two images	4s 102ms	4s 102ms
Model loading	1s 31ms	7s 541ms
Loading two images	0s 430ms	0s 330ms
Detection time	50s 938ms	61s 983ms
Tracking time	0s 37ms	0s 34ms
Estimation time	0s 11ms	0s 16ms
Saving the result	0s 667ms	0s 662ms
Total time	**57s 216ms**	**75s 668ms**

Performance

Power Consumption:

- Defined "cycles" of operations:
  - Idle consumption
  - Sleep consumption
  - Frame acquisition
  - Inferencing
  - Transmission
- Based on:
  - The computation time of different operations
  - For the two different detection models
- Estimate for three different sleep cycles:
  - 5 minutes sleep
  - 10 minutes sleep
  - 15 minutes sleep

	Cycles	Interval	YOLO		MobileNet SSD	
			Time (s)	Energy (J)	Time (s)	Energy (J)
Energy Consumed	**5 min cycle**	Sleep Mode (5min)	300	5.760	300	5.760
		For One Cycle	359.45	68.027	377.55	84.875
		For One Hour	3600	681.313	3600	809.299
		For One Day (24h)	86400	**16351.509**	86400	**19423.169**
	10 min cycle	Sleep Mode (10min)	600	11.520	600	11.520
		For One Cycle	659.45	73.787	677.55	90.635
		For One Hour	3600	402.811	3600	481.568
		For One Day (24h)	86400	**9667.471**	86400	**11557.643**
	15 min cycle	Sleep Mode (15min)	900	17.280	900	17.280
		For One Cycle	959.45	79.547	977.55	96.395
		For One Hour	3600	298.473	3600	354.992
		For One Day (24h)	86400	**7163.352**	86400	**8519.815**

Table-3.11 Energy consumption per different cycles

Experiment

Location:

- Urban area of the city of RIMINI
- Similar to the actual harvesting scenario

Condition:

- 4 input samples (During 20 minutes)
- Every five minutes, one input sample
- Two inputs for Yolo model, and two for MobileNet SSD

Vahid Farjood Chafi
M.S. Computer and Automation Engineering

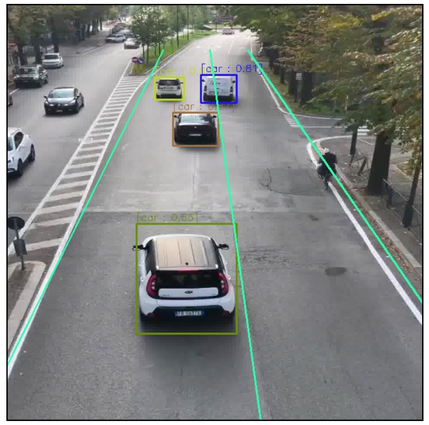
University of Siena

Experiment

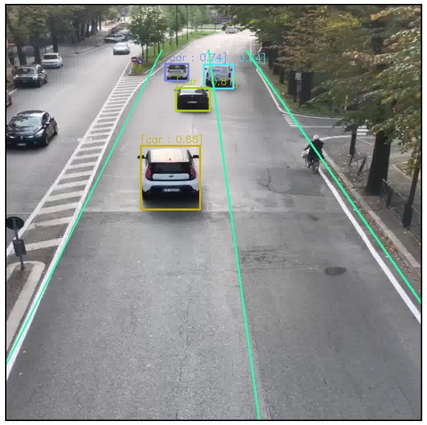
Result (for two inputs):

- YOLO:
  - Detection : 100% correct
  - Tracking: 100% correct
  - Estimation: Low Traffic (correct)
- MobileNet SSD:
  - Detection : 50% correct
  - Tracking: 50% correct
  - Estimation: Low Traffic (correct)

Frame 1



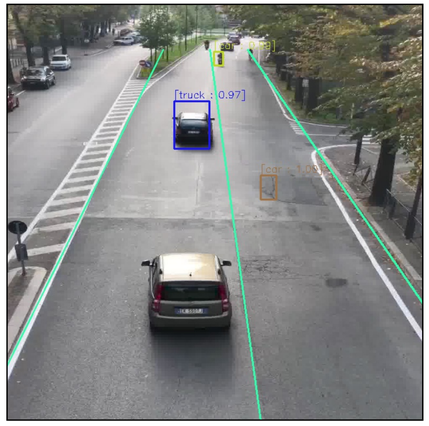
Frame 2



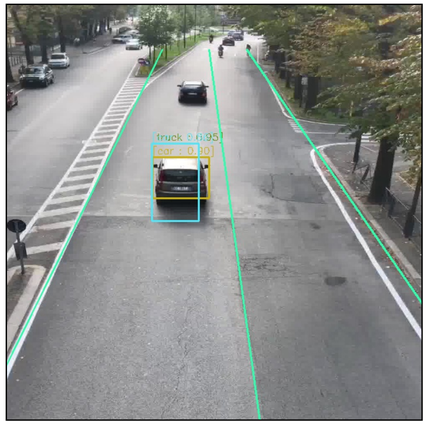
Result:

Output for **YOLO**:
• Traffic Status: Low
• Number of cars: 4
• Flow density: 13%
• Flow average speed: 31
• Highest flow lane number: 1

Frame 1



Frame 2



Result:

Output for **MobileNet_SSD**:
• Traffic Status: Low
• Number of cars: 5
• Flow density: 8%
• Flow average speed: 5
• Highest flow lane number: 1

Vahid Farjood Chafi
M.S. Computer and Automation Engineering

University of Siena

Conclusion

- **Due to hardware and power supply constraints:**
  - Used sub-sampling of the traffic flow
  - Used Deep Learning models only for object detection task
- **The accuracy of the detection models has a significant impact on the final result.**
- **The computation time of the detection, has an impact on the overall energy consumption.**
- **The long-term energy consumption could be reduced by transferring messages under specific circumstances.**