Hardware Accelerated Vehicle Detection Using Computer Vision for a Dynamic Traffic System

Ubiquitous Computing Laboratory
Patrick Celon | Timothy Chua | Paul Ilaga
Advisers : Nestor Tiglao | Jethro Limjoco

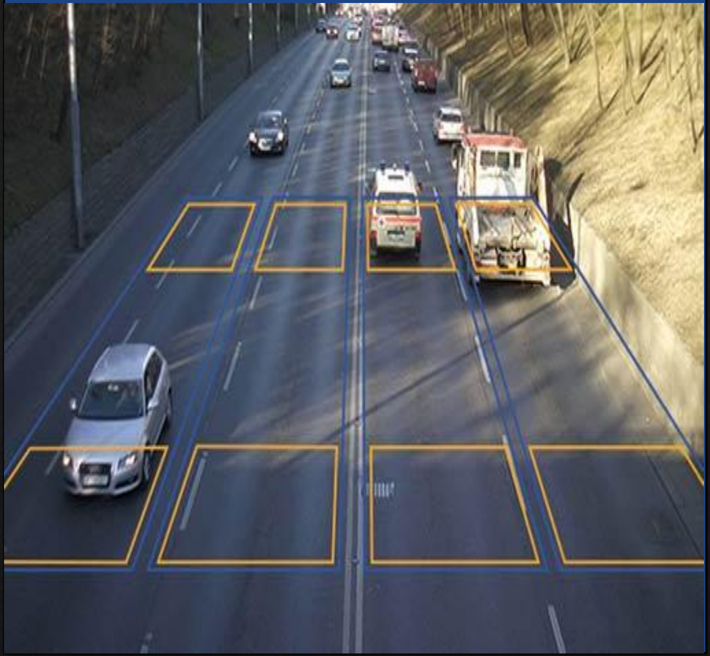
UCL04

A typical day in EDSA

Yee, J. (2016, October). *MMDA now says Edsa Traffic faster by 21 mins, not just 3.* Retrieved from http://newsinfo.inquirer.net/825479/mmda-now-says-edsa-traffic-faster-by-21-mins-not-just-3

Current Traffic Monitoring or Regulating Techniques

Induction Loops



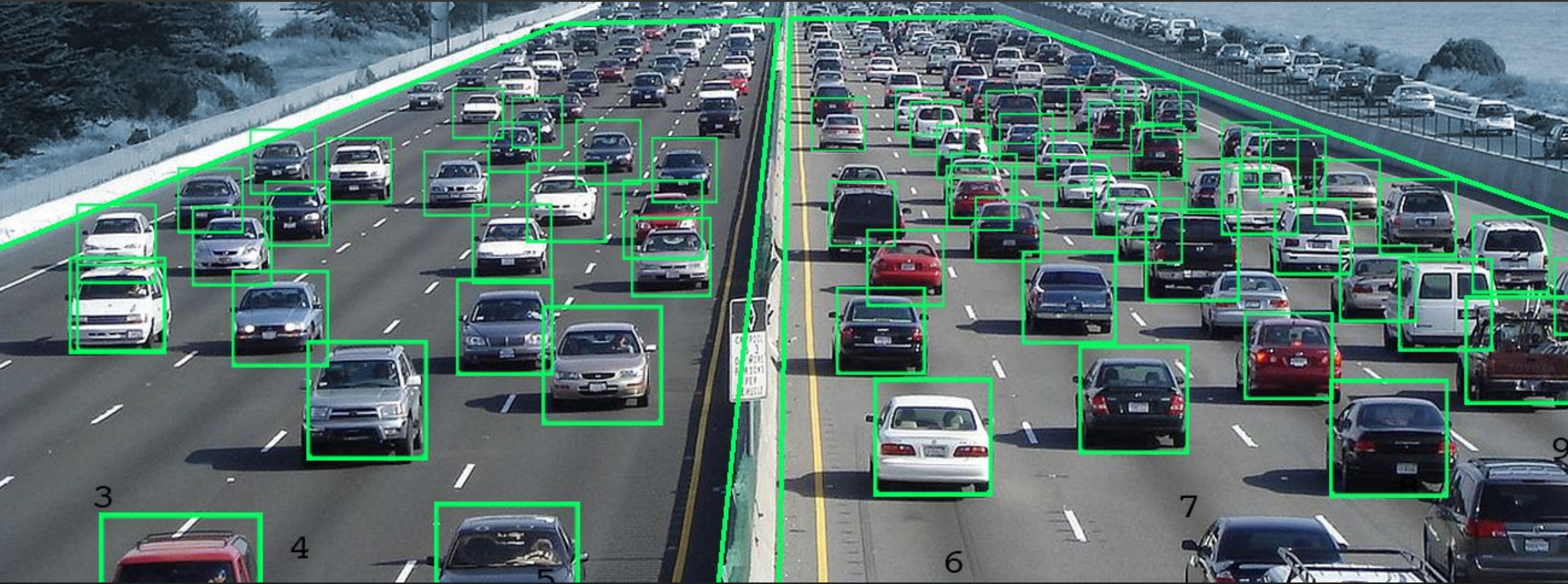
RFID [2]

Computer Vision[3]

[1] R. F. A. M. Nor, F. H. K. Zaman and S. Mubdi, "Smart traffic light for congestion monitoring using LoRaWAN," *2017 IEEE 8th Control and System Graduate Research Colloquium (ICSGRC)*, SHAH ALAM, Malaysia, 2017, pp. 132-137.

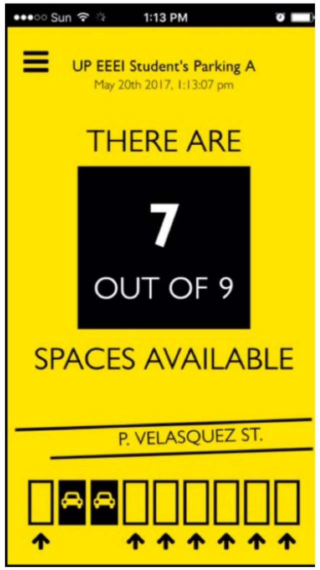
[2] A. Saikar, M. Parulekar, A. Badve, S. Thakkar and A. Deshmukh, "TrafficIntel: Smart traffic management for smart cities," *2017 International Conference on Emerging Trends & Innovation in ICT (ICEI)*, Pune, 2017, pp. 46-50.

[3] T. Osman, S. S. Psyche, J. M. S. Ferdous and H. U. Zaman, "Intelligent traffic management system for cross section of roads using computer vision," *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, 2017, pp. 1-7.



Computer Vision for Vehicle Counting

High Performance Vision-Based Systems



●●●○○ Sun 1:13 PM

UP EEEI Student's Parking A
May 20th 2017, 1:13:07 pm

THERE ARE

7

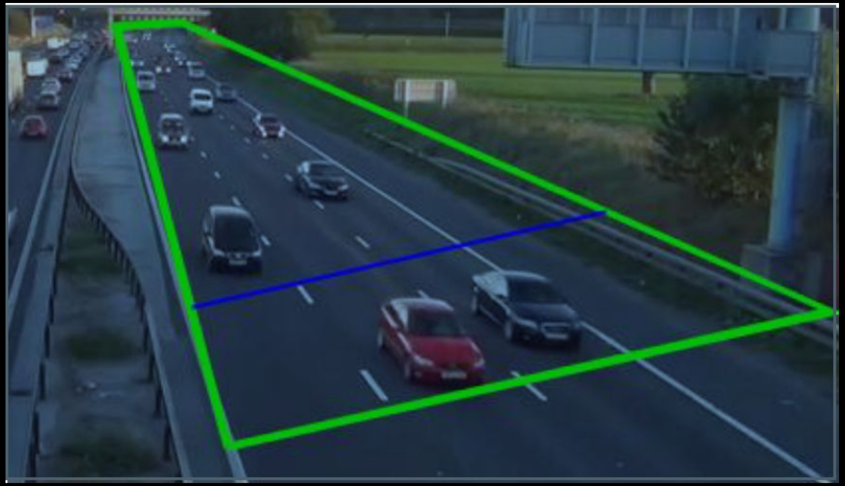
OUT OF 9

SPACES AVAILABLE

P. VELASQUEZ ST.

Parakeet

C. Dizon, L. Magpayo, and A. Uy, “An implementation of an open-space visual smart parking system, Bachelor’s thesis, University of the Philippines, Diliman, Diliman, 2017.

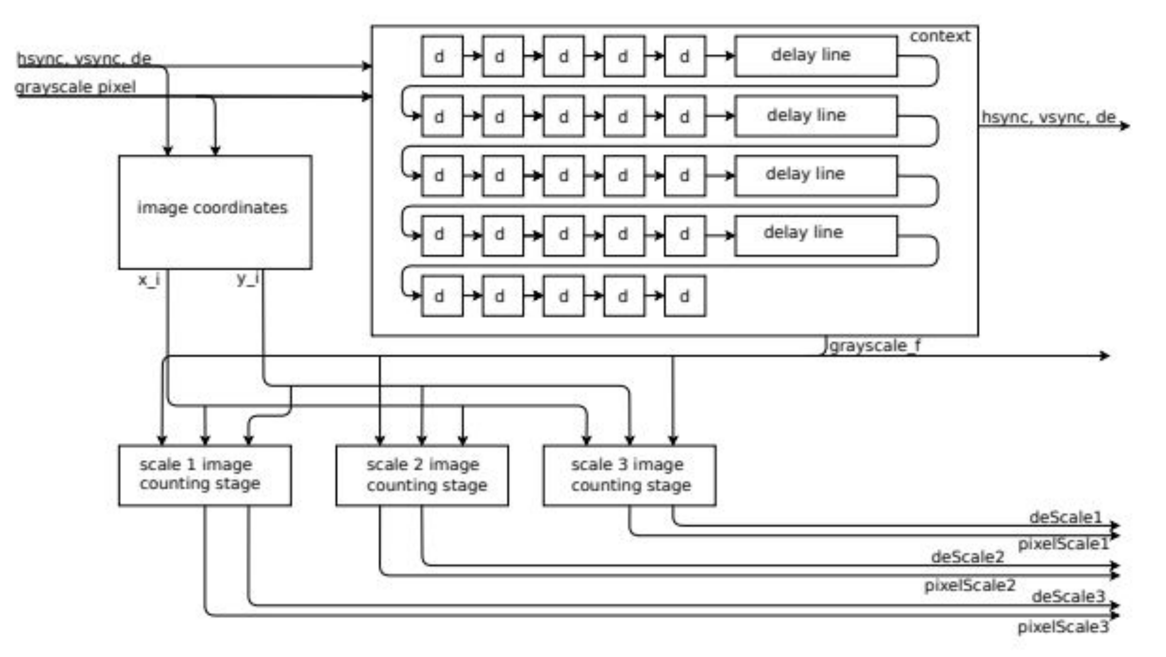


Velazquez-Pupo, Roxana & Sierra-Romero, Alberto & Torres-Roman, Deni & V Shkvarko, Yuriy & Santiago-Paz, Jayro & Gomez-Gutierrez, David & Robles-Valdez, Daniel & Hermosillo-Reynoso, Fernando & Romero-Delgado, Misael. (2018). Vehicle Detection with Occlusion Handling, Tracking, and OC-SVM Classification: A High Performance Vision-Based System. Sensors. 18. 374. 10.3390/s18020374.

