Joint Object Detection and Viewpoint Estimation using CNN features

VII LSI PhD Workshop

Carlos Guindel
cguindel@ing.uc3m.es
Intelligent Systems Laboratory · Universidad Carlos III de Madrid

Leganés · 19 June 2017

Joint Object Detection and Viewpoint Estimation using CNN features

C. Guindel, D. Martín, and J. M. Armingol, “Joint Object Detection and Viewpoint Estimation using CNN features,” in IEEE International Conference on Vehicular Electronics and Safety (ICVES), 2017.

Outline

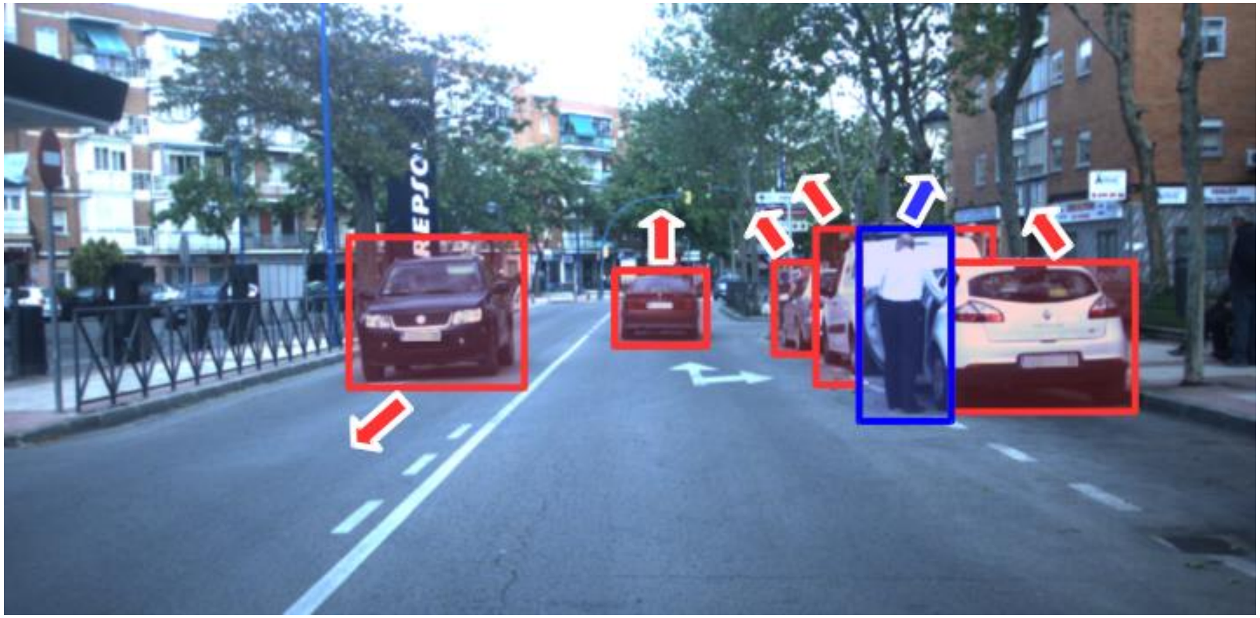
- Introduction
- Object detection
- Viewpoint estimation
- Results
- Conclusion

