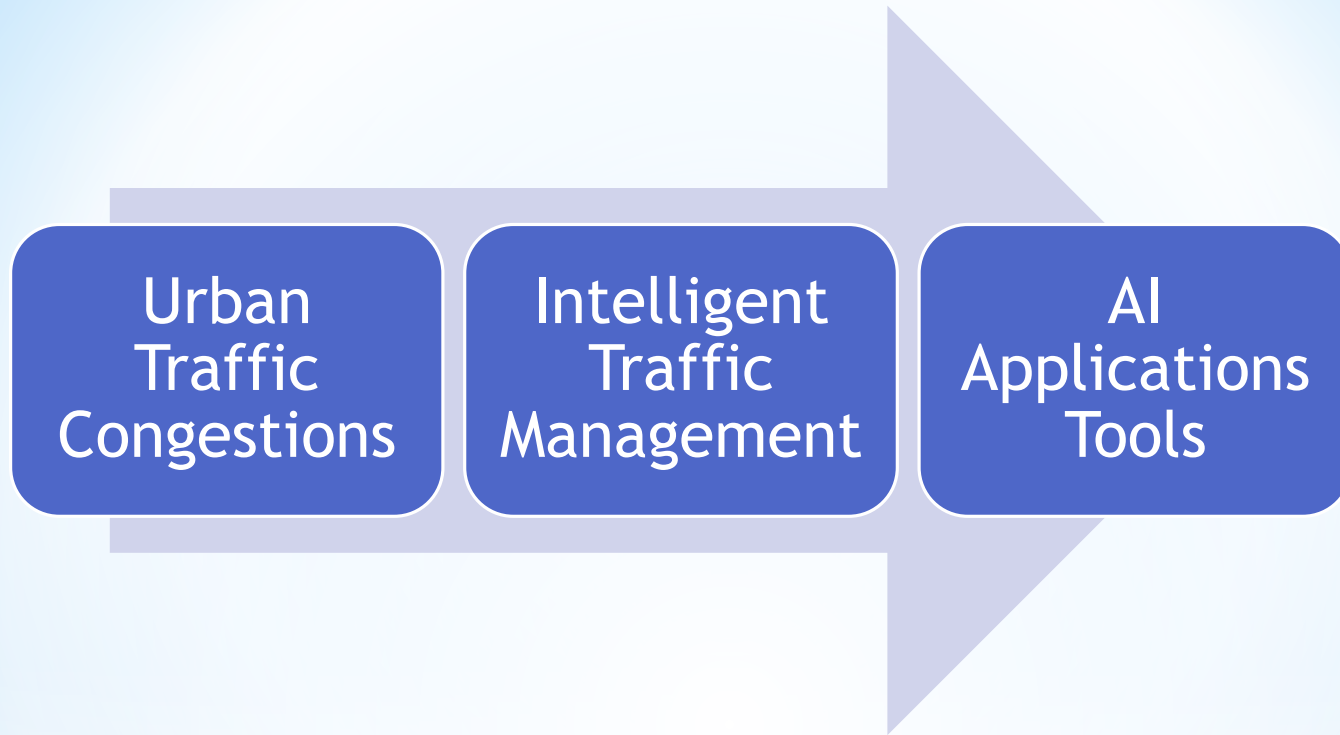


# \* An Application Of Deep Learning In Smart City Traffic Management

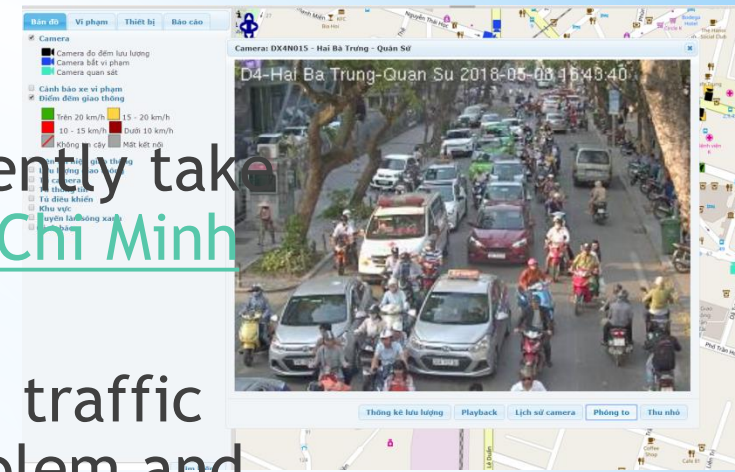
Assoc. Prof. Pham Hong Quang

Center for Informatics and Computing -VAST



# \*Main Contents

- \* Traffic jams in Vietnam frequently take place in [Hanoi](#) capital and [Ho Chi Minh City](#).
- \* Even in new city like Da Nang, traffic congestion became a real problem and time by time more serious in social life.
- \* There are many solutions were planned to reduce traffic congestion in Hanoi, Danang and HCMC