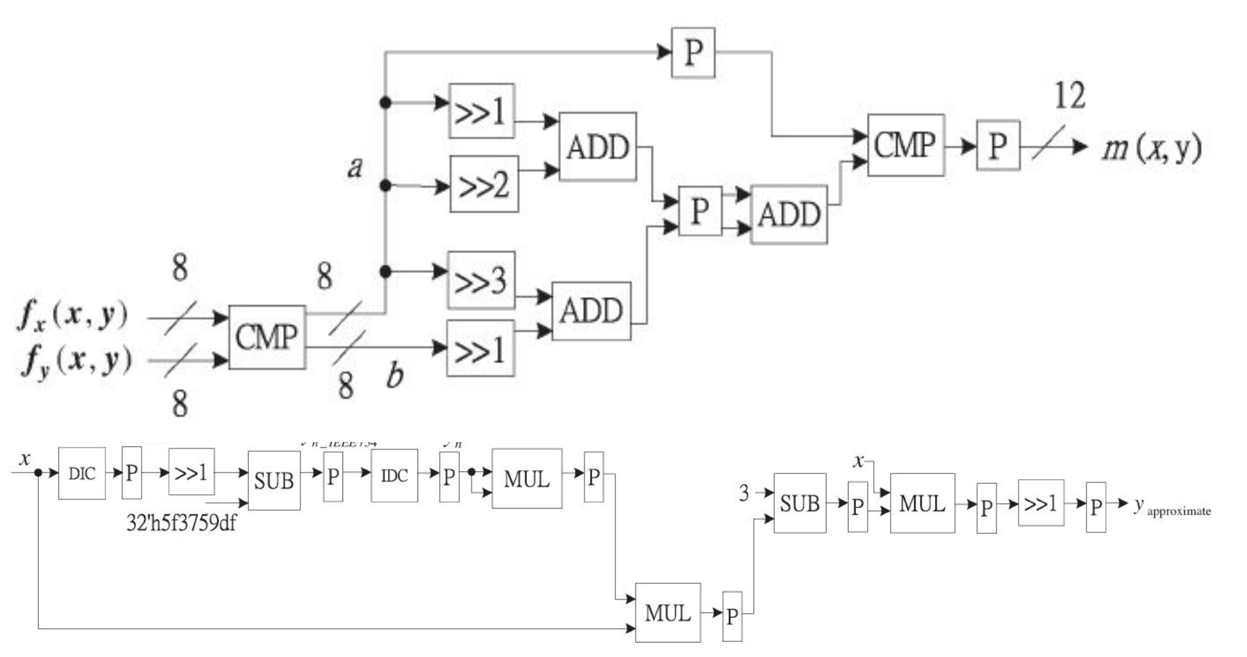
Hardware Acceleration via FPGAs

- Computer Vision algorithms are **very slow**!
  - Lots of convolutional filters
  - Floating point calculations
  - Complicated operations ($\tan^{-1}$, 1/sqrt)
  - Normally implemented in powerful desktop PCs

- FPGAs are an option to **accelerate** CV algorithms
  - Allow parallelization in computation
  - Low power consumption allows embedded solutions



Hardware-accelerated image scaling



Hardware-accelerated square root and inverse square root operations

Many implementations, but little research on how CV algorithms can be implemented with ***dynamic traffic management algorithms***

Some Dynamic Traffic Management Algorithms

	Classic Traffic Light Control	Dynamic Traffic Light Control	Oldest Arrival First	Intelligent Traffic Light Control
Infrastructure	Traffic Control Box	VITCO/RITCO	VANETs	VANETs
Scheduling	Static Round-Robin	Threshold-based	Oldest Arrival First	Density Based

[1] P. S. Chakraborty, P. R. Sinha and A. Tiwari, "Real Time Optimized Traffic Management Algorithm for Intelligent Transportation Systems," 2015 IEEE International Conference on Computational Intelligence & Communication Technology, Ghaziabad, 2015, pp. 744-749.

[2] T. Osman, S. S. Psyche, J. M. S. Ferdous and H. U. Zaman, "Intelligent traffic management system for cross section of roads using computer vision," *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, 2017, pp. 1-7.

Some Dynamic Traffic Management Algorithms

	Classic Traffic Light Control	Dynamic Traffic Light Control	Oldest Arrival First	Intelligent Traffic Light Control
Infrastructure	Traffic Control Box	VITCO/RITCO	VANETs	VANETs
Scheduling	Static Round-Robin	Threshold-based	Oldest Arrival First	Density Based

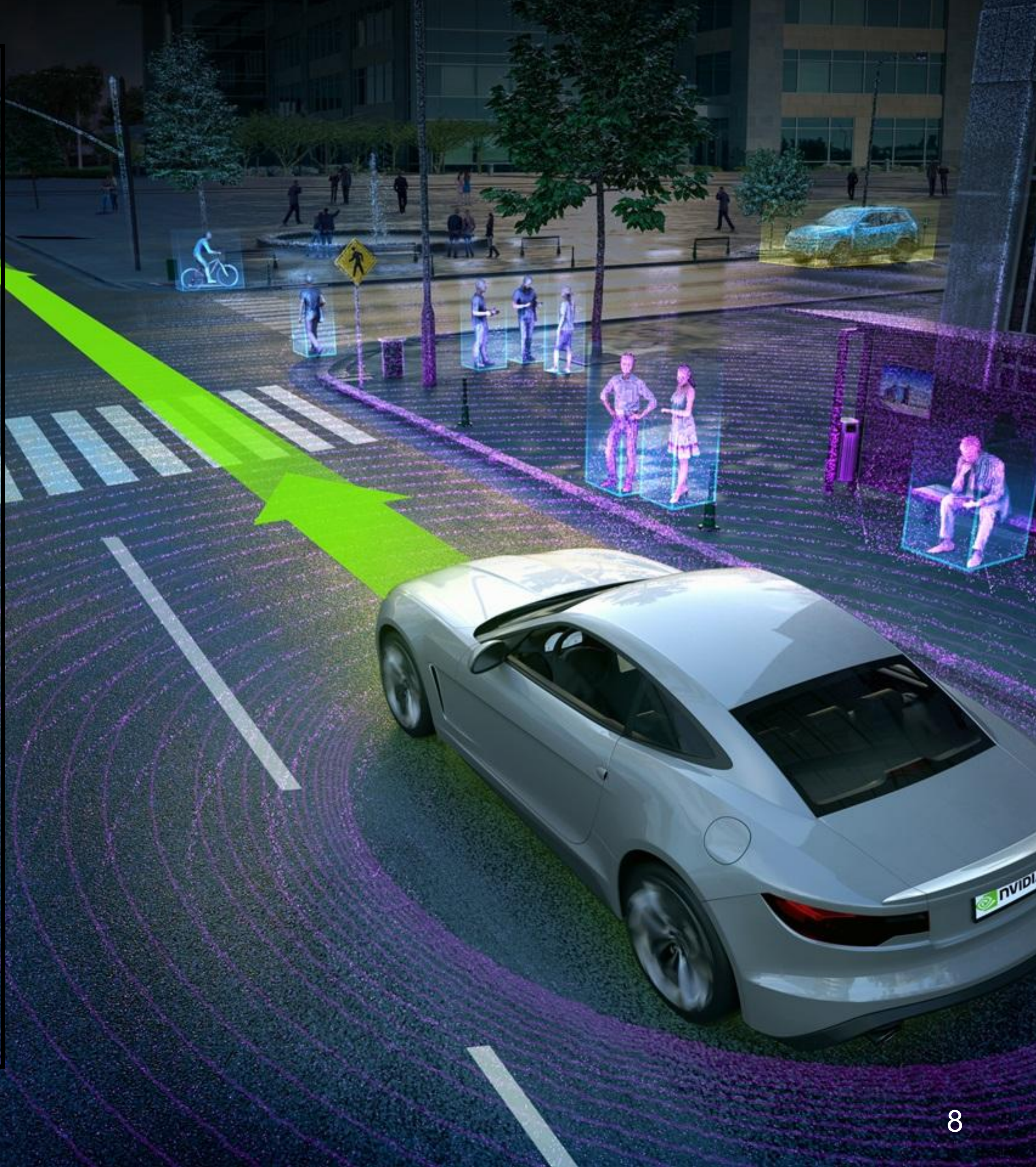
All use non-visual techniques!

[1] P. S. Chakraborty, P. R. Sinha and A. Tiwari, "Real Time Optimized Traffic Management Algorithm for Intelligent Transportation Systems," 2015 IEEE International Conference on Computational Intelligence & Communication Technology, Ghaziabad, 2015, pp. 744-749.

[2] T. Osman, S. S. Psyche, J. M. S. Ferdous and H. U. Zaman, "Intelligent traffic management system for cross section of roads using computer vision," *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, 2017, pp. 1-7.

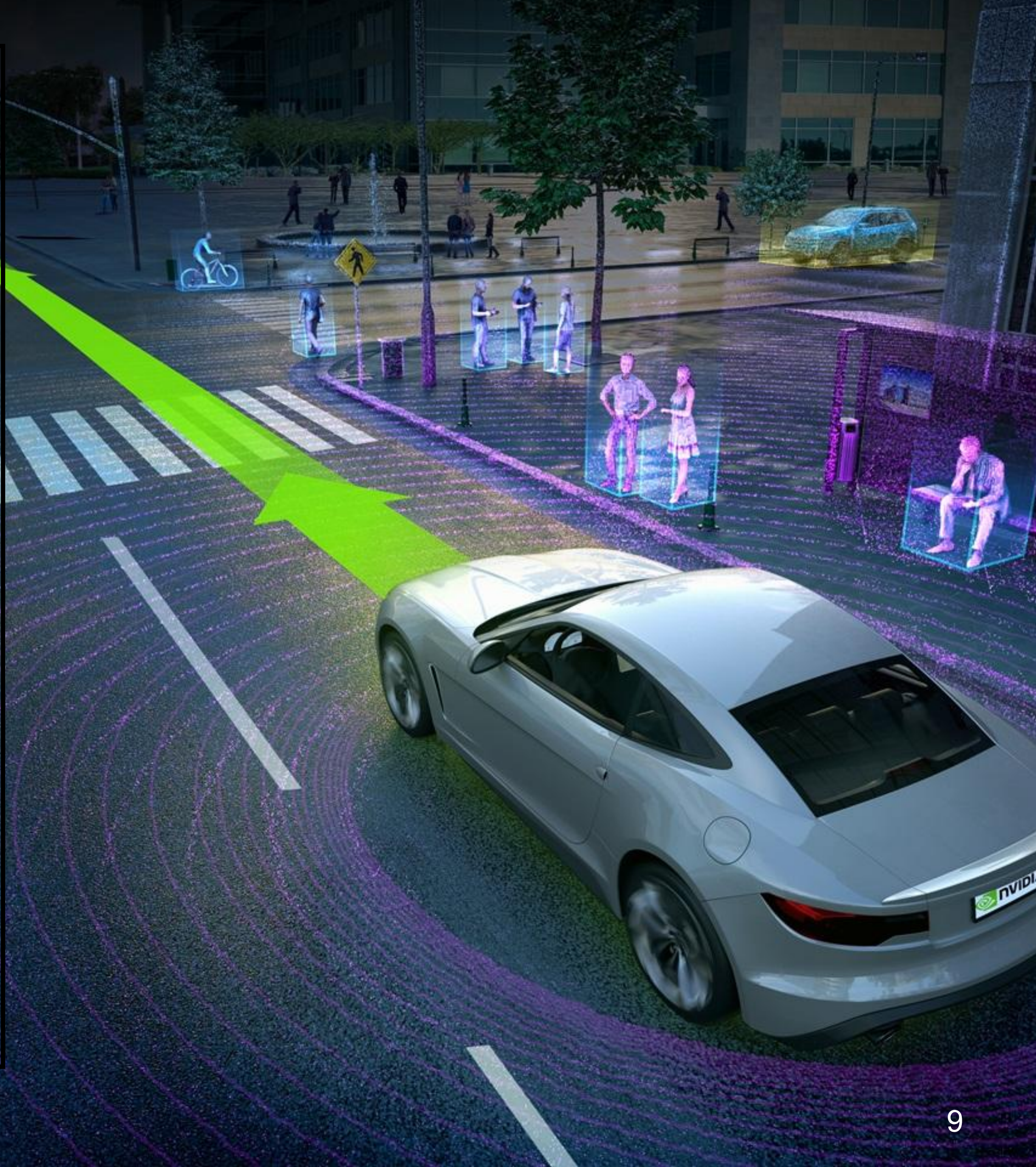
Problem Statement

Would using a dynamic traffic signal scheduling system through computer vision alleviate the heavy traffic in Metro Manila?



Problem Statement

Computer vision that is implemented in software runs slowly because of its immense computing power requirements. Would offloading the slowest parts of the algorithm into the FPGA improve the system by a huge enough margin?



What are we aiming for?