Outline

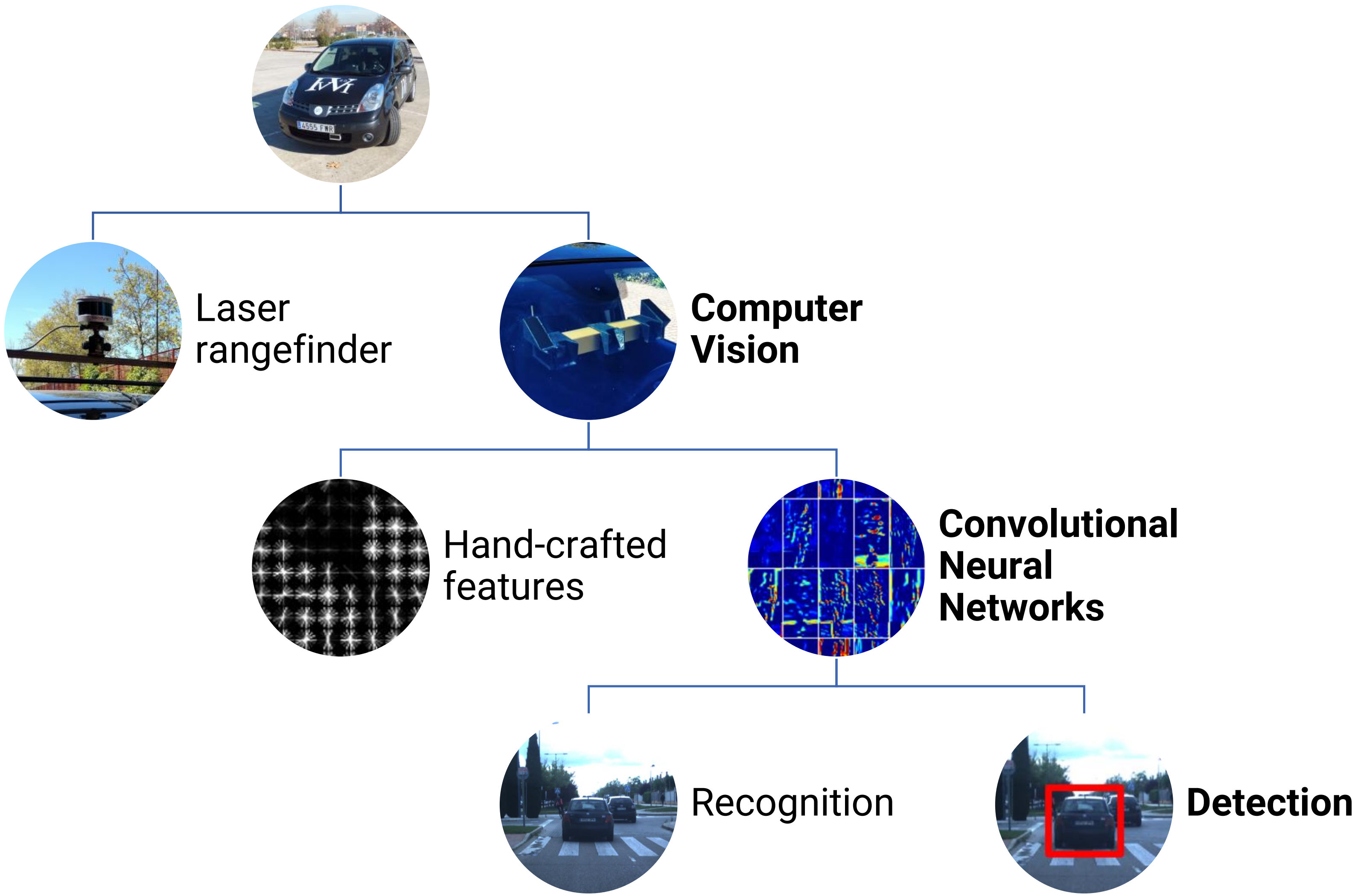
- **Introduction**
- Object detection
- Viewpoint estimation
- Results
- Conclusion

Situational awareness for vehicles

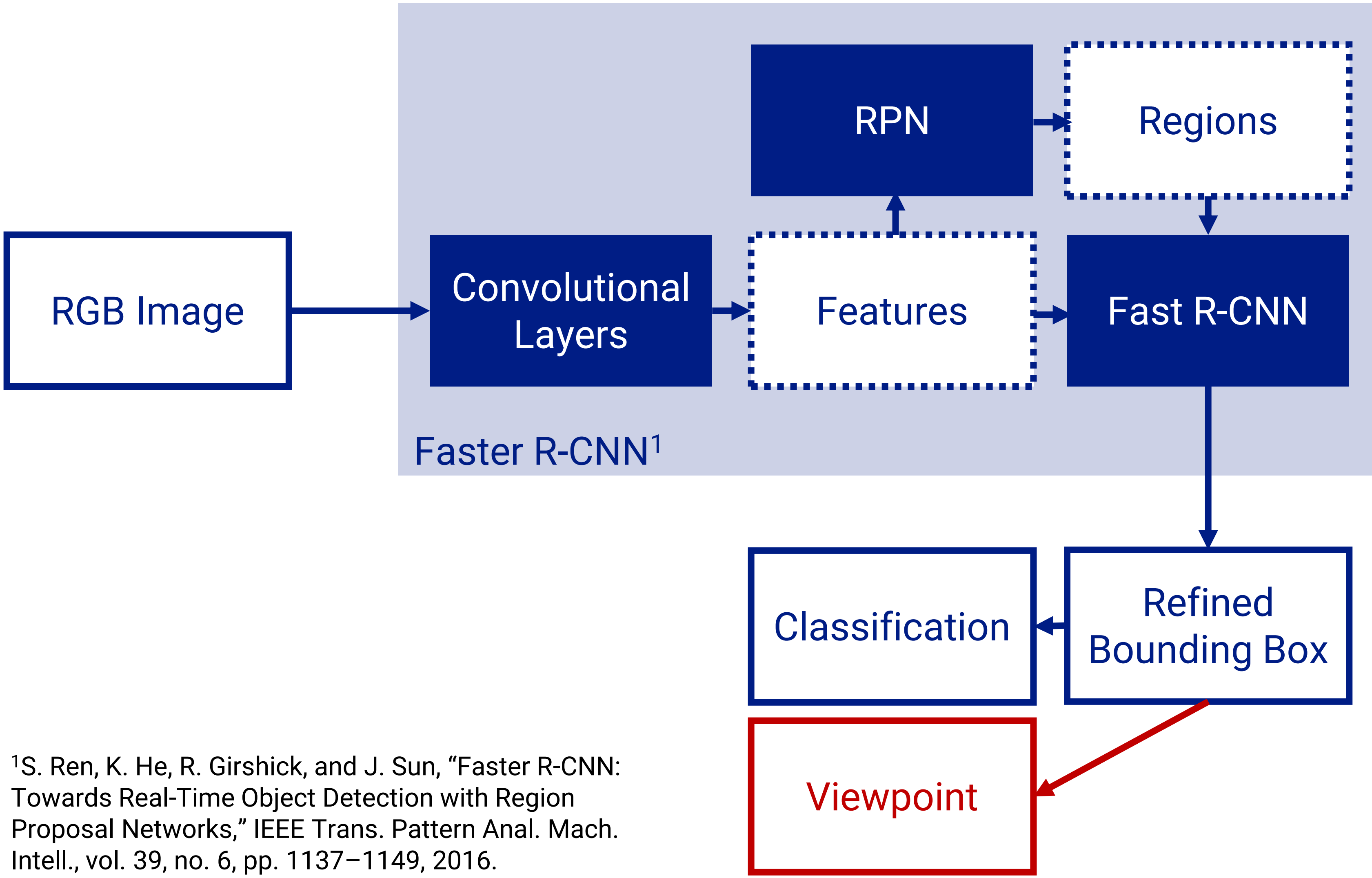
- Advanced Driver Assistance Systems (ADAS) and autonomous vehicles rely on a trustable on-board **obstacle detection** module.
- A precise **classification** of the obstacles enables accurate predictions of future traffic situations, including those involving VRU.
- Another significant cue that can be used to anticipate future events is the **orientation** of the objects moving on the ground plane.



On-board object detection



Proposal overview



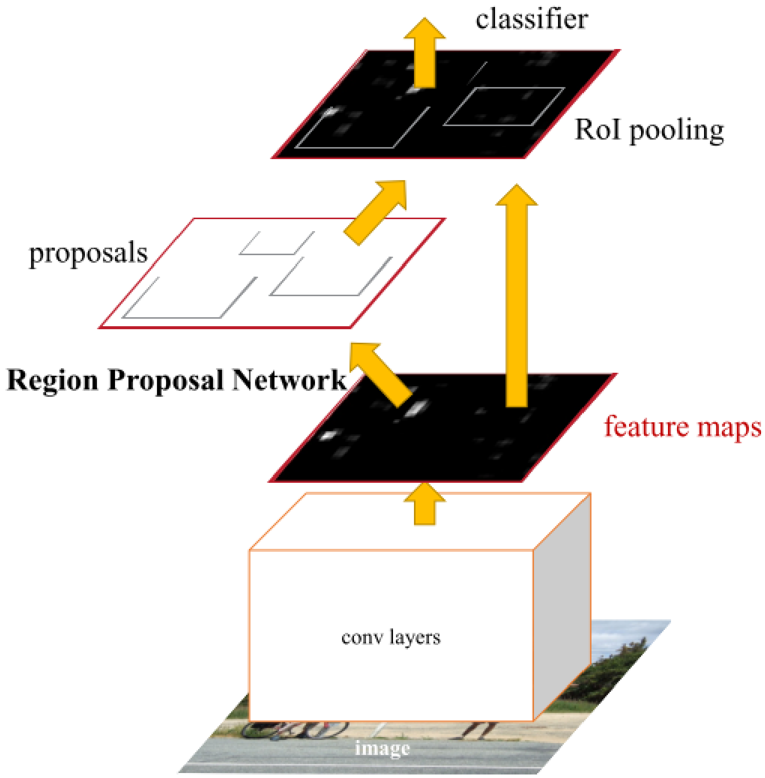
[1] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, 2016.

Outline

- Introduction
- **Object detection**
- Viewpoint estimation
- Results
- Conclusion

Object detection

- Traffic environments
  - Diversity of agents
  - Unstructured environment
- Faster R-CNN
  - End-to-end feature learning
  - Highly efficient
  - No prior constraints about the location of objects in the image
  - Meant for more than 21 classes



S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, 2016.

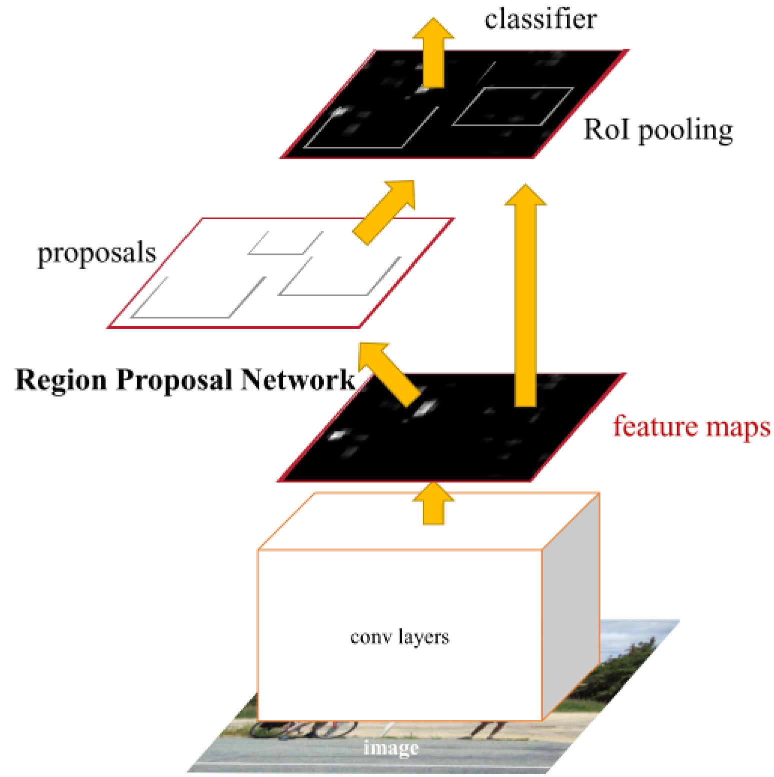
Faster R-CNN framework

Parameters are learned through a multi-task loss

Conv. features in these regions are pooled for **classification**

A **RPN** generates proposals wrt. a fixed set of anchors

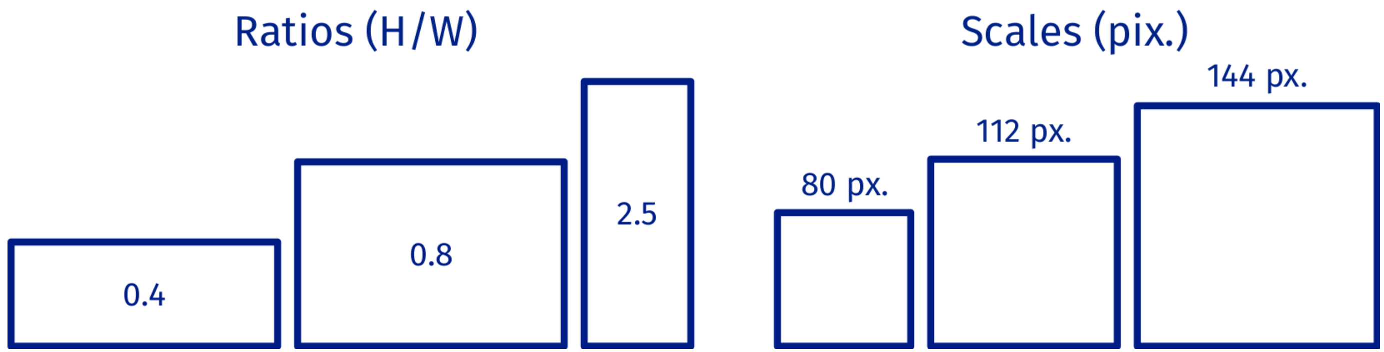
Convolutional features computed **only once** per image



S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, 2016.

Fine-tuning for traffic environments

- Optimized anchors



- Management of class imbalance
  - Information gain multinomial logistic loss for the **class** inference

$$L = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} H_{l_n,k} \log(\hat{p}_{n,k})$$

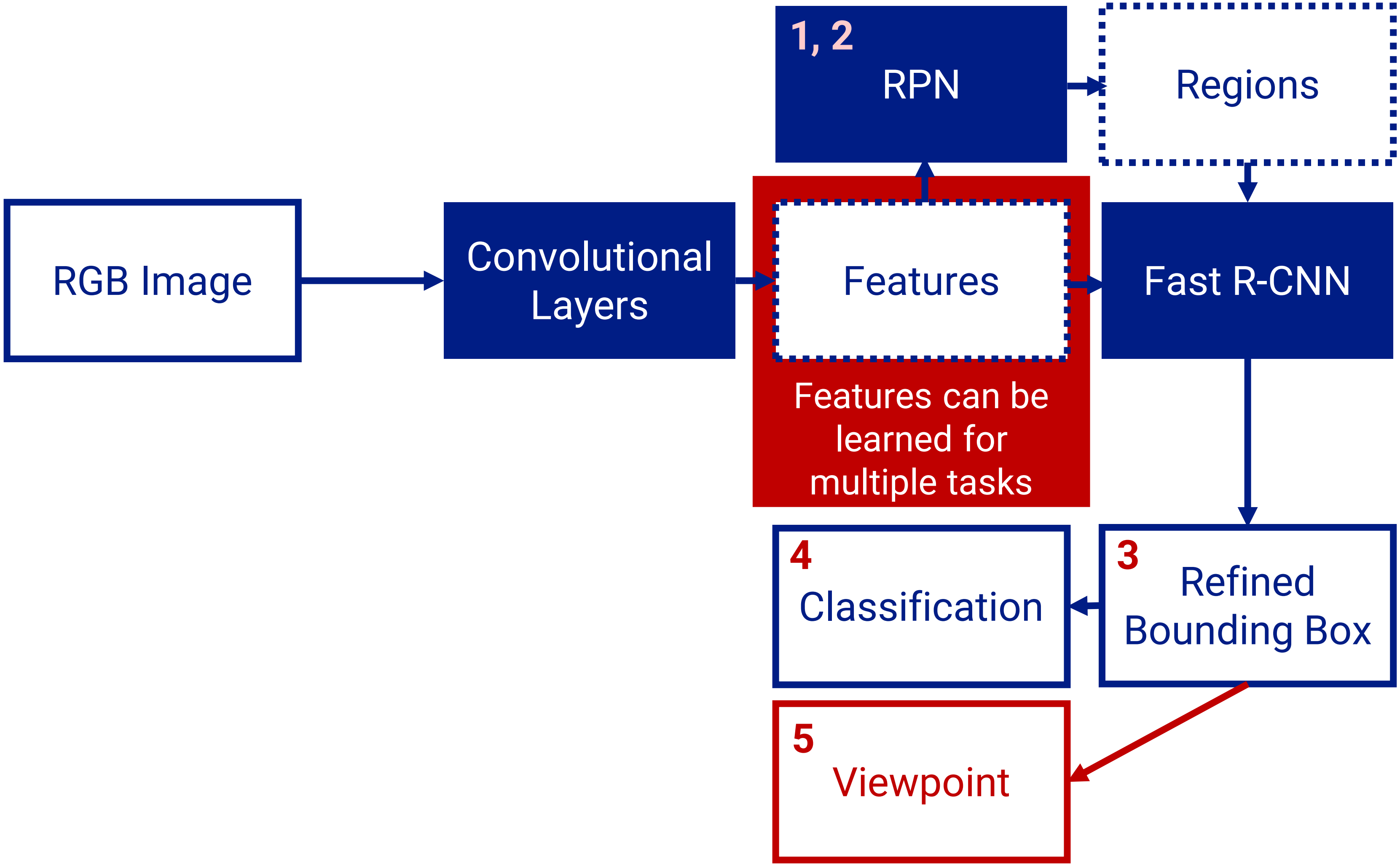
Infogain matrix

$$\begin{pmatrix} H_{0,0} & \cdots & 0 \\ 0 & & \\ \vdots & \ddots & \vdots \\ & & 0 \\ 0 & \cdots & H_{K,K} \end{pmatrix}$$

Outline

- Introduction
- Object detection
- **Viewpoint estimation**
- Results
- Conclusion