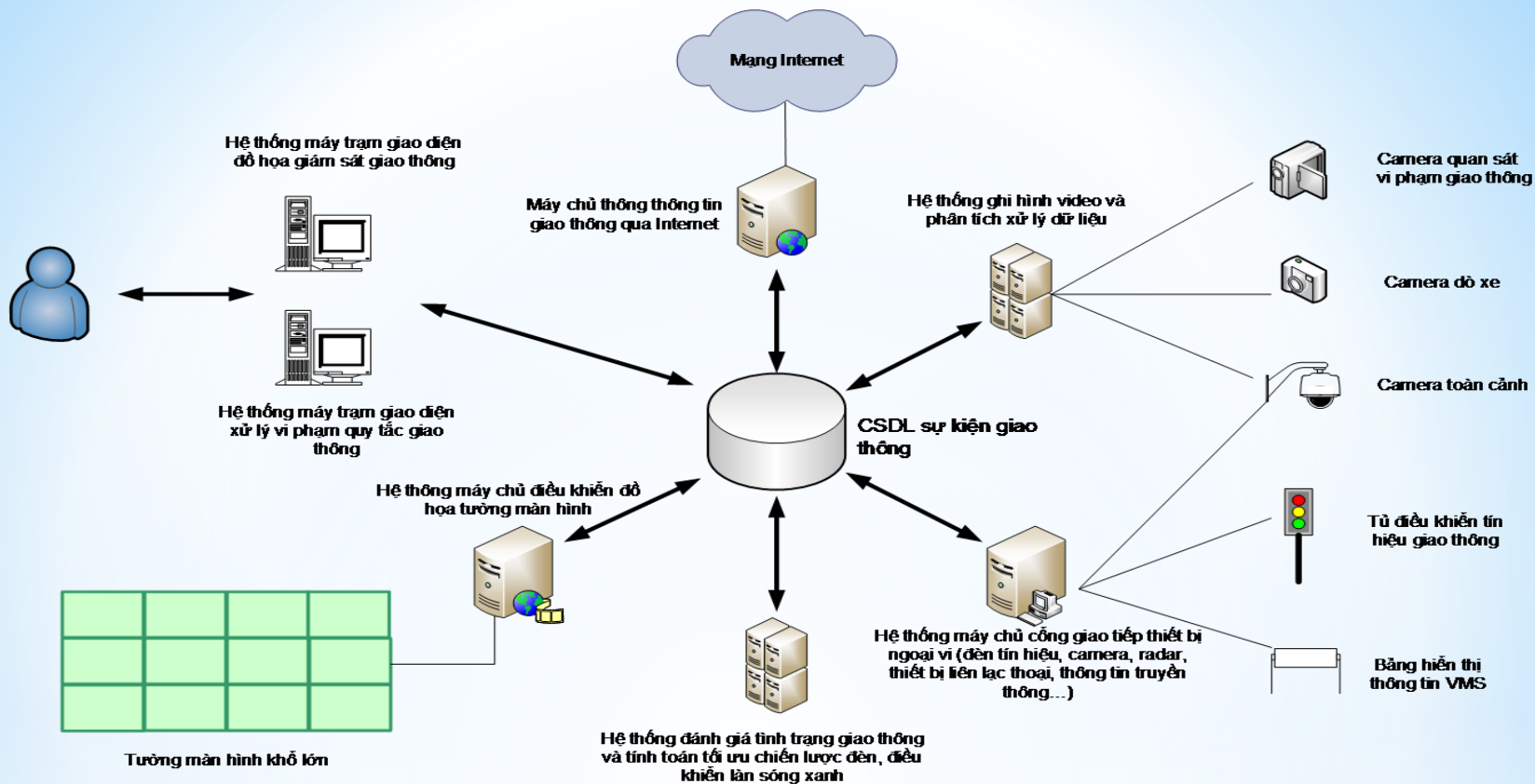


# \* Traffic Problems in big cities in Vietnam

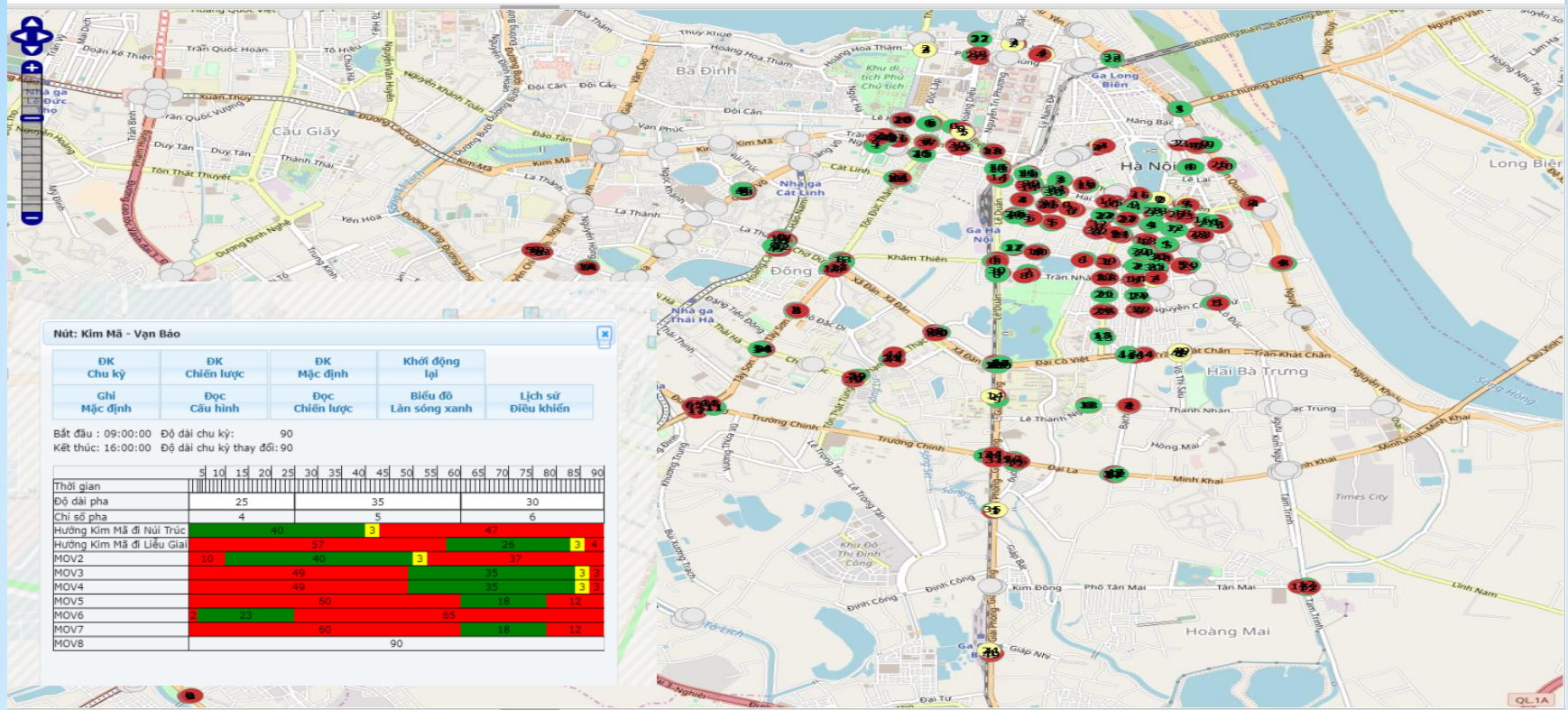
- \* Among those solutions, Intelligent Traffic Management System (ITMS) is a relative low cost, fast-responded effective solution.
- \* The first (ITMS) was installed (2014) and is operating now in Hanoi Traffic Management Center.
- \* The system includes intelligent video analytics to automatically extract traffic information for optimal traffic network management.



# Intelligent Urban Traffic Management



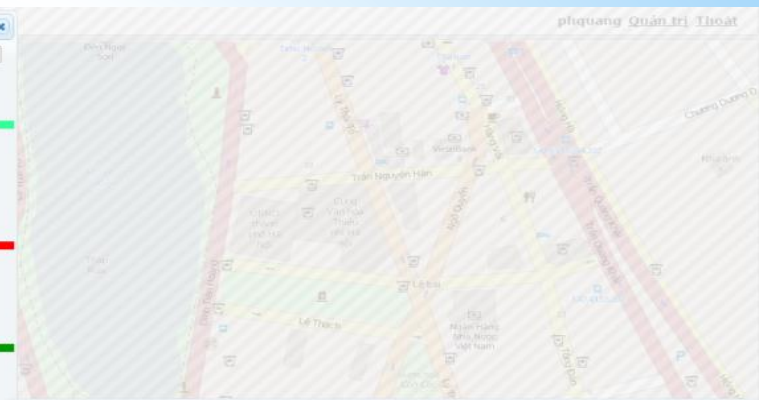
**\* ITMS ICT  
Infrastructure**



\* 308 Centralized traffic signal controllers



# \*Traffic Signal Strategies for Maximizing Vehicle Flow



\* Synchronized traffic signal on green light



Bản đồ | Vĩ phạm | Thiết bị

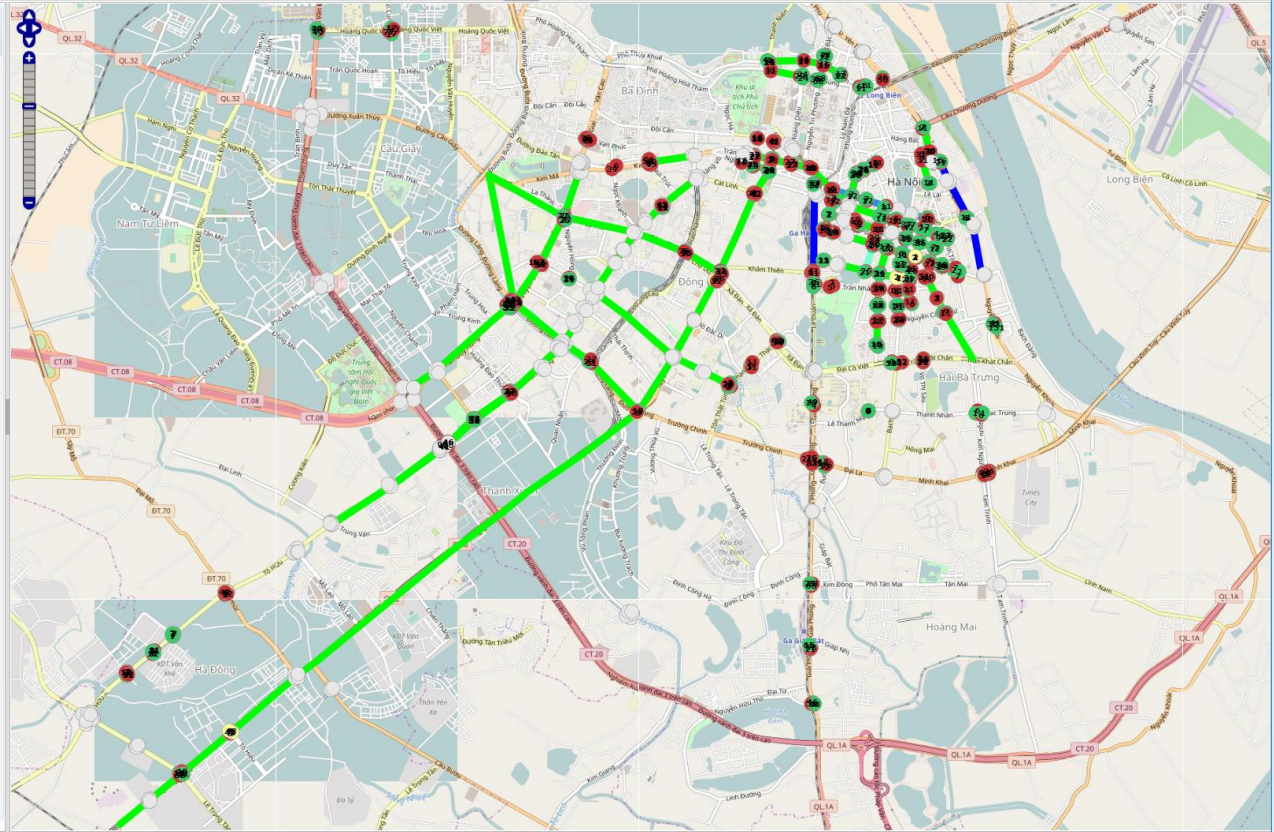
**Bảo cáo**

- Camera
- Cảnh báo xe vi phạm
- Điểm đèn giao thông
- Đèn tín hiệu giao thông
- Lưu lượng giao thông
- Tủ camera
- Tủ thông tin
- Tủ điều khiển
- Khu vực
- Tuyến làn sống xanh

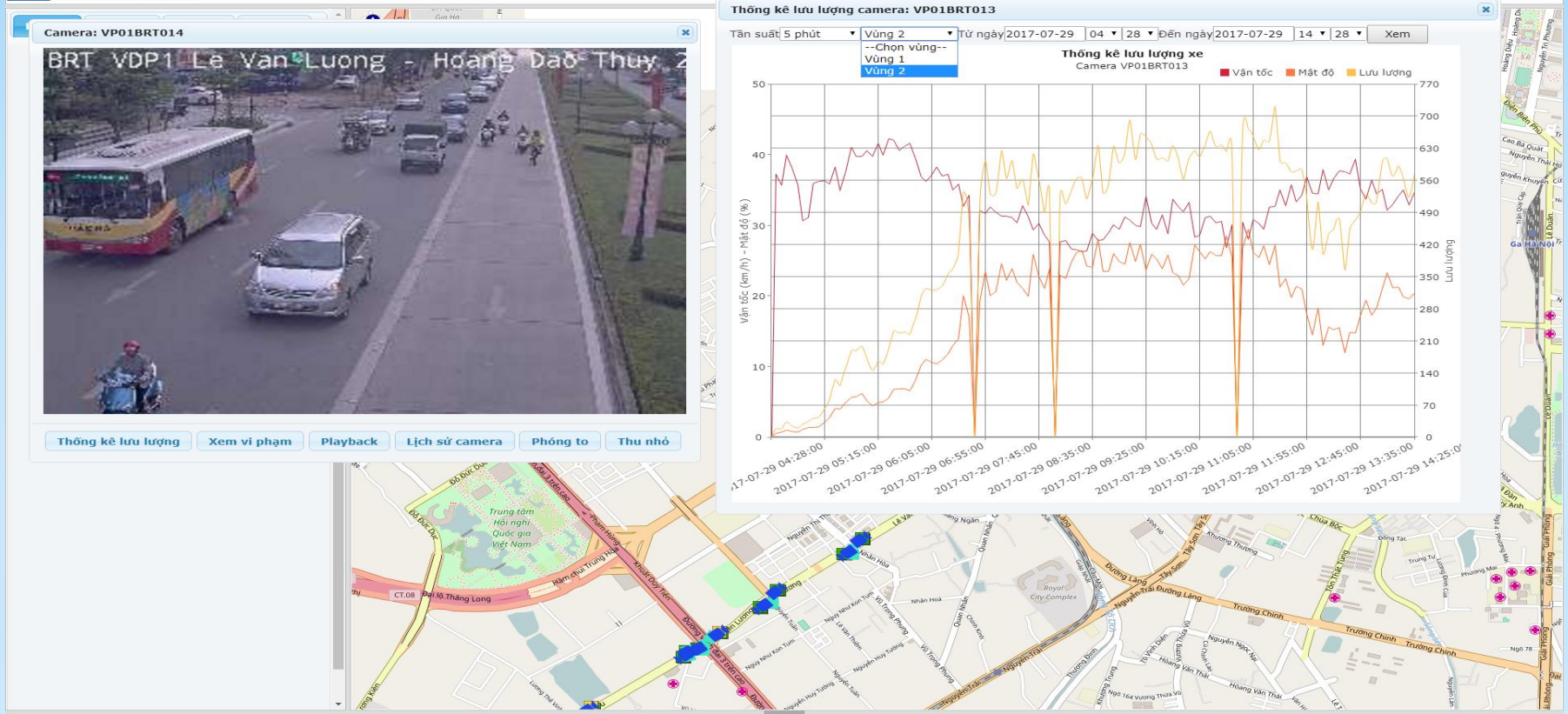
Check All / Uncheck All

- Tuyến Quang Trung
- Tuyến Bà Triệu
- Tuyến Phố Huế
- Tuyến Trần Phú
- Tuyến Thi -> Cửa Nam
- Hai Bà Trưng; Cửa Nam - Điện Biên Phủ -> Hai Bà Trưng - Phan Chu Trinh
- Tuyến Nguyễn Thái Học - Chiểu Tây Đông
- Tuyến Lý Thường Kiệt
- Tuyến Trần Hưng Đạo
- Tuyến Trần Hưng Đạo đi Viên 108
- Tuyến Nguyễn Du
- Tuyến Ngô Quyền
- Tuyến Giảng Võ - Láng Hạ
- Tuyến Lê Văn Lương
- Tuyến Lê Văn Lương hướng đi Láng Hạ
- Tuyến Lý Thường Kiệt - Chiểu Tây Đông
- Tuyến Tôn Đức Thắng - Tây Sơn
- Tuyến Nguyễn Trãi - Hà Đông
- Tuyến Nguyễn Chí Thanh
- Tuyến Láng
- Tuyến Đê La Thành
- Tuyến Thái Hà - Chùa Bộc
- Tuyến Lô Đức
- Tuyến Phan Đình Phùng
- Tuyến Hàng Bông
- Tuyến Lý Thường Kiệt - Chiểu Tây Đông
- Nguyễn Hữu Huân - Lý Thái Tổ
- Tuyến Phan Bội Châu
- Tuyến Quán Sứ - Chiểu Bắc Nam
- Tuyến Quán Sứ - Chiểu Nam Bắc
- Cửa Nam -> Điện Biên Phủ -> Trần Phú
- Phan Chu Trinh; Lô Đức -> Tràng Tiền
- Hai Bà Trưng; Hai Bà Trưng - Phan Chu Trinh -> Cửa Nam - Điện Biên Phủ
- Phoi kim ma - nui truc va kim ma giang van minh
- Phoi Lê Duẩn - N.T.Học và Lê

Tim kiếm

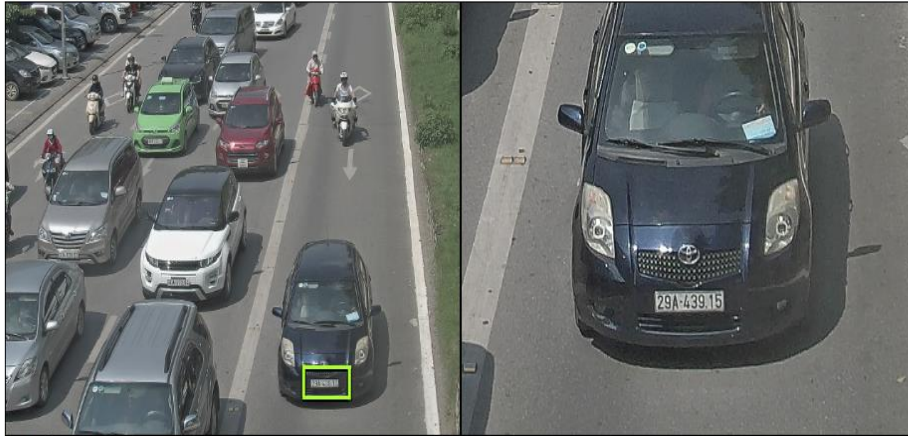


\* 14 Green light corridors



\* 450 intelligent traffic video analytic cameras

Play Video1



Thời điểm: . Tại vị trí:



**Thông số lọc dữ liệu** VP02BRT006. Các khả năng có vi phạm: Điều khiển xe đi không đúng phần đường quy định. Điều khiển xe đi không đúng chiều đường quy định. Điều khiển xe đi vào đường cấm. Điều khiển xe chạy quá tốc độ quy định. Khi tín hiệu đèn giao thông đã chuyển sang màu đỏ nhưng không dừng lại trước vạch dừng mà vẫn tiếp tục đi. Dừng, đỗ xe không đúng quy định. Có chướng ngại vật trên đường. Xe ưu tiên. Không chấp hành hiệu lệnh của đèn tín hiệu giao thông. Xe chở vật liệu nguy hiểm. Xe an cấp. Xe quá khổ, quá tải. Xe hết hạn đăng kiểm. Xe ngoại giao. Xe chữa cháy, cứu thương. Điều khiển xe đi không đúng làn đường quy định;

Cập nhật số liệt

[1](#)
[2](#)
[3](#)
[4](#)
[5](#)
[6](#)
[7](#)
[8](#)
[9](#)
[10](#)
[11](#)
[>](#)

Số bản ghi > 132

STT	Thời gian	Biển số	Vị trí	Loại xe	Vận tốc (km/h)	Cỡ xe	Cảnh báo	TDDL	Trọng số
1	29-07-2017 14:13:38	<del>29A-439.15</del> <a href="#">439.15</a>	Giảng Võ - Trần Huy Liệu(VP02BRT006:Lane L)	Xe con		Xe nhỏ	Điều khiển xe đi không đúng làn đường quy định;	0	1071
2	29-07-2017 13:49:41	<del>19A-159.76</del> <a href="#">159.76</a>	Giảng Võ - Trần Huy Liệu(VP02BRT006:Lane L)	Xe con			Điều khiển xe đi không đúng làn đường quy định;	0	1068
3	29-07-2017 13:49:26	<del>29B-154.72</del> <a href="#">154.72</a>	Giảng Võ - Trần Huy Liệu(VP02BRT006:Lane L)	Xe khách			Xe ưu tiên;	0	68
4	29-07-2017 13:27:56	<del>30E-077.31</del> <a href="#">077.31</a>	Giảng Võ - Trần Huy Liệu(VP02BRT006:Lane L)				Điều khiển xe đi không đúng làn đường quy định;	0	1079
5	29-07-2017 13:27:48	<del>30E-122.23</del> <a href="#">122.23</a>	Giảng Võ - Trần Huy Liệu(VP02BRT006:Lane L)	Xe con			Điều khiển xe đi không đúng làn đường quy định;	0	1072
6	29-07-2017 13:17:44	<del>29B-150.92</del> <a href="#">150.92</a>	Giảng Võ - Trần Huy Liệu(VP02BRT006:Lane L)	Xe khách			Xe ưu tiên;	0	61
7	29-07-2017 13:13:23	<del>29B-150.32</del> <a href="#">150.32</a>	Giảng Võ - Trần Huy Liệu(VP02BRT006:Lane L)	Xe khách			Xe ưu tiên;	0	70
8	29-07-2017 13:03:04	<del>29B-154.16</del> <a href="#">154.16</a>	Giảng Võ - Trần Huy Liệu(VP02BRT006:Lane L)	Xe khách			Xe ưu tiên;	0	61
9	29-07-2017 12:57:50	<del>29B-153.20</del> <a href="#">153.20</a>	Giảng Võ - Trần Huy Liệu(VP02BRT006:Lane L)	Xe khách			Xe ưu tiên;	0	72
10	29-07-2017 12:52:40	<del>29B-146.67</del> <a href="#">146.67</a>	Giảng Võ - Trần Huy Liệu(VP02BRT006:Lane L)	Xe khách			Xe ưu tiên;	0	63
11	29-07-2017 12:44:16	<del>30E-681.35</del> <a href="#">681.35</a>	Giảng Võ - Trần Huy Liệu(VP02BRT006:Lane L)				Điều khiển xe đi không đúng làn đường quy định;	0	1080
12	29-07-2017 12:42:52	<del>29B-154.60</del> <a href="#">154.60</a>	Giảng Võ - Trần Huy Liệu(VP02BRT006:Lane L)	Xe khách			Xe ưu tiên;	0	67

# \* ALPR and traffic violation detection

**Bản đồ** | **Vị phạm** | **Thiết bị** | **Bảo cáo**

- Camera**
  - Camera đo đếm lưu lượng
  - Camera bắt vi phạm
  - Camera quan sát
- Cảnh báo xe vi phạm
- Điểm đếm giao thông
- Bảng điện tử
- Đèn tín hiệu giao thông
- Lưu lượng giao thông
- Tủ camera
- Tủ thông tin
- Tủ điều khiển

**Camera: SP1N006** | **Camera: VP1N015**

Camera GS tốc độ-CMTB-K181-01/20/2017-15:44:20  
**Vận tốc: 33.8 (km/h)**

**Xem vi phạm** | **Playback** | **Lịch sử camera** | **Phóng to** | **Thu nhỏ**

# \* Speed and Traffic Signal Violation Automatic detections

CadPro **HỆ THỐNG GIÁM SÁT VÀ ĐIỀU KHIỂN MẠNG GIAO THÔNG** phquang Thoát

Bản đồ | Vi phạm | Thiết bị | Báo cáo

Camera

- Camera đo đếm lưu lượng
- Camera bắt vi phạm
- Camera quan sát


Cảnh báo xe vi phạm

Đếm đếm giao thông

- Trên 20 km/h
- 15 - 20 km/h
- 10 - 15 km/h
- Dưới 10 km/h

Camera: DX4N015 - Hai Bà Trưng - Quán Sứ

D4-Hai Ba Trưng-Quan Su 2018-05-09 11:33:24



Tràng Thi - Hai Bà Trưng

Thời điểm	Vận tốc (km/h)	Mật độ (%)	Lưu lượng
2018-05-09 11:30:00	12.6	20.7	133
2018-05-09 11:25:00	11.5	19.1	112
2018-05-09 11:20:00	9.1	27.9	129
2018-05-09 11:15:00	12.1	16.2	101
2018-05-09 11:10:00	7.9	20.4	82

Thống kê lưu lượng | Playback | Lịch sử camera | Phóng to | Thu nhỏ

\* Automatic traffic flow estimation

Bản đồ
Vi phạm
Thiết bị
Báo cáo

Lưu lượng giao thông

Tần suất đếm lưu lượng

5 phút

Từ ngày: 2017-08-12    Giờ: 07    Phút: 00    Từ tốc độ:    Đến tốc độ:   

Đến ngày: 2017-08-12    Giờ: 08    Phút: 59    Từ mật độ:    Đến mật độ:   

Chọn mật độ giao thông

Điểm đếm xe

--- Tất cả điểm đếm xe--

Xem
Xuất Excel

Hàng Khay					
<a href="#">Tôn Đức Thắng - Cát Linh -&gt; Tôn Đức Thắng - Hàng Cháo</a>	<a href="#">2017-08-12 07:00:00</a>	27.21	5.37	142.01	
<a href="#">Tôn Đức Thắng - Cát Linh -&gt; Tôn Đức Thắng - Hàng Cháo</a>	<a href="#">2017-08-12 07:05:00</a>	26.51	6.9	177.86	
<a href="#">Tôn Đức Thắng - Cát Linh -&gt; Tôn Đức Thắng - Hàng Cháo</a>	<a href="#">2017-08-12 07:10:00</a>	27.08	7.93	208.75	
<a href="#">Tôn Đức Thắng - Cát Linh -&gt; Tôn Đức Thắng - Hàng Cháo</a>	<a href="#">2017-08-12 07:15:00</a>	26.83	5.66	147.7	
<a href="#">Tôn Đức Thắng - Cát Linh -&gt; Tôn Đức Thắng - Hàng Cháo</a>	<a href="#">2017-08-12 07:20:00</a>	27.77	4.91	132.65	
<a href="#">Tôn Đức Thắng - Cát Linh -&gt; Tôn Đức Thắng - Hàng Cháo</a>	<a href="#">2017-08-12 07:25:00</a>	26.02	7.22	182.71	
<a href="#">Tôn Đức Thắng - Cát Linh -&gt; Tôn Đức Thắng - Hàng Cháo</a>	<a href="#">2017-08-12 07:30:00</a>	26.86	10.86	283.58	
<a href="#">Tôn Đức Thắng - Cát Linh -&gt; Tôn Đức Thắng - Hàng Cháo</a>	<a href="#">2017-08-12 07:35:00</a>	26.93	9.31	243.9	
<a href="#">Tôn Đức Thắng - Cát Linh -&gt; Tôn Đức Thắng - Hàng Cháo</a>	<a href="#">2017-08-12 07:40:00</a>	25.87	6.59	165.8	
<a href="#">Tôn Đức Thắng - Cát Linh -&gt; Tôn Đức Thắng - Hàng Cháo</a>	<a href="#">2017-08-12 07:45:00</a>	25.89	8.49	213.8	
<a href="#">Tôn Đức Thắng - Cát Linh -&gt; Tôn Đức Thắng - Hàng Cháo</a>	<a href="#">2017-08-12 07:50:00</a>	24.83	9.43	227.67	

Không kết nối;    Từ Điều Khiển BGT\_TH\_KM5+360/Tổ Hữu - Km5+360(2017-08-11 14:50:28):Không kết nối;    Từ Điều Khiển CAG\_N4BT\_DA/Ngã 4 biển thể Đông Anh(2017-08-11

# \* Real-time road surface flow occupation estimation

Camera

Camera: DX2N034 - Lê Duẩn - Khâm Thiên

DX2-Le Duan-Kham Thien 2018-05-09 11:52:01



Khâm Thiên - Lê Duẩn đi Ô Chờ Dừa

Thời điểm	Vận tốc (km/h)	Mật độ (%)	Lưu lượng
2018-05-09 11:50:00	18.3	18.4	290
2018-05-09 11:45:00	16.8	21.2	306
2018-05-09 11:40:00	15.3	22.1	291
2018-05-09 11:35:00	19.8	17.3	295
2018-05-09 11:30:00	19.6	17.8	300



# Vehicle Counting Algorithm



# \* Real videos from Traffic monitoring system





\* Real videos from  
Traffic monitoring  
system

- \* Accurate Traffic Object Detection
- \* At different distances=> Different object scales
- \* CPU or GPU requirements: Computing resource
- \* Computing Architecture: Embedded or Centralized/ Arm<>Intel CPU platforms
- \* Processing Speed

# \* Experimental Application of AI to get Traffic Flow Data

- \* We test the same input traffic video for Tensorflow's object detection models: Faster R-CNN, R-FCN, SSD and YOLOv3.
- \* Those models are some of the best and most widely used object detection models out there right now.
- \* The tests are made with pre-trained models. If we do train the models with real on-site traffic monitoring videos, the results can be much better

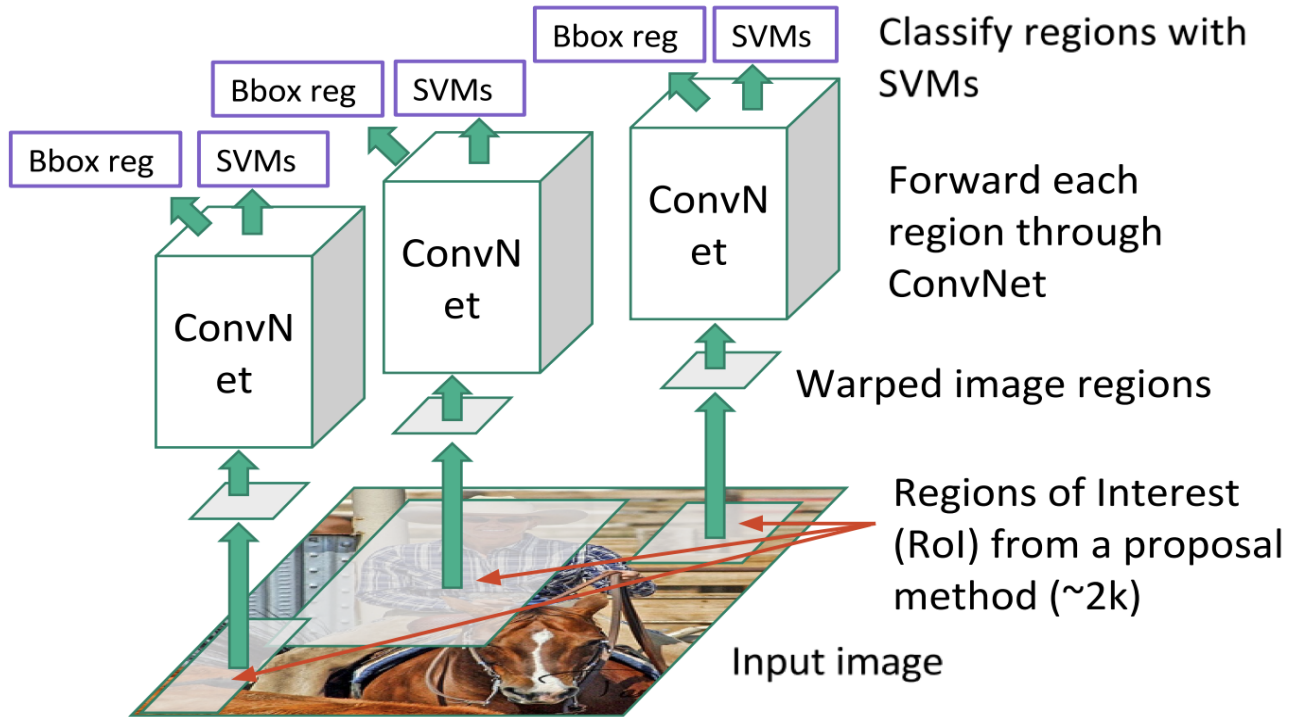
# \* AI - Deep Learning Tool Test for Traffic Flow Estimation

- \* Scan the input image for possible objects using an algorithm called Selective Search, generating ~2000 **region proposals**
- \* Run a convolutional neural net (**CNN**) on top of each of these region proposals
- \* Take the output of each **CNN** and feed it into a) an **SVM** to classify the region and b) a linear regressor to tighten the bounding box of the object, if such an object exists.

# \* **R-CNN Region-based Convolutional Neural Network**

# R-CNN

Linear Regression for bounding box offsets



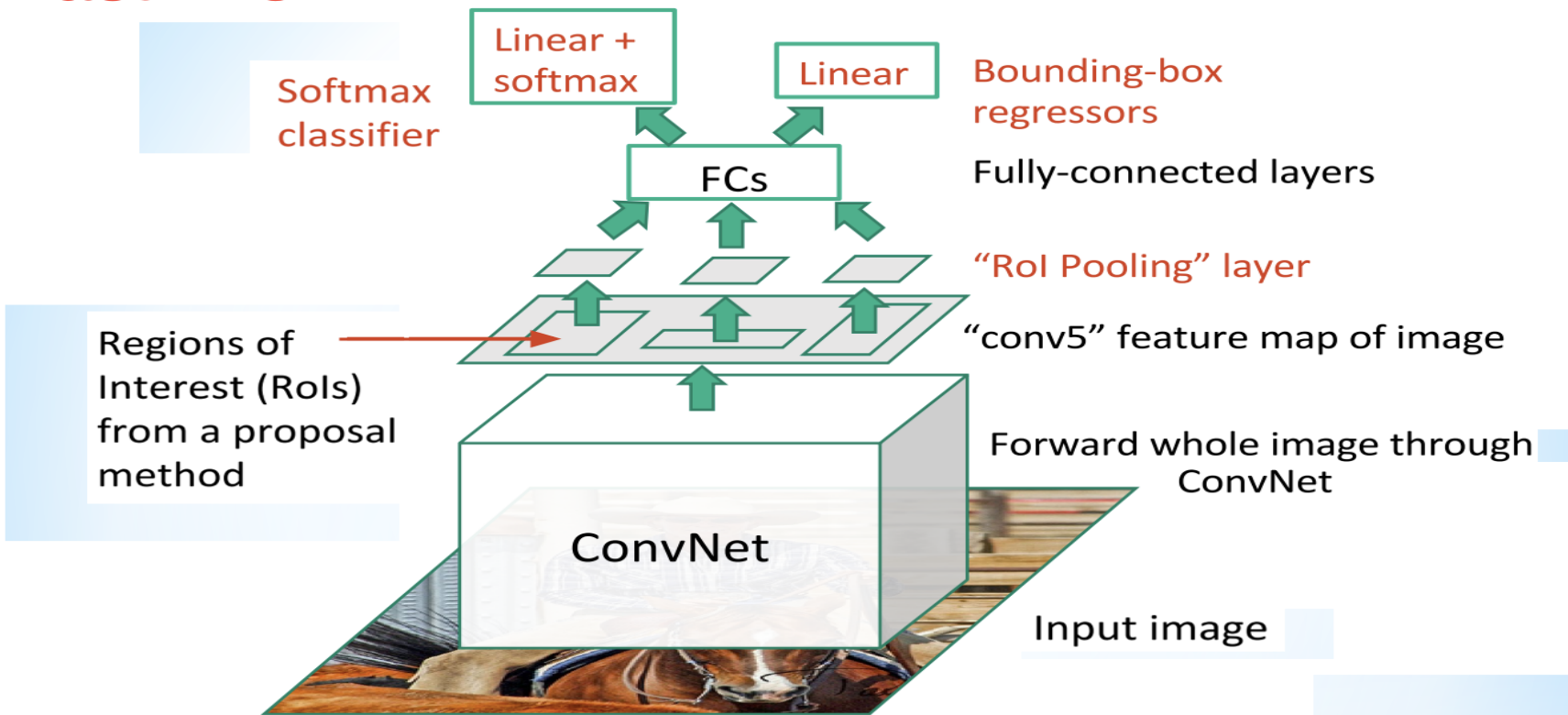
## \*R-CNN Region-based Architecture

Improved on its detection speed through two main augmentations

- \* Performing feature extraction over the image **before** proposing regions, thus only running one CNN over the entire image instead of 2000 CNN's over 2000 overlapping regions
- \* Replacing the SVM with a softmax layer, thus extending the neural network for predictions instead of creating a new model

\* **FAST R-CNN**

# Fast R-CNN

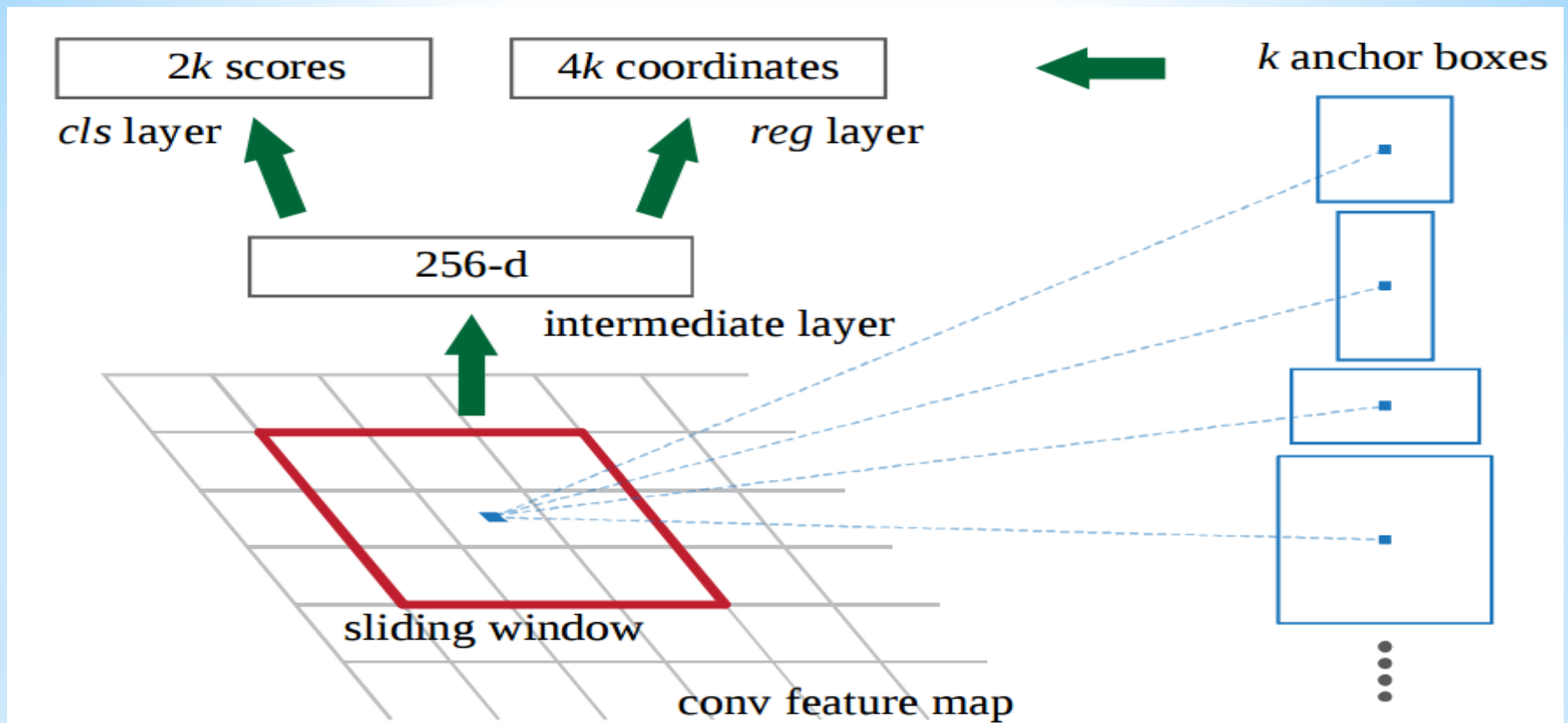


## \* FAST R-CNN Architecture

- \* The main insight of Faster R-CNN was to replace the slow selective search algorithm with a fast neural net. Specifically, it introduced the **region proposal network (RPN)**.
  - \* At the last layer of an initial CNN, a 3x3 sliding window moves across the feature map and maps it to a lower dimension (e.g. 256-d)
  - \* For each sliding-window location, it generates *multiple* possible regions based on  $k$  fixed-ratio **anchor boxes** (default bounding boxes)
  - \* Each region proposal consists of a) an “objectness” score for that region and b) 4 coordinates representing the bounding box of the region

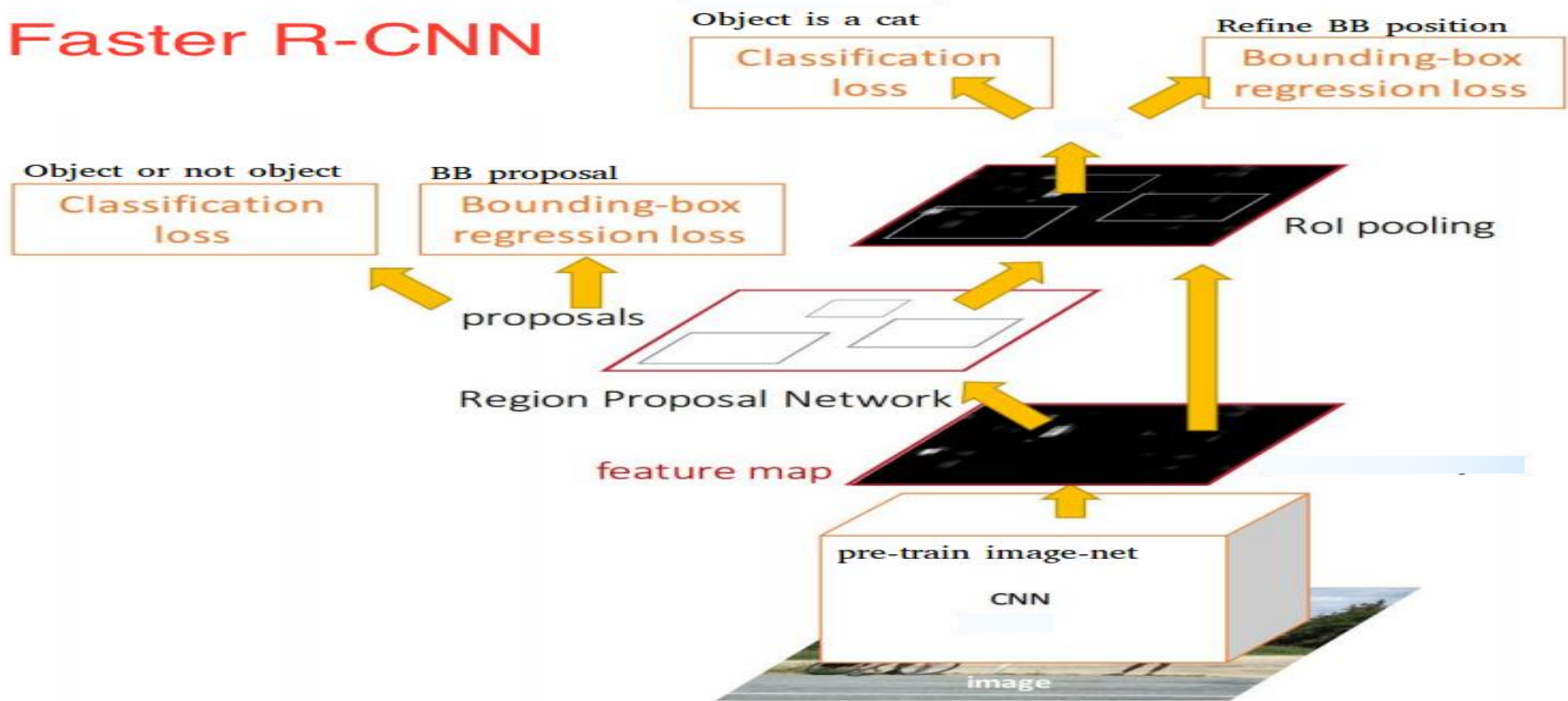
\* **Faster R-CNN = RPN +  
Fast R-CNN**





\* **Faster R-CNN: hypothesize object regions and then classify them**

# Faster R-CNN



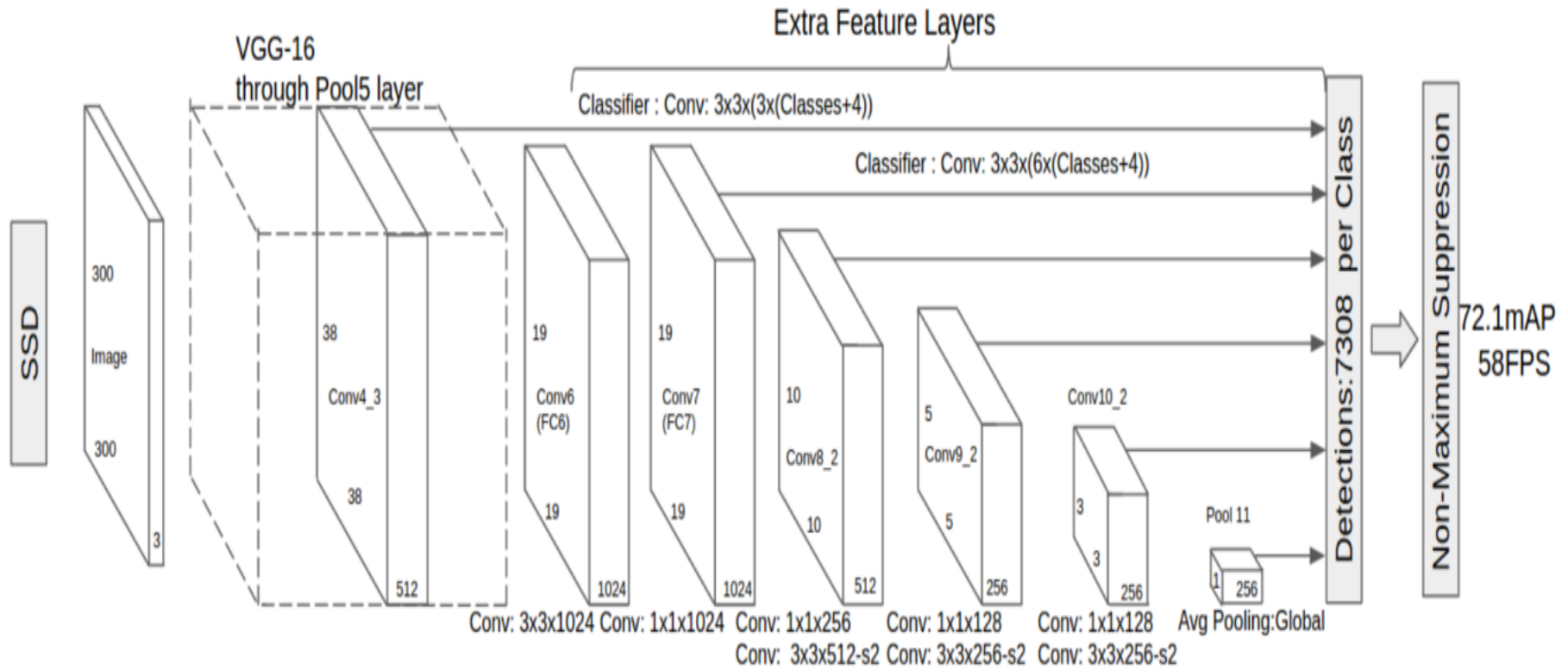
\* **Faster R-CNN  
Architecture**

- \* Faster R\_CNN performed region proposals and region classifications in two separate steps:
  - \* A region proposal network to generate regions of interest;
  - \* Either fully-connected layers or position-sensitive convolutional layers to classify those regions.
- \* SSD does the two in a “single shot,” simultaneously predicting the bounding box and the class as it processes the image.

## \* SSD: Single-Shot Detector

- \* Pass the image through a series of convolutional layers, yielding several sets of feature maps at different scales (e.g. 10x10, then 6x6, then 3x3, etc.)
- \* For each location in *each* of these feature maps, use a 3x3 convolutional filter to evaluate a small set of default bounding boxes. These default bounding boxes are essentially equivalent to Faster R-CNN's anchor boxes.
- \* For each box, simultaneously predict a) the bounding box offset and b) the class probabilities
- \* During training, match the ground truth box with these predicted boxes based on [IoU](#). The best predicted box will be labeled a "positive," along with all other boxes that have an IoU with the truth >0.5.

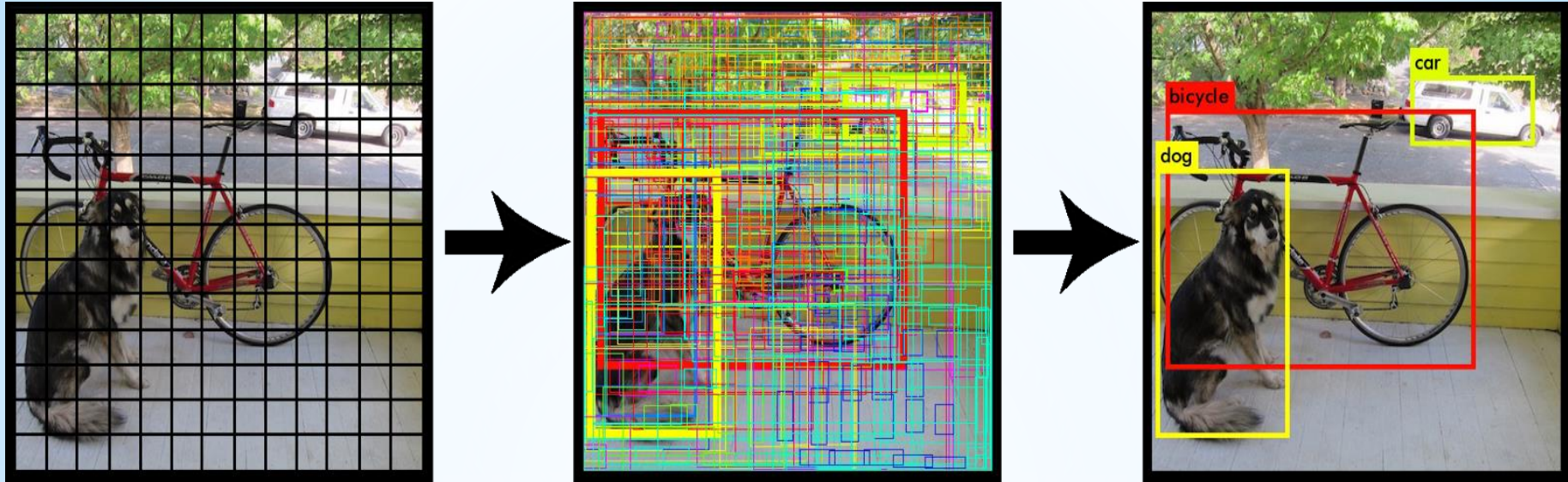
## \* SSD steps



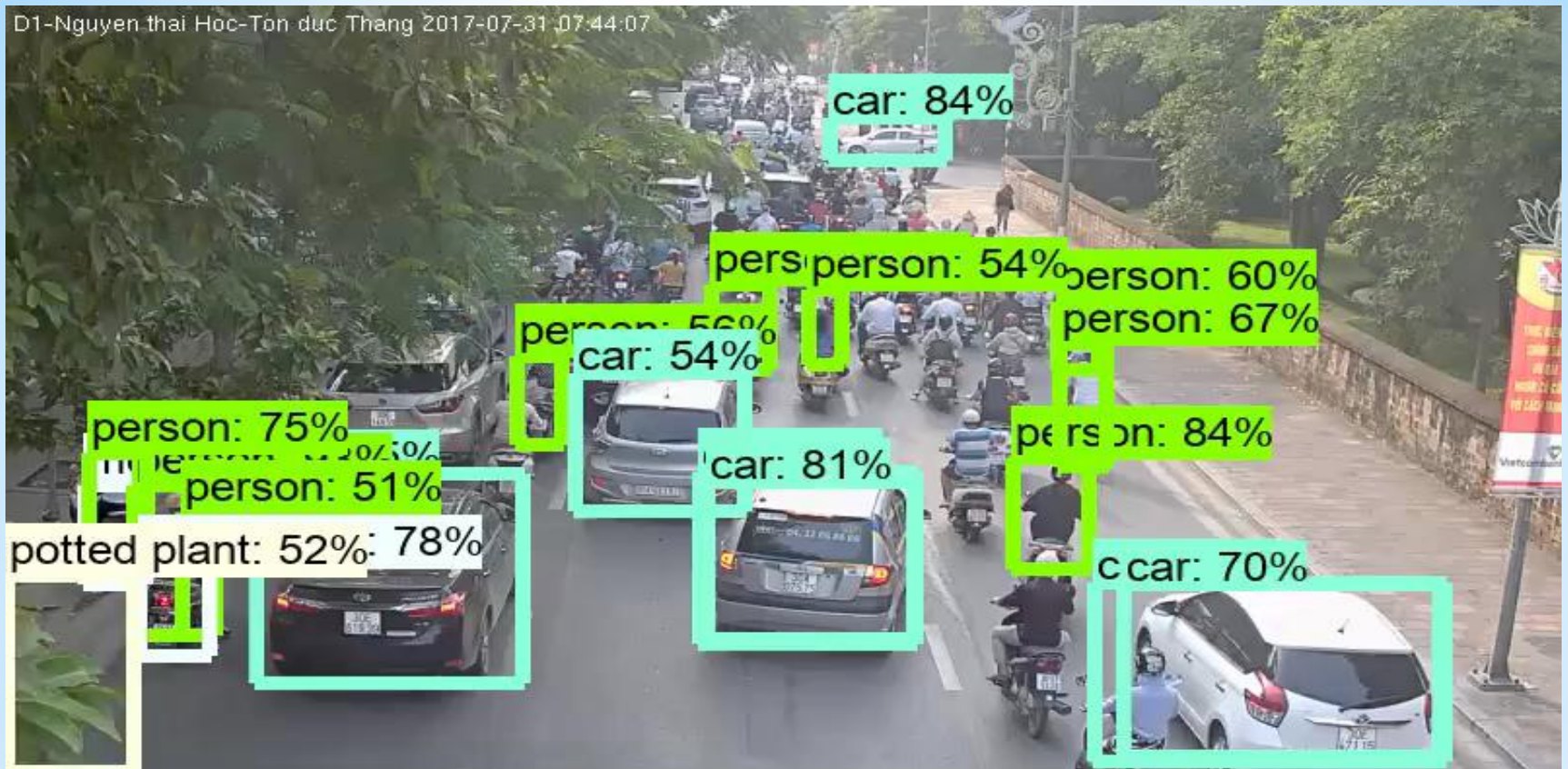
# \* SSD Architecture

- \* Prior detection systems repurpose classifiers or localizers to perform detection. They apply the model to an image at multiple locations and scales. High scoring regions of the image are considered detections.
- \* YOLO applies a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

## \* YOLO: Real-Time Object Detection

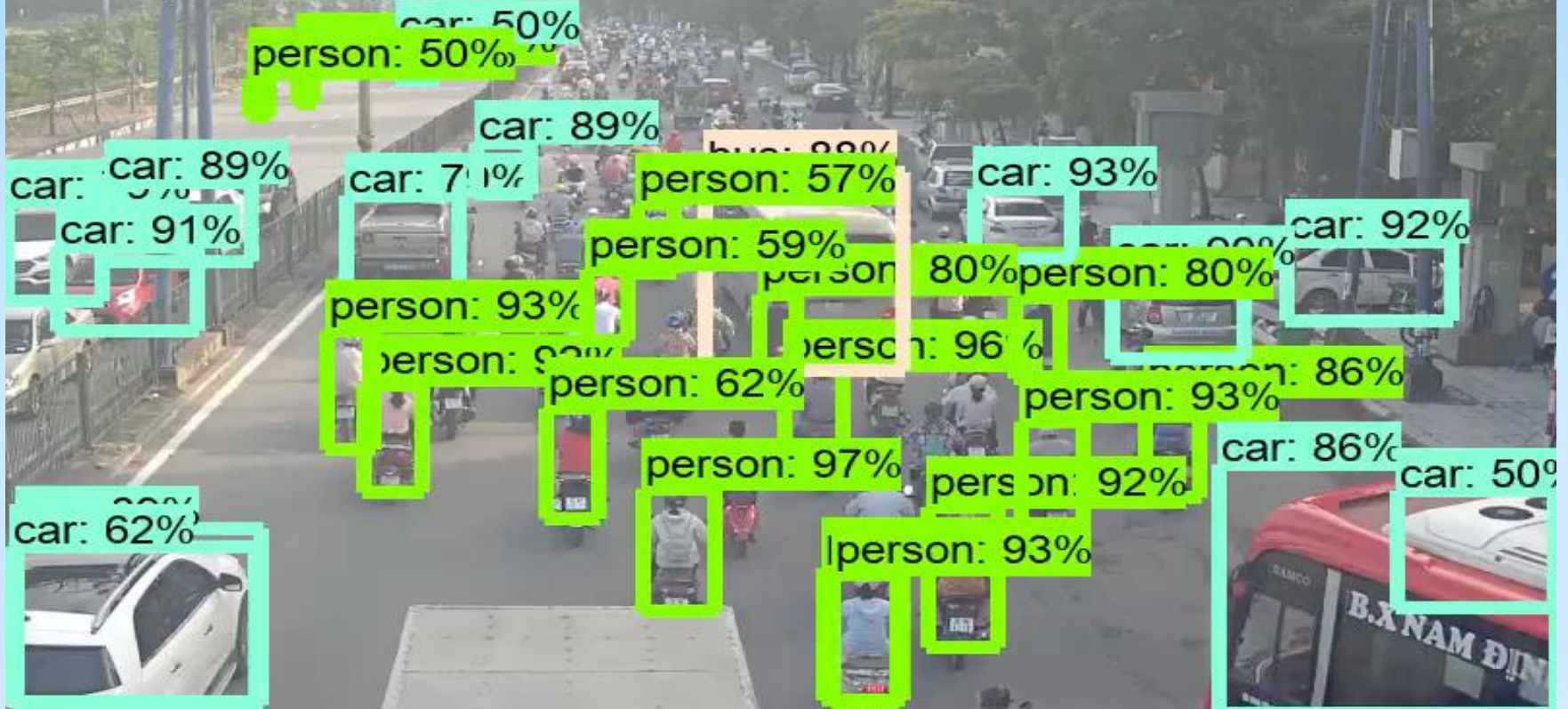


\*YOLO sample scheme

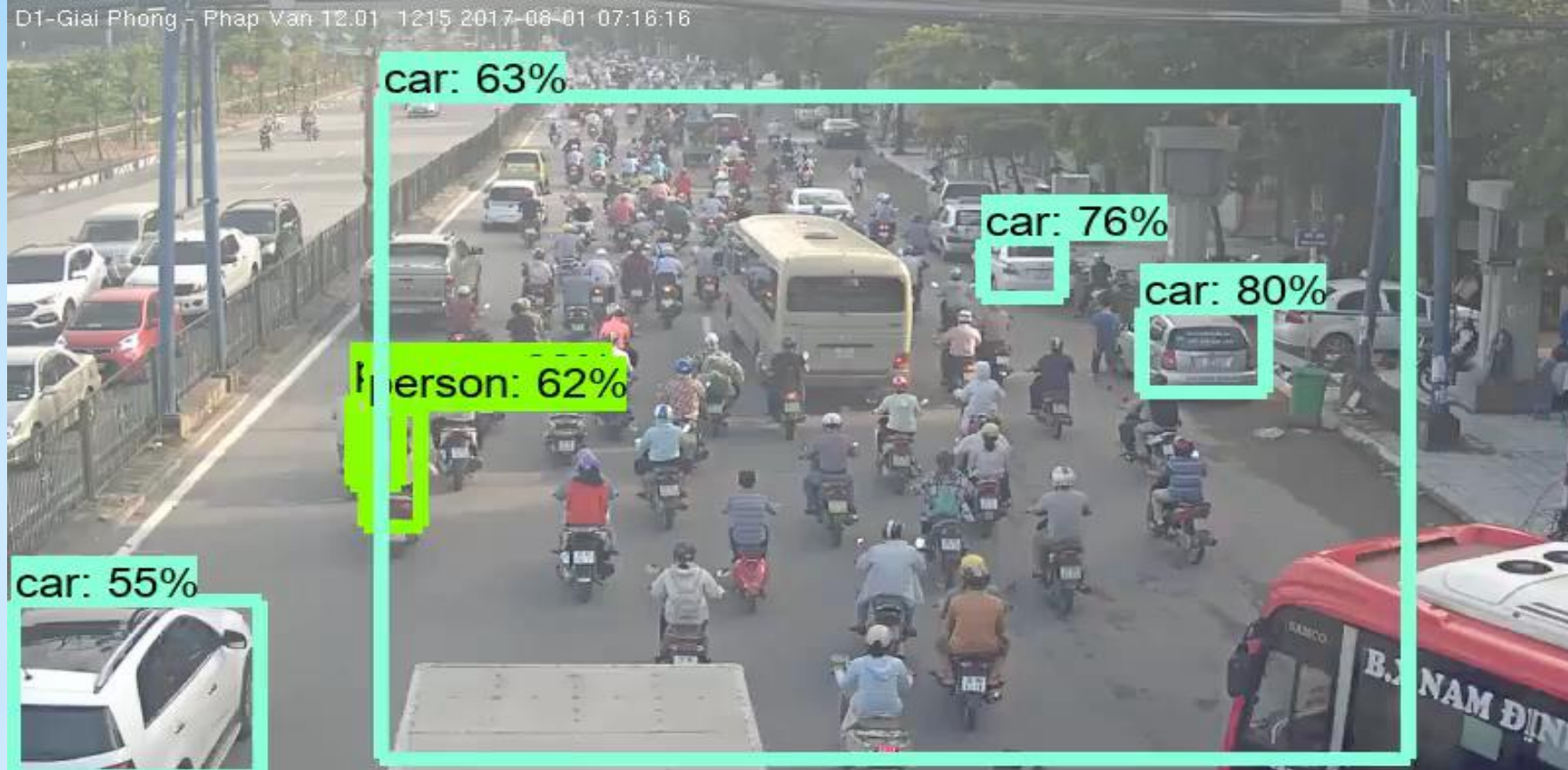


\*faster\_rcnn\_inception  
n\_v2\_coco  
2018\_01\_28

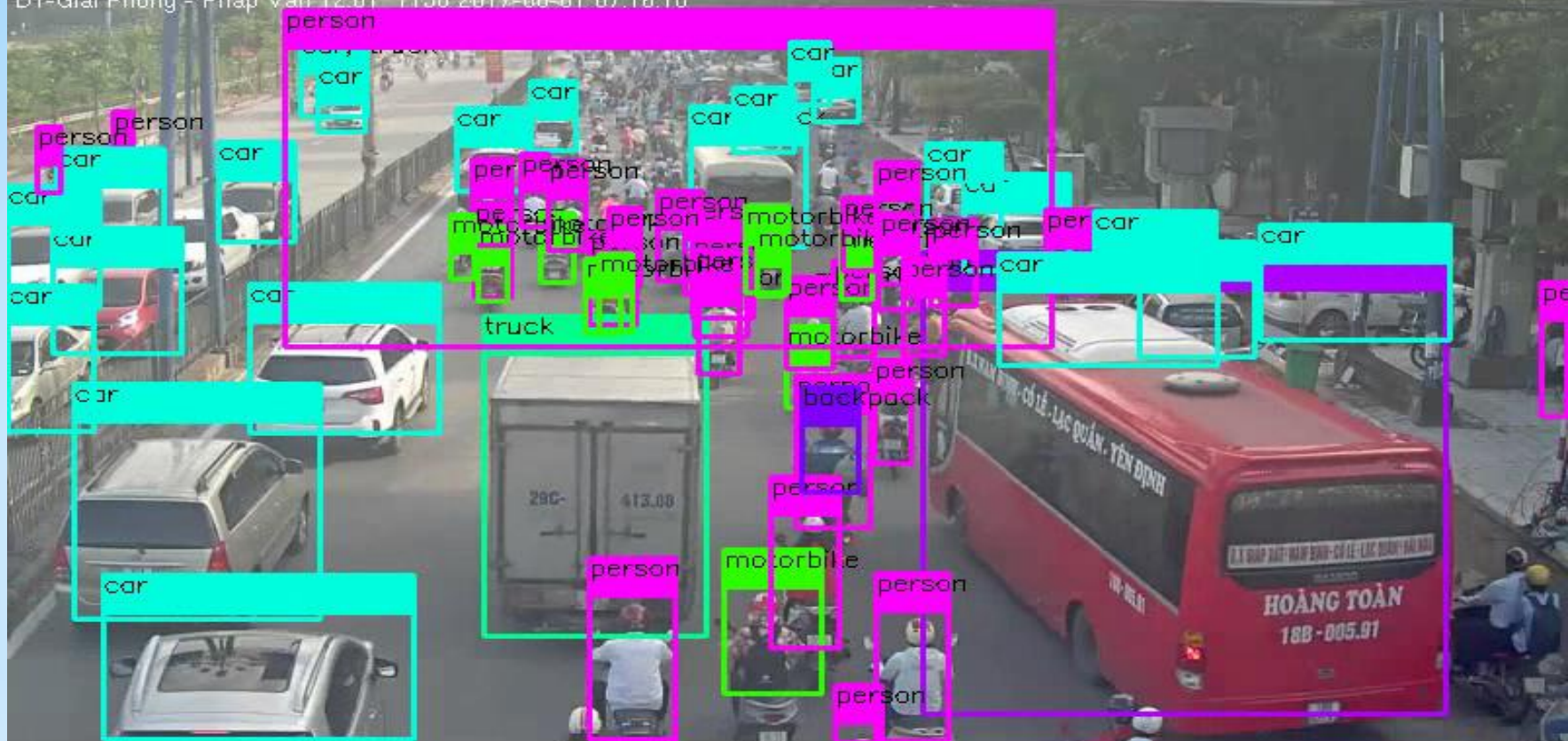




\*faster\_rcnn\_resnet50  
\_coco 2018\_01\_28



\* **ssd\_mobilenet\_v2\_co**  
**co 2018\_03\_29**



\*YOLOv3

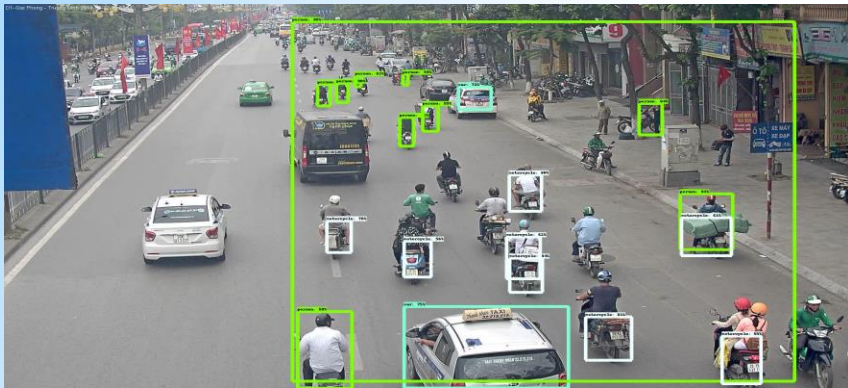
- \* Samsung Exynos5422 Cortex-A15 2Ghz and Cortex-A7 Octa core CPUs
- \* 2Gbyte LPDDR3 RAM PoP stacked
- \* Gigabit Ethernet port, USB 2.0 Host
- \* Linux server OS
- \* Operating Temperature: up to 60°C, suitable for road-side installation in Vietnam



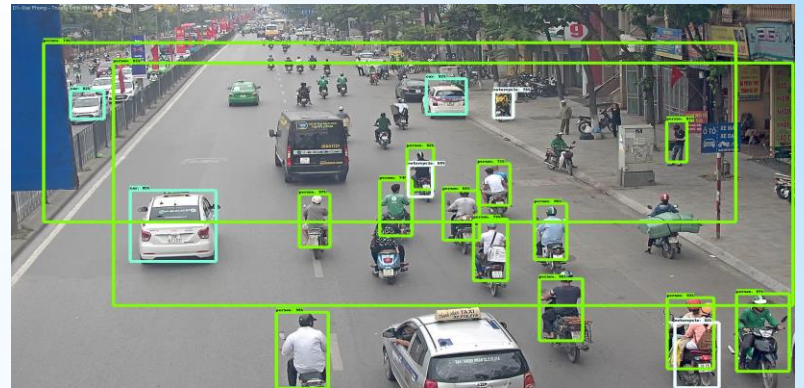
# \* Self-made Embedded Platform

Model name	1 frame proc.time	maxcpu
ssd_mobilenet_v2_coco	7-8.5	250%
faster_rcnn_inception_v2_coco	18.5-19.5	740%
faster_rcnn_resnet50_coco	42-43	700%
faster_rcnn_resnet101_coco	55-56	780%
yolov1	7.2	380%

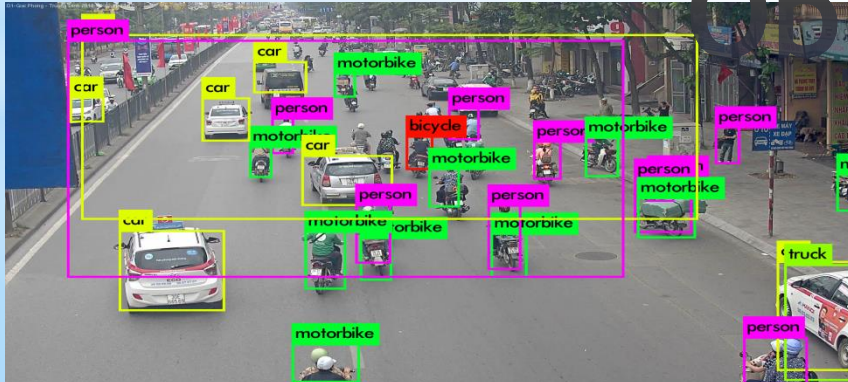
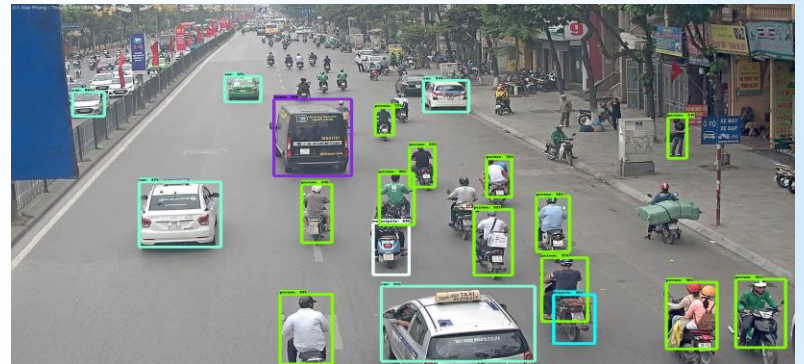
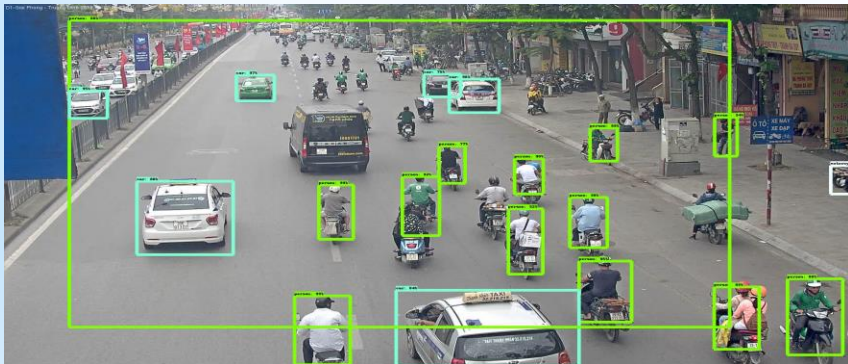
\* Processing  
Performance On  
Embedded Platform



ssd\_mobilenet 8s 250%  
 faster\_rcnn\_resnet50 42s 700%



faster\_rcnn\_inception 19s 740%  
 faster\_rcnn\_resnet101 55s 780%



# \* Object Detection Performances

Yolov1 7s 380%

- \* Some AI object detection models are installed and tested in specific Vietnamese made embedded computing platform.
- \* Best object detection performance/computing resources ratio is selection criteria
- \* Automatic Flow Data Extraction from Camera Monitoring System is ready for Intelligent Traffic Management with high efficiency
- \* Development of Yoyo-embedded AI model will be focused in future plan

## \* Conclusion