The objectives of this project are the following:

- To develop and implement a **dynamic closed-loop traffic management system covering two intersections**
- Measure the **throughput of the implemented system** and **compare it** against the **throughput of a static scheduling system**
- Implement the Computer Vision learning algorithm on **RASPI** and **FPGA** for **benchmarking**

Computer Vision

Implement a hardware-accelerated Computer Vision algorithm to detect vehicles in traffic

+

Computer Networks

A network based approach to communication between intersections

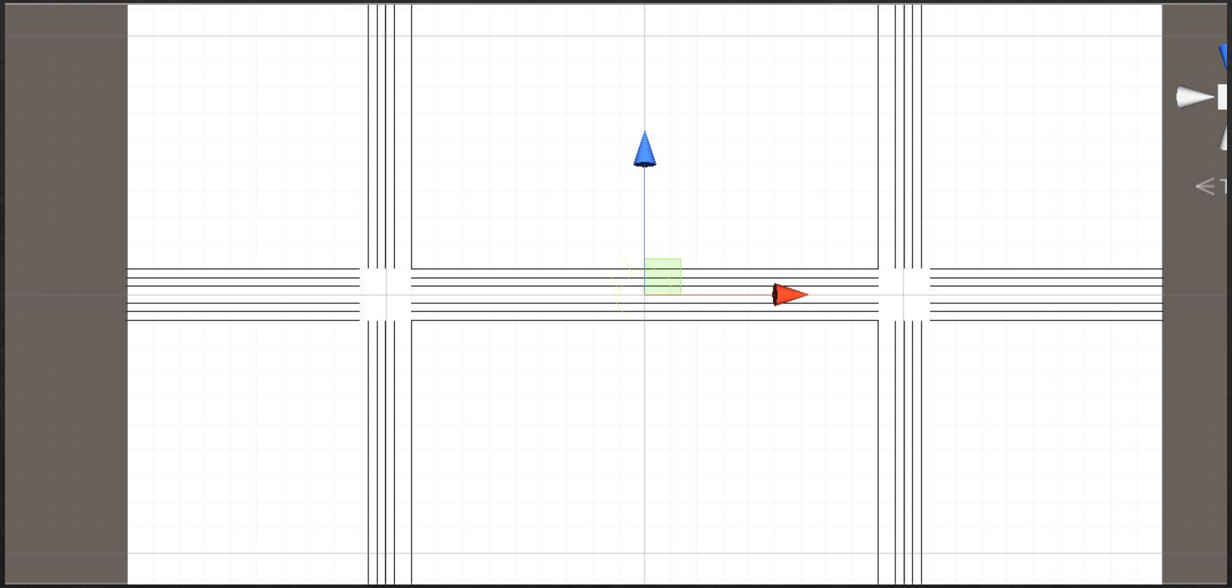
+

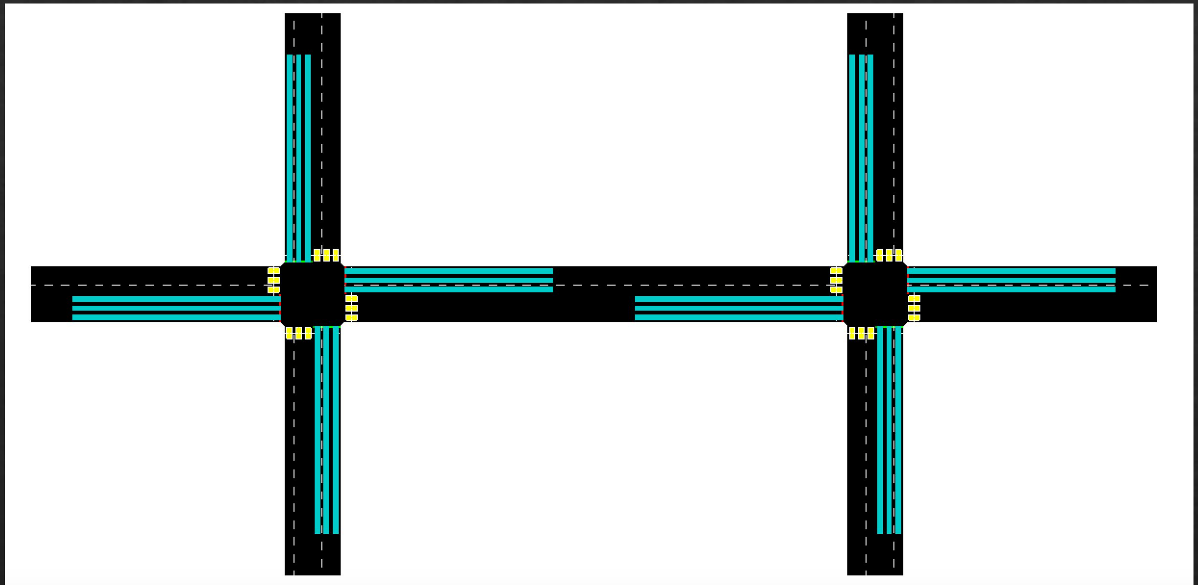
Dynamic Traffic Control

A demand-based green light allocation algorithm (ITLC)

Our System

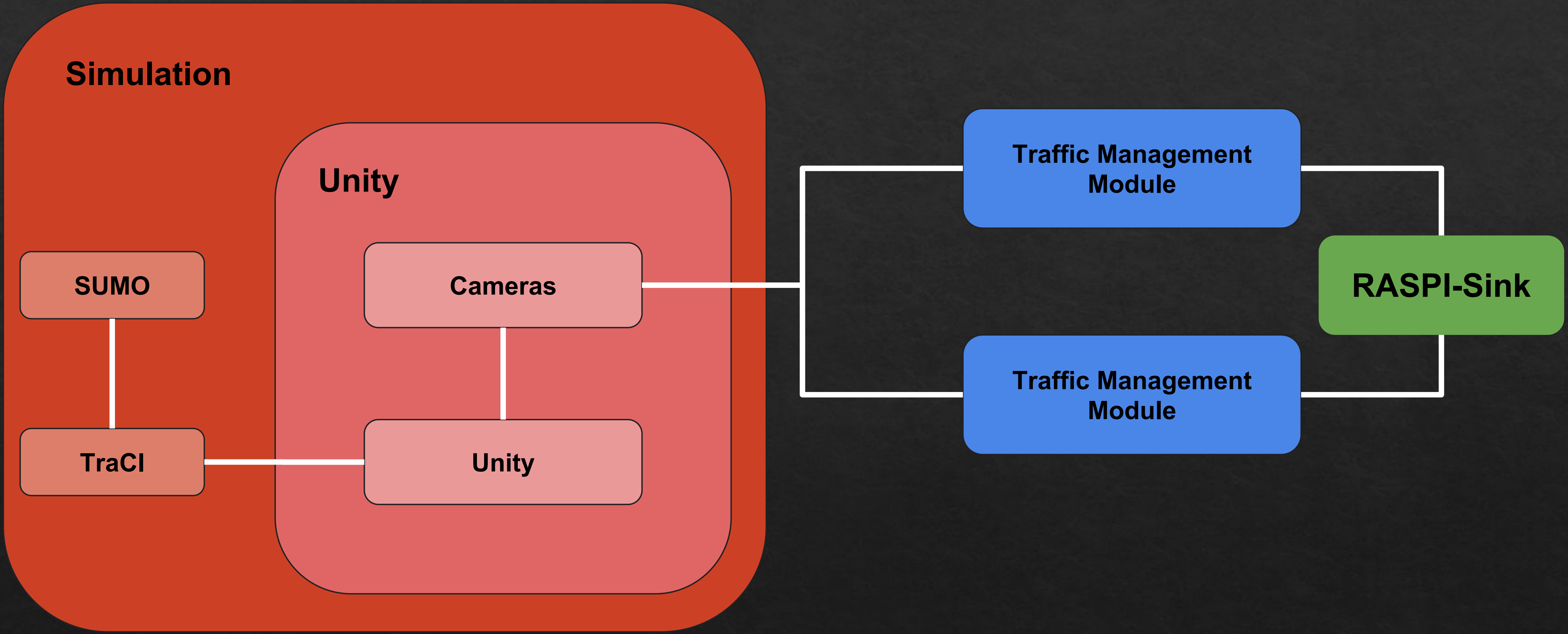
Simulation Setup



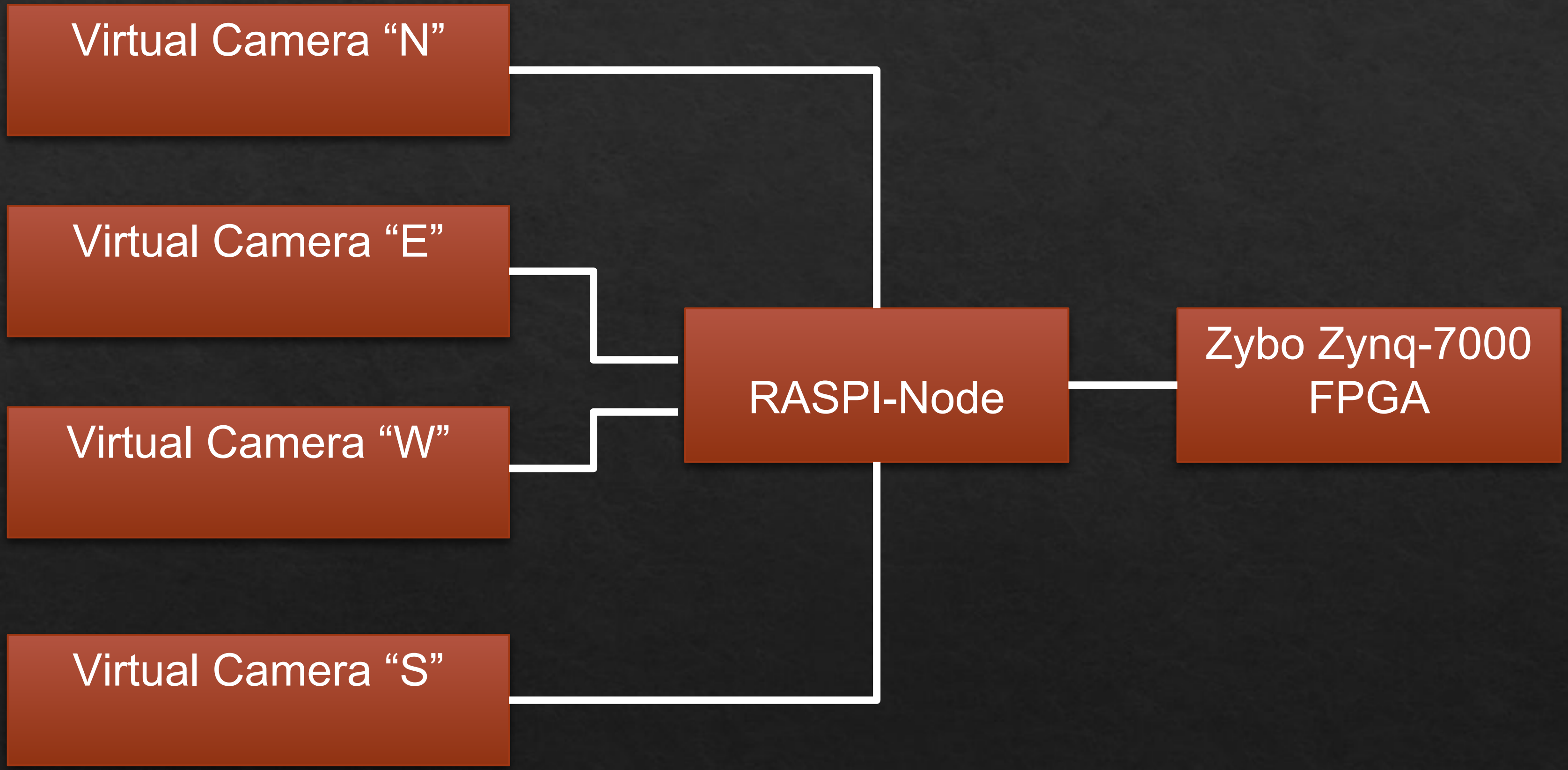


Multiple Intersections: Scalable Approach Through Computer Networks

Representation of the Complete System



Traffic Management Module



Virtual Camera “N”

Virtual Camera “E”

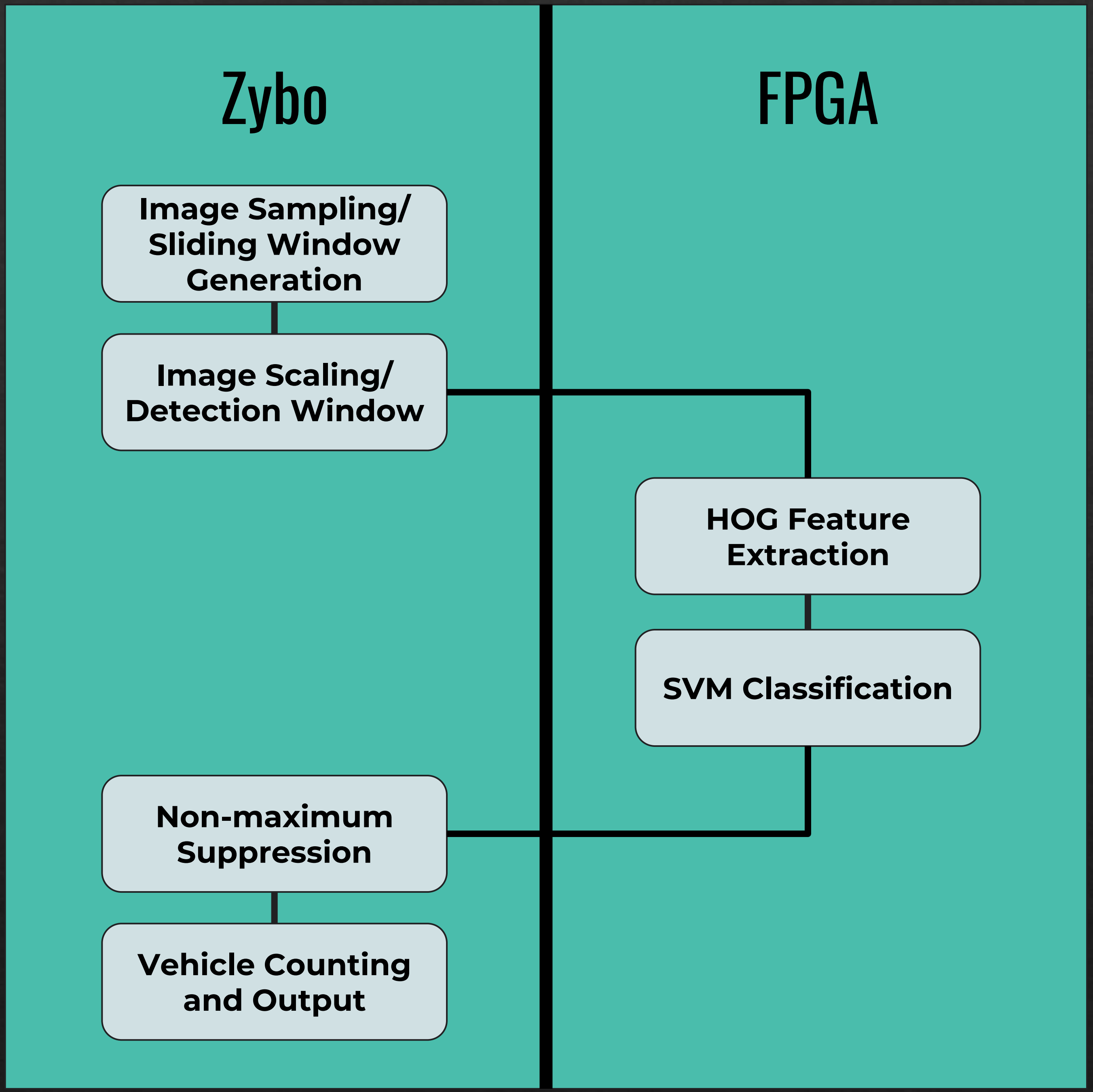
Virtual Camera “W”

Virtual Camera “S”

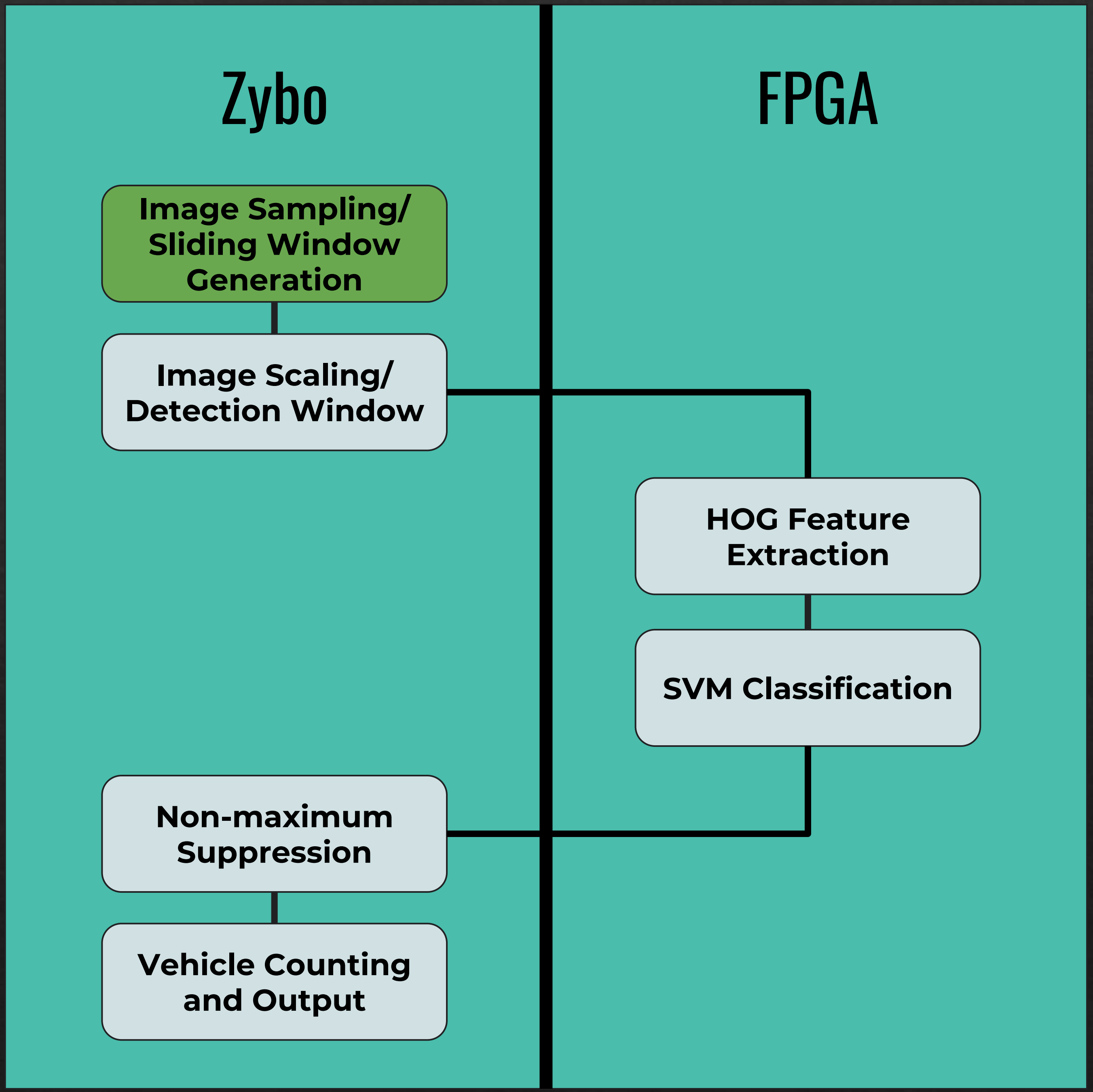
RASPI-Node

Zybo Zynq-7000 FPGA

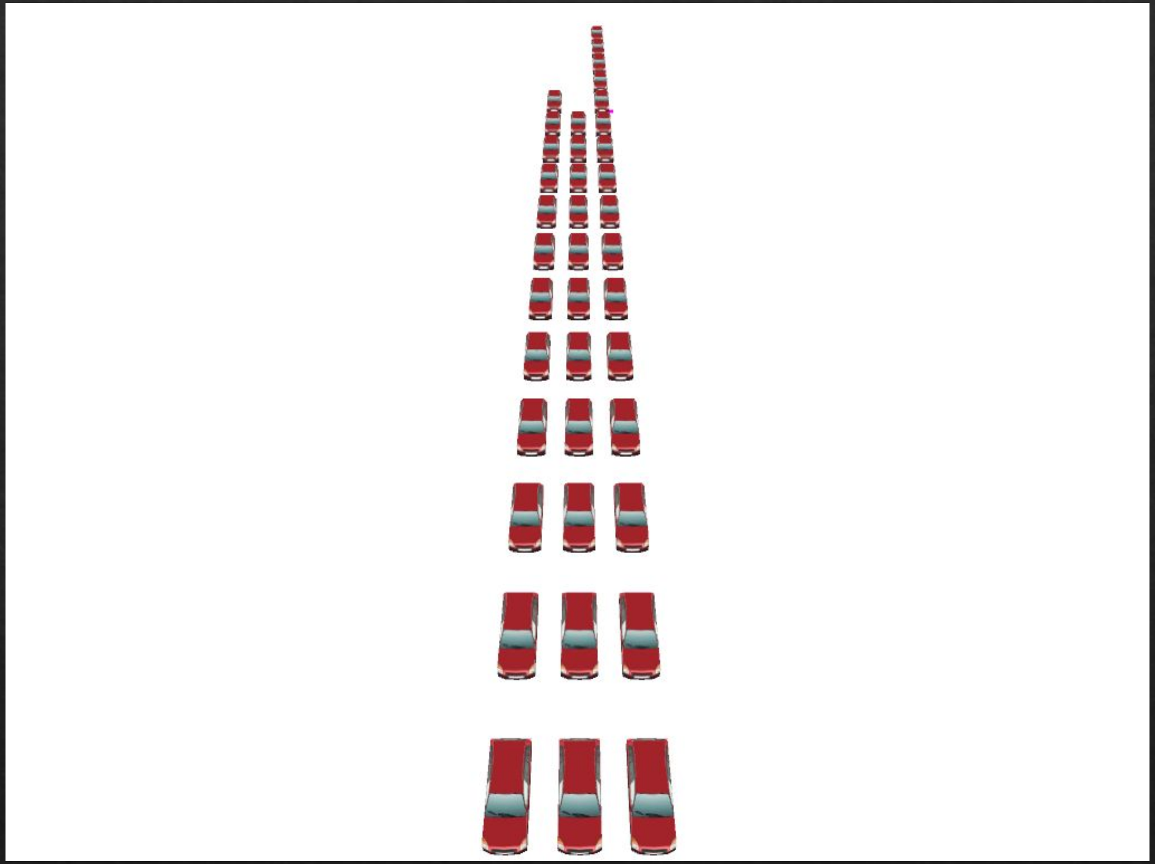
Interplay of the FPGA and the on-board Zybo SoC/SBC


Zybo
FPGA
Image Sampling/ Sliding Window Generation
Image Scaling/ Detection Window
HOG Feature Extraction
SVM Classification
Non-maximum Suppression
Vehicle Counting and Output

Interplay of the FPGA and the on-board Zybo SoC/SBC

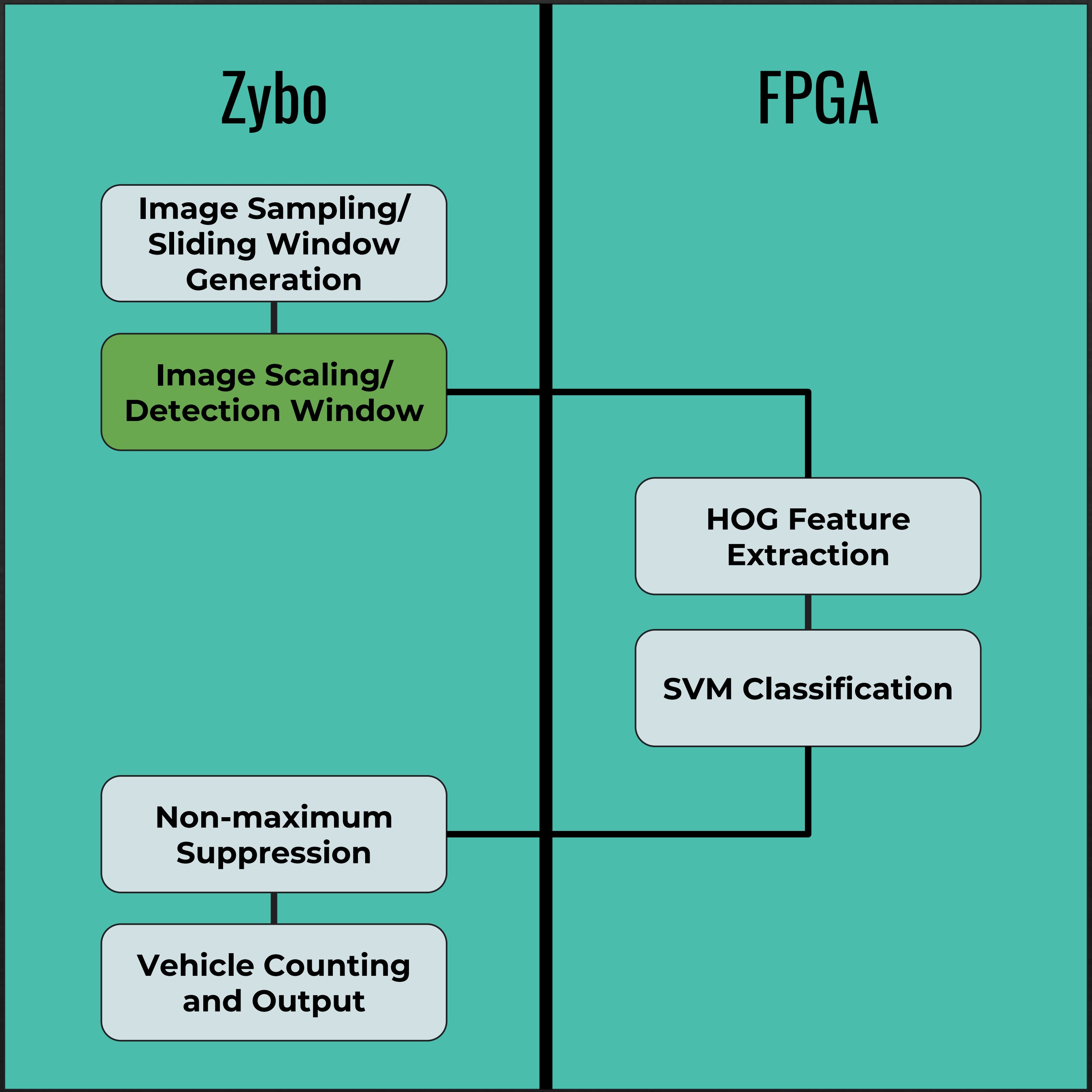

Zybo
FPGA
Image Sampling/ Sliding Window Generation
Image Scaling/ Detection Window
HOG Feature Extraction
SVM Classification
Non-maximum Suppression
Vehicle Counting and Output

An in-game view of a virtual camera



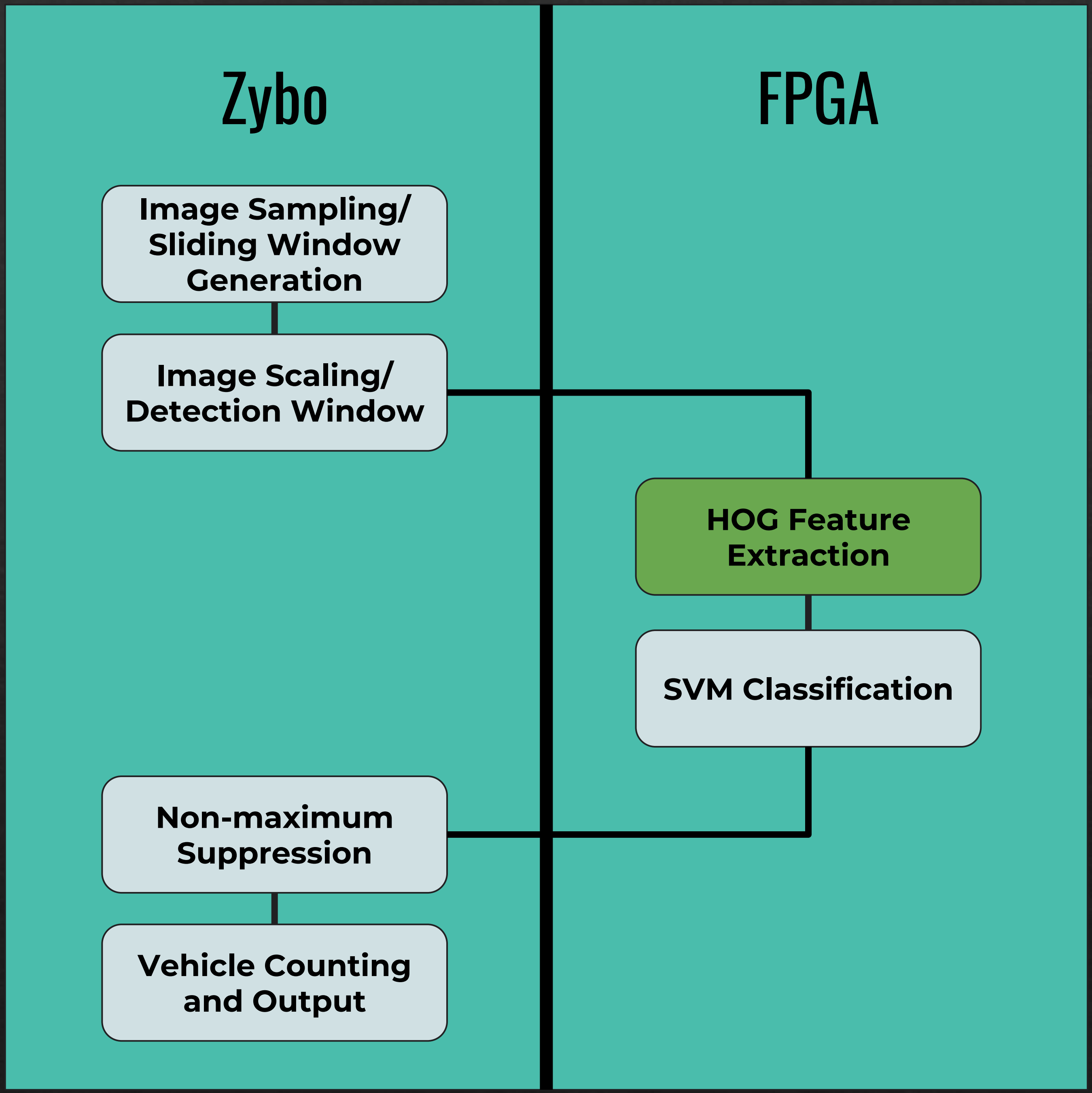
Camera view rendered in Unity

Interplay of the FPGA and the on-board Zybo SoC/SBC

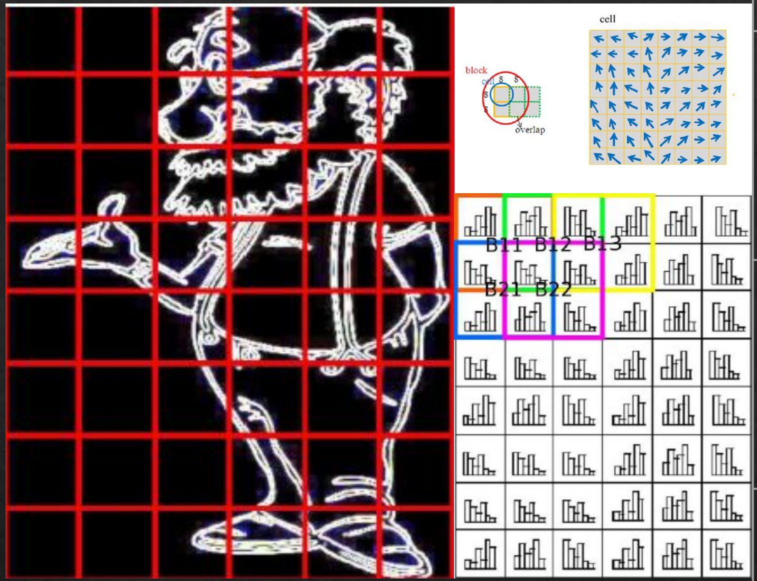

Zybo
FPGA
Image Sampling/ Sliding Window Generation
Image Scaling/ Detection Window
HOG Feature Extraction
SVM Classification
Non-maximum Suppression
Vehicle Counting and Output

Sliding Window Optimization

Interplay of the FPGA and the on-board Zybo SoC/SBC


Zybo
FPGA
Image Sampling/ Sliding Window Generation
Image Scaling/ Detection Window
HOG Feature Extraction
SVM Classification
Non-maximum Suppression
Vehicle Counting and Output

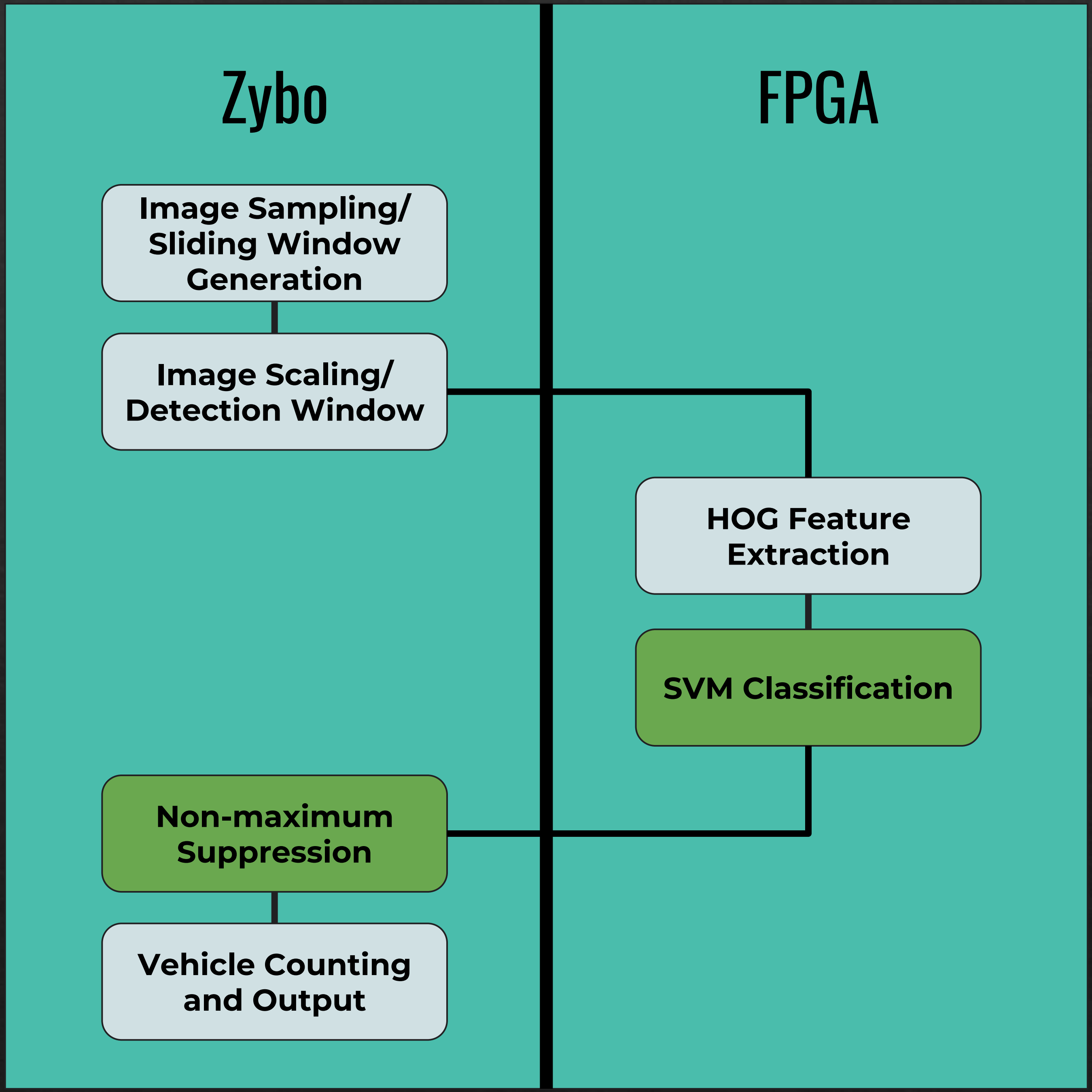
Hardware Approximation of HOG



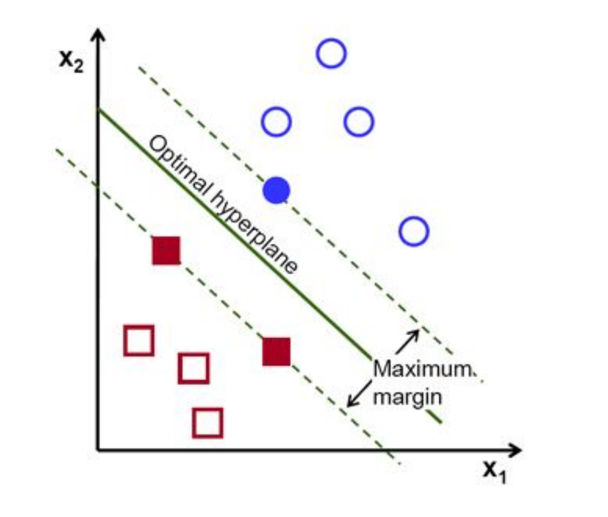
Operation	Software Implementation	Hardware Implementation[1]
Gradient Magnitude	Root Mean Square $mag_{x,y} = \sqrt{dX_{x,y}^2 + dY_{x,y}^2}$	$max\left(a - (a >> 3) + (b >> 1),\ a\right)$ $a = max(dx, dy),\ b = min(dx, dy)$
Orientation Binning	Arctangent $angle_{x,y} = atan\left(\frac{dY_{x,y}}{dX_{x,y}}\right)$	$dx \times \tan\theta_{\mathrm{i}} \leq dy \leq dx \times \tan\theta_{\mathrm{i+1}}$
Block Normalization	Inverse square root $\frac{1}{\sqrt{x}}$	$y_d\left(\frac{3 - x\ y_d{}^2}{2}\right)$ $y_d = Decimal\{\ (x\_{\mathrm{IEEE754}} >> 1) - 0x5F3759DF\ \}$

[1] P. Y. Chen, C. C. Huang, C. Y. Lien and Y. H. Tsai, "An efficient hardware implementation of HOG feature extraction for human detection," in IEEE Transactions on Intelligent Transportation Systems 15.2, 2014, pp. 656-662.

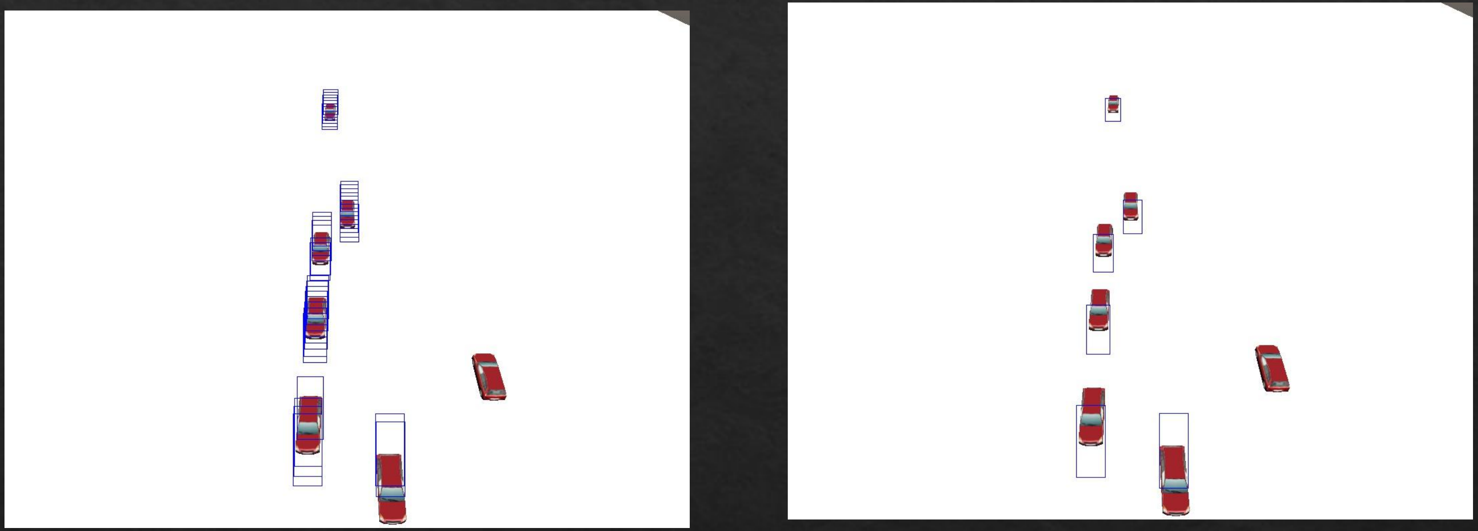
Interplay of the FPGA and the on-board Zybo SoC/SBC


Zybo
FPGA
Image Sampling/ Sliding Window Generation
Image Scaling/ Detection Window
HOG Feature Extraction
SVM Classification
Non-maximum Suppression
Vehicle Counting and Output

Classification to Suppression

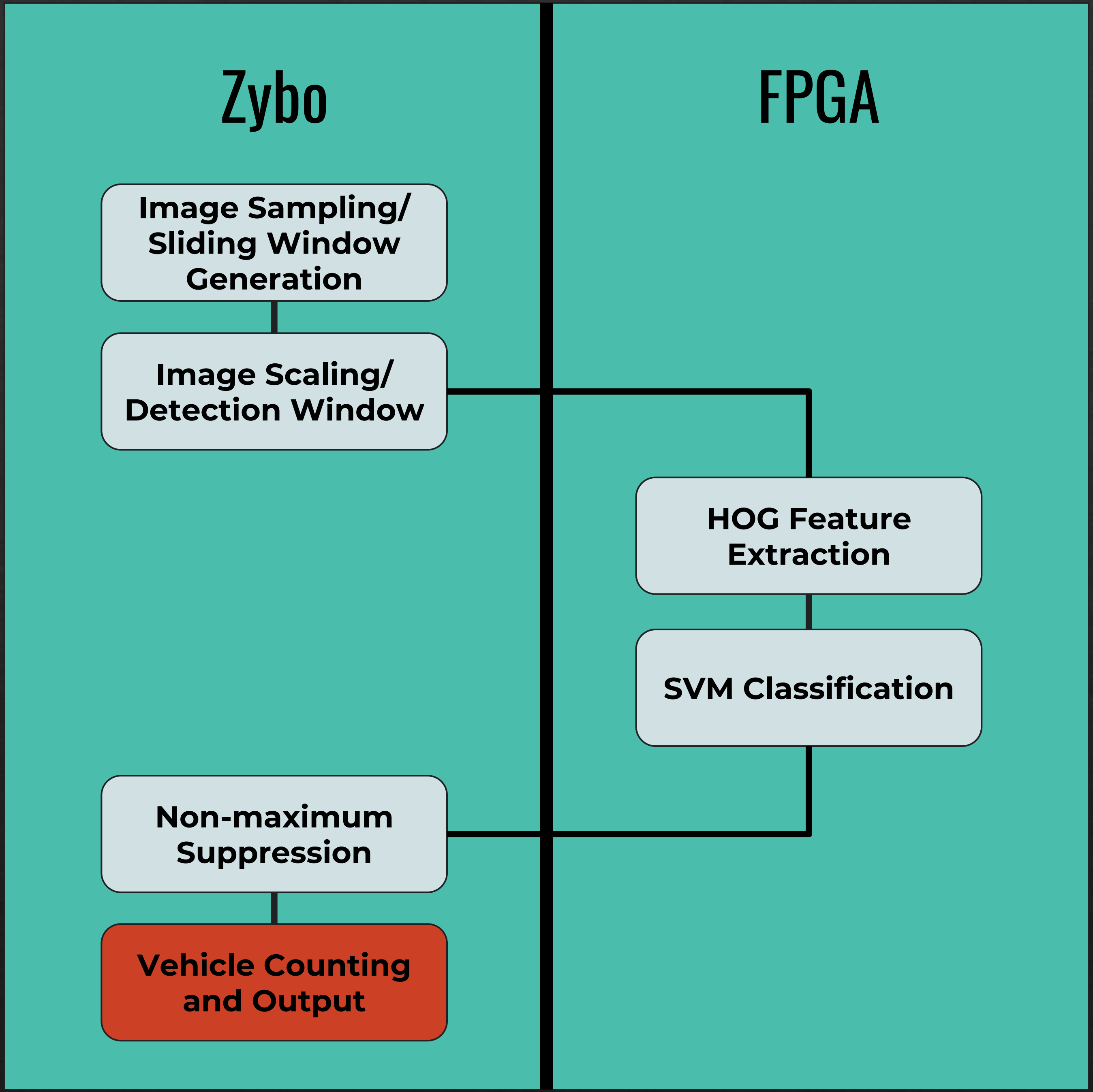


SVM

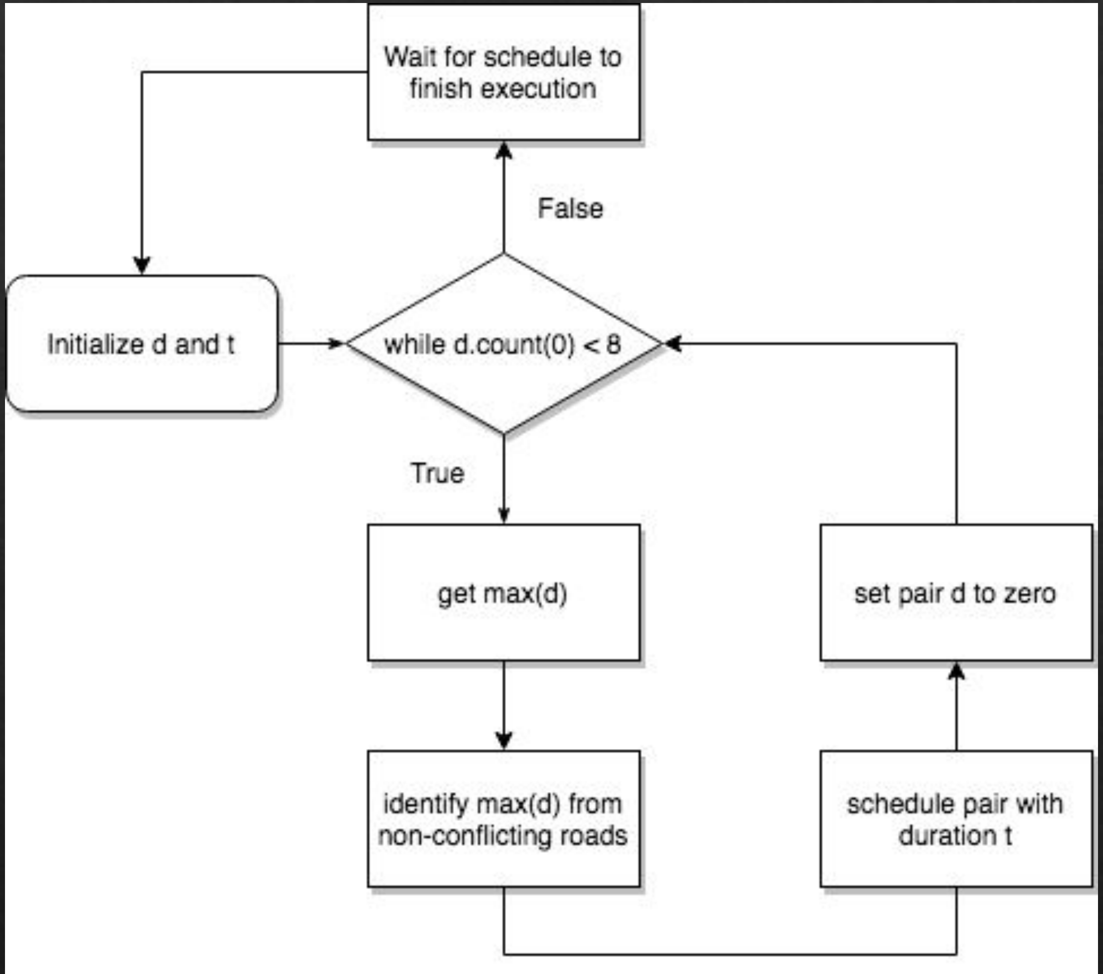


Non-Maximum Suppression

Interplay of the FPGA and the on-board Zybo SoC/SBC

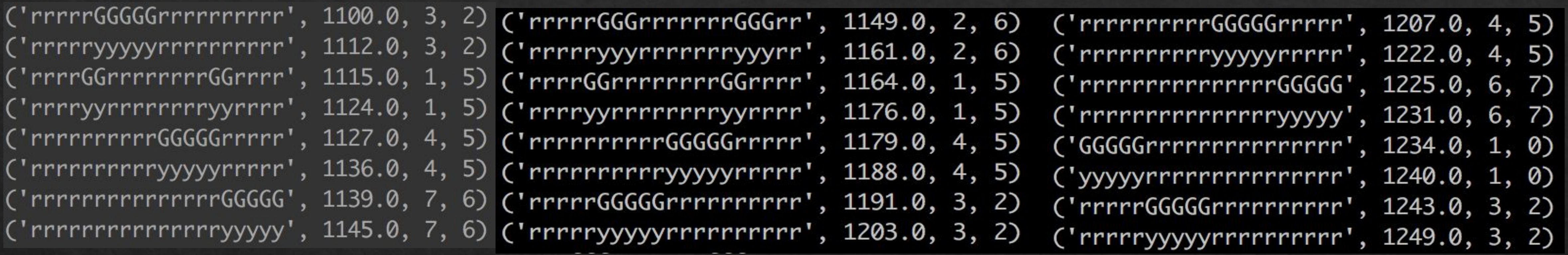

Zybo
FPGA
Image Sampling/ Sliding Window Generation
Image Scaling/ Detection Window
HOG Feature Extraction
SVM Classification
Non-maximum Suppression
Vehicle Counting and Output

Intelligent Traffic Light Scheduling Algorithm


Wait for schedule to finish execution
False
Initialize d and t
while d.count(0) < 8
True
get max(d)
set pair d to zero
identify max(d) from non-conflicting roads
schedule pair with duration t

M. B. Younes and A. Boukerche, “An intelligent traffic light scheduling algorithm through vanets,” in 39th Annual IEEE Conference on Local Computer Networks Workshops, pp. 637– 642, Sept 2014.

Timeline of Signals on a Single Road
Terminal Output

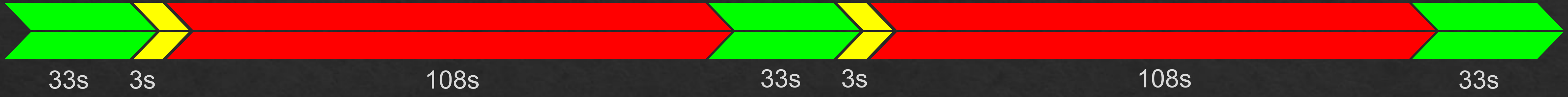


```
('rrrrrGGGGGrrrrrrrrrr', 1100.0, 3, 2) ('rrrrrGGGrrrrrrrGGGrr', 1149.0, 2, 6)  ('rrrrrrrrrrGGGGGrrrrr', 1207.0, 4, 5)
('rrrrryyyyyrrrrrrrrrr', 1112.0, 3, 2) ('rrrrryyyrrrrrrryyyrr', 1161.0, 2, 6)  ('rrrrrrrrrryyyyyrrrrr', 1222.0, 4, 5)
('rrrrGGrrrrrrrrGGrrrr', 1115.0, 1, 5) ('rrrrGGrrrrrrrrGGrrrr', 1164.0, 1, 5)  ('rrrrrrrrrrrrrrrGGGGG', 1225.0, 6, 7)
('rrrryyrrrrrrrryyrrrr', 1124.0, 1, 5) ('rrrryyrrrrrrrryyrrrr', 1176.0, 1, 5)  ('rrrrrrrrrrrrrrryyyyy', 1231.0, 6, 7)
('rrrrrrrrrrGGGGGrrrrr', 1127.0, 4, 5) ('rrrrrrrrrrGGGGGrrrrr', 1179.0, 4, 5)  ('GGGGGrrrrrrrrrrrrrrr', 1234.0, 1, 0)
('rrrrrrrrrryyyyyrrrrr', 1136.0, 4, 5) ('rrrrrrrrrryyyyyrrrrr', 1188.0, 4, 5)  ('yyyyyrrrrrrrrrrrrrrr', 1240.0, 1, 0)
('rrrrrrrrrrrrrrrGGGGG', 1139.0, 7, 6) ('rrrrrGGGGGrrrrrrrrrr', 1191.0, 3, 2)  ('rrrrrGGGGGrrrrrrrrrr', 1243.0, 3, 2)
('rrrrrrrrrrrrrrryyyyy', 1145.0, 7, 6) ('rrrrryyyyyrrrrrrrrrr', 1203.0, 3, 2)  ('rrrrryyyyyrrrrrrrrrr', 1249.0, 3, 2)
```

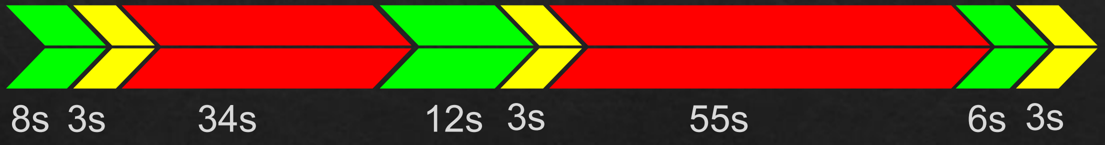
[1] P. S. Chakraborty, P. R. Sinha, and A. Tiwari, “Real time optimized traffic management algorithm for intelligent transportation systems,” in 2015 IEEE International Conference on Computational Intelligence Communication Technology, pp. 744–749, Feb 2015.

Timeline of Signals on a Single Road Graphical Representation (3 green lights)

Static



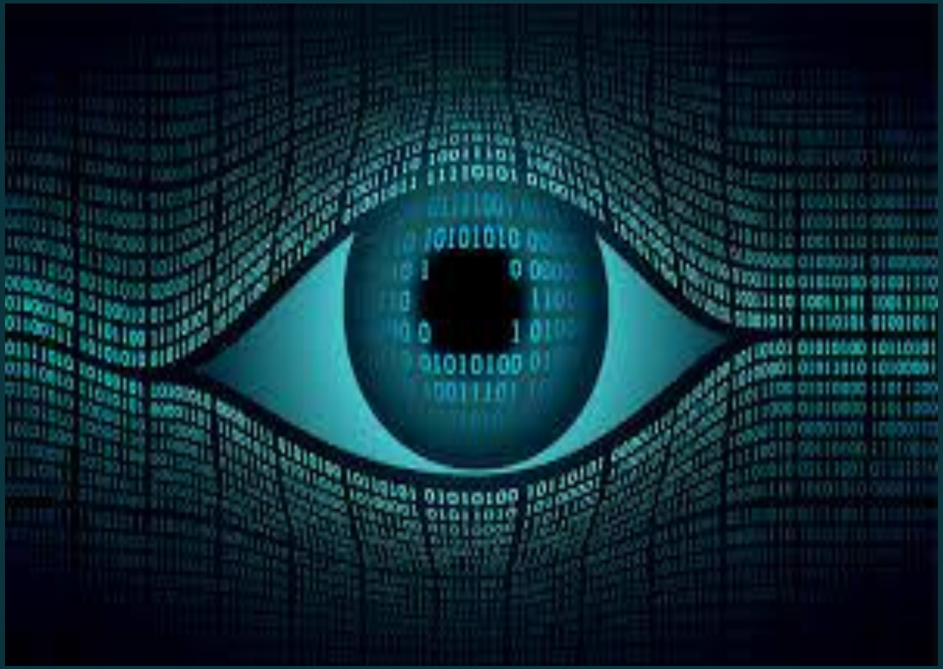
Dynamic



NOT TO SCALE

[1] P. S. Chakraborty, P. R. Sinha, and A. Tiwari, “Real time optimized traffic management algorithm for intelligent transportation systems,” in 2015 IEEE International Conference on Computational Intelligence Communication Technology, pp. 744–749, Feb 2015.

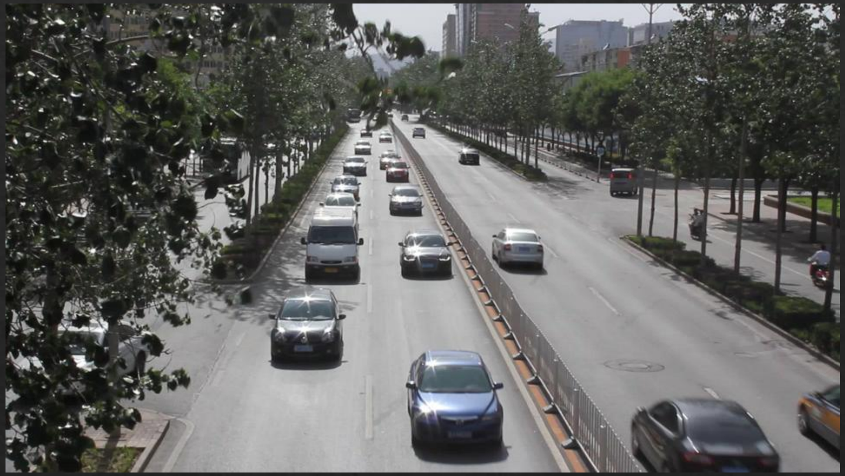
Results and Analysis



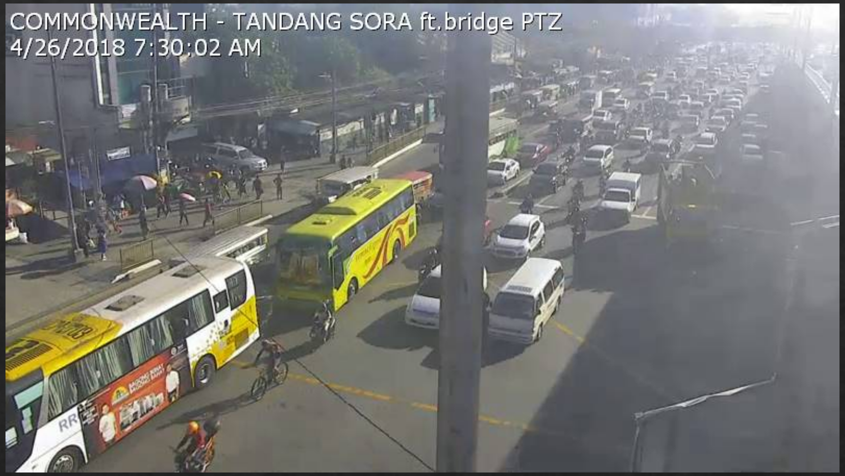
RasPi is slightly more robust than FPGA

		FPGA			RASPI		
Video Name	Description	Accuracy	F1-score	MCC	Accuracy	F1-score	MCC
MVI_39031	Direct front view, daytime, low occlusion	94.5%	0.97	0.946	96.8%	0.98	0.96
MVI_39211		88.3%	0.84	0.84	95.3%	0.85	0.85
MVI_39311		88.5%	0.802	0.795	91%	0.88	0.87
MMDA_3017	EDSA - Aurora Intersection, high occlusion	68.7%	0.51	0.52	71.8%	0.59	0.60
MMDA_3100	Roxas - EDSA Intersection, medium occlusion	80.5%	0.598	0.60	78.7%	0.62	0.62
MMDA_4079	Roxas Boulevard - Quirino Avenue intersection, medium Occlusion	68.99%	0.656	0.65	64.4%	0.66	0.65

Camera placement matters!



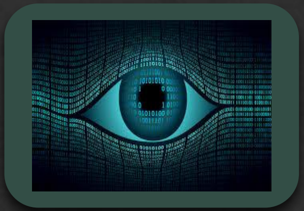
DETRAC Dataset[1]
94% Accuracy



Unusable MMDA Dataset
41% Accuracy

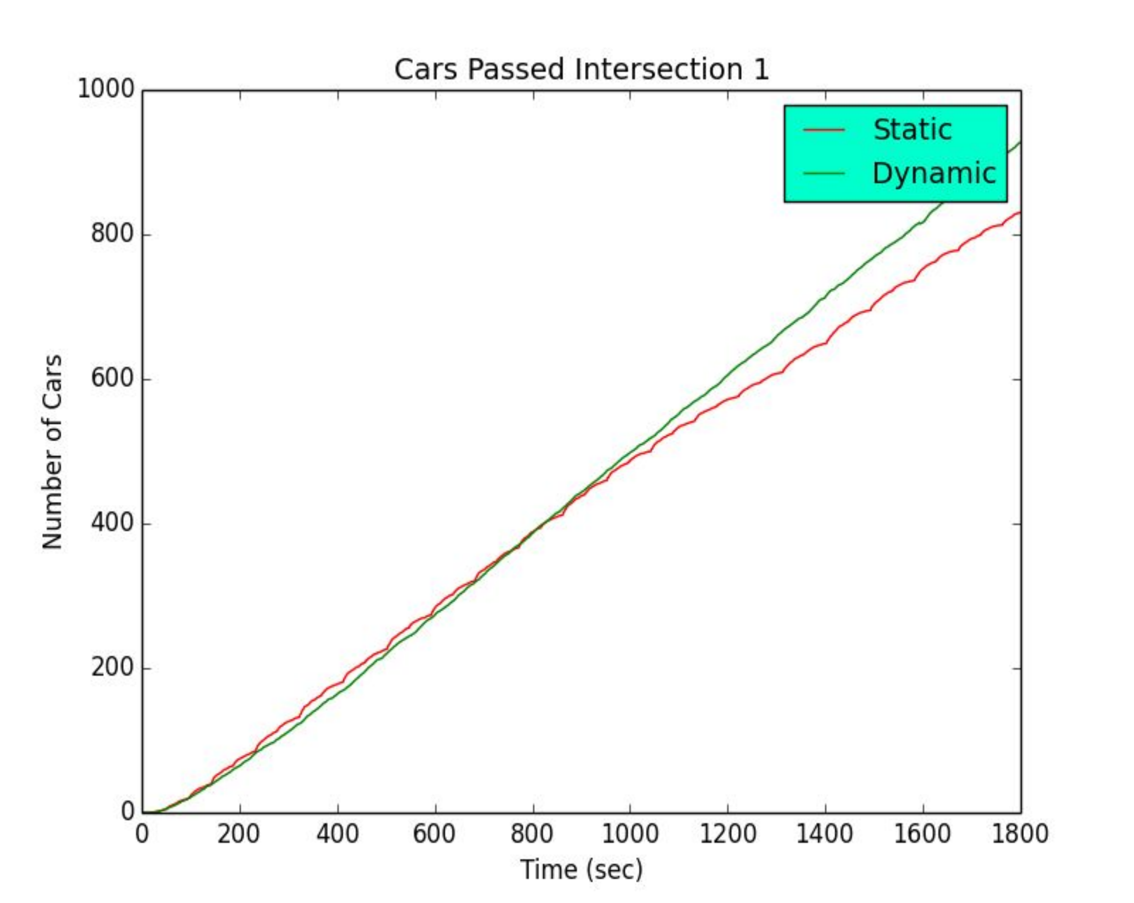
[1] L. Wen, D. Du, Z. Cai, Z. Lei, M. Chang, H. Qi, J. Lim, M. Yang and S. Lyu, "UA-DETRAC: A New Benchmark and Protocol for Multi-Object Detection and Tracking", Arxiv.org, 2018. [Online]. Available: https://arxiv.org/abs/1511.04136. [Accessed: 23- May- 2018].

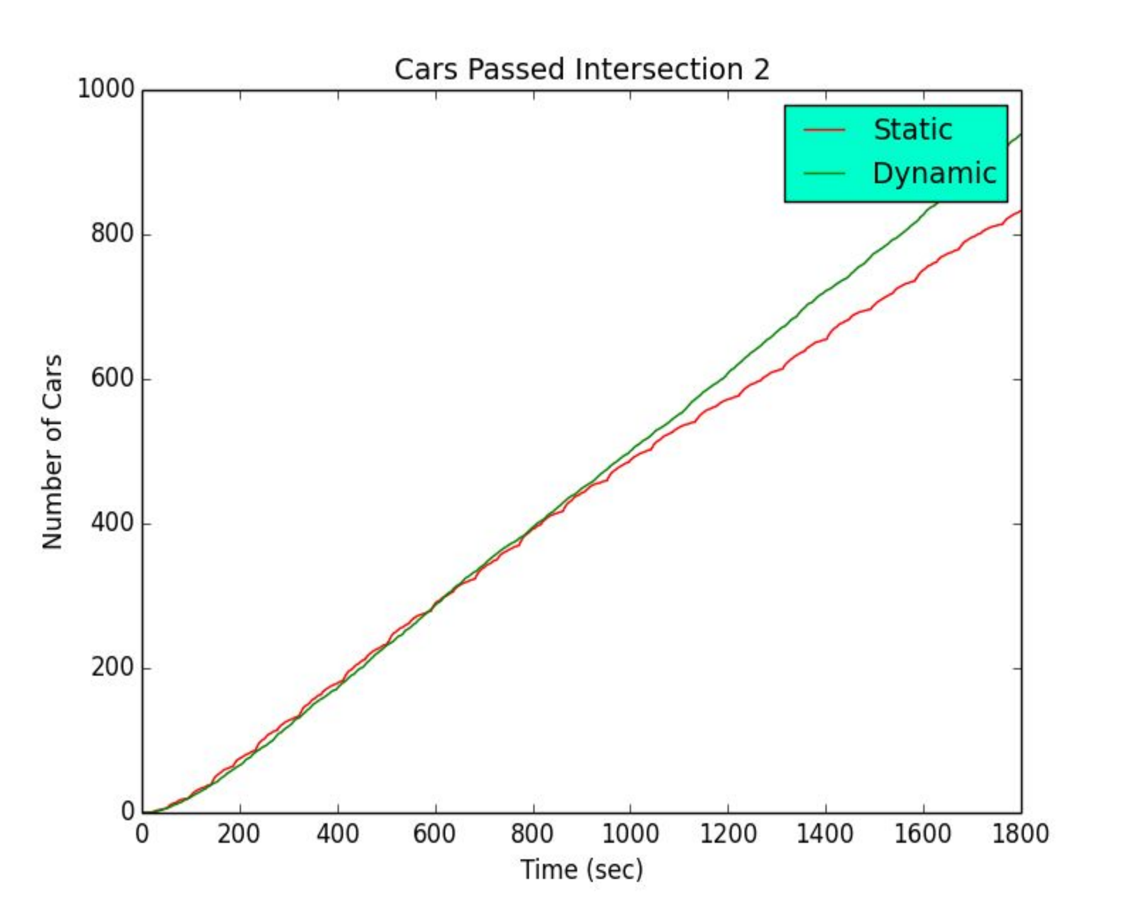
FPGA performs 13x faster than the RasPi



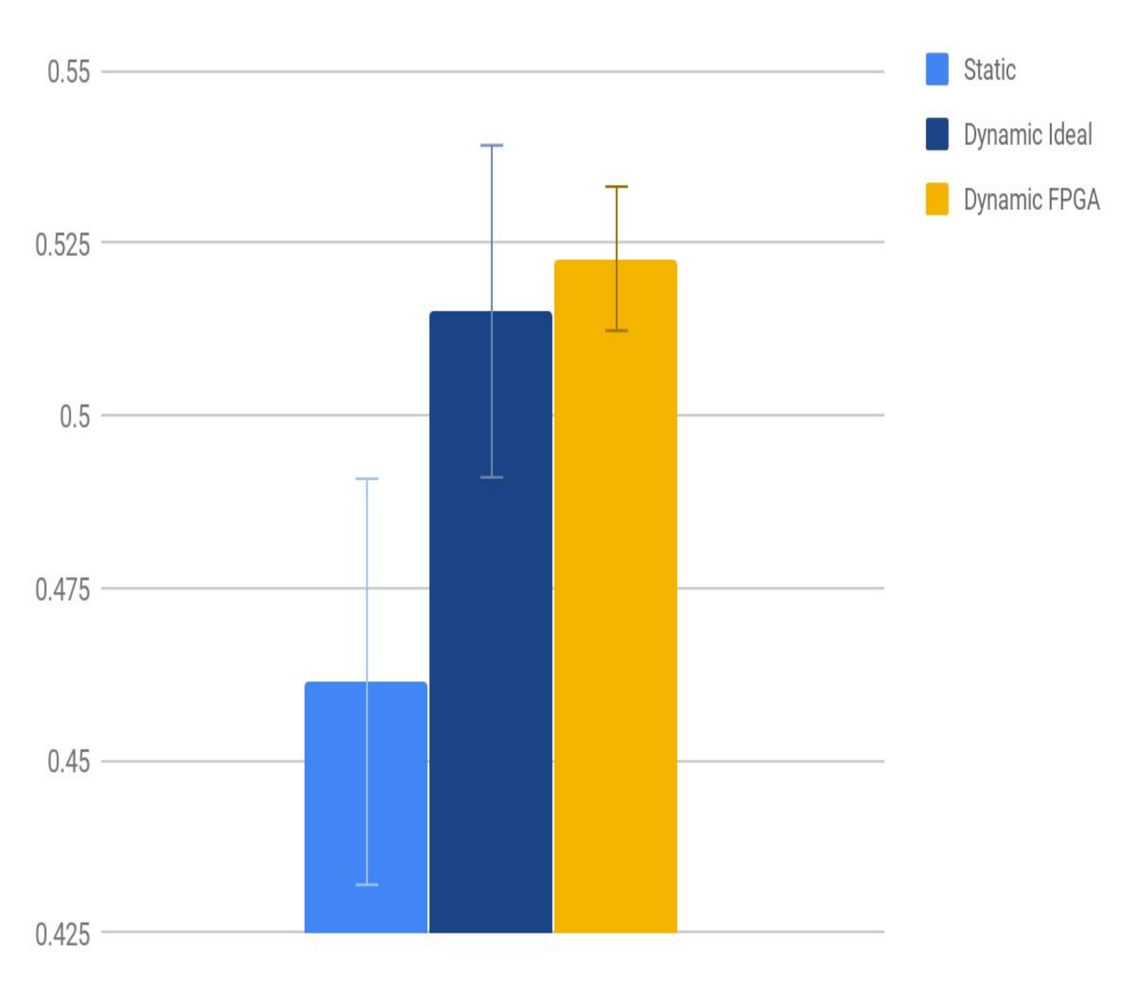
Video Name	Number of Pictures	FPGA Proc Time (sec)	RASPI Proc Time (sec)	Speed Up
MVI_39031	2568	0.26615258	3.6345863	13.65602469
MVI_39211	1703	0.18806675	2.362984754	12.56460667
MVI_39311	1583	0.16284416	2.243037446	13.77413501
MMDA_3017	13366	1.33831585	19.16865903	14.32297084
MMDA_3100	10502	1.05670986	15.10151064	14.29106627
MMDA_4079	11492	1.15280146	16.48849331	14.30297744

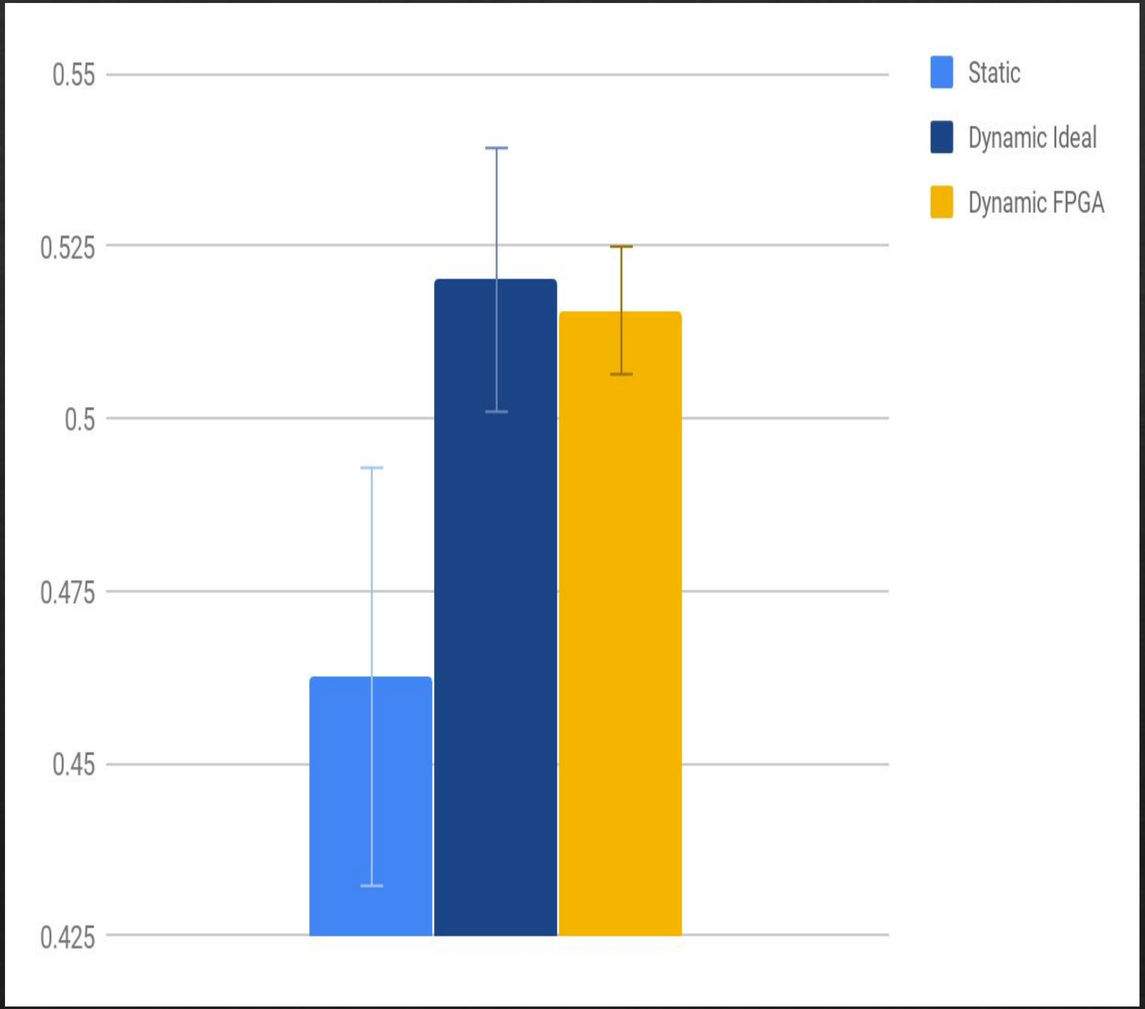
Traffic System Output
Static vs Dynamic





Average Throughput Improvements





Average Throughput Improvements

	Intersection 1	Intersection 2
An Ideal System	11.63%	12.43%
Our System	13.29%	11.48%

Conclusion

- The dynamic scheduling system was able to adjust to heavy traffic and **provide an increase in throughput** in comparison to the static scheduling system.

- Furthermore, the FPGA was shown to provide a **significant speedup** in performing the HOG + SVM algorithm in comparison to the purely software implementation from OpenCV.

Future Works

- This system can be **expanded** to cover more than **two intersections.** Also, this system should be tested through **real life deployment** to increase system robustness on different scenarios.

- To further increase speed up, **sliding window generation can also be implemented in the FPGA**. Also, implementing partial reconfiguration on the FPGA would enable the FPGA to adapt to various lane orientations.

Hardware Accelerated Vehicle Detection Using Computer Vision for a Dynamic Traffic System

Ubiquitous Computing Laboratory
Patrick Celon | Timothy Chua | Paul Ilaga
Advisers : Nestor Tiglao | Jethro Limjoco

UCL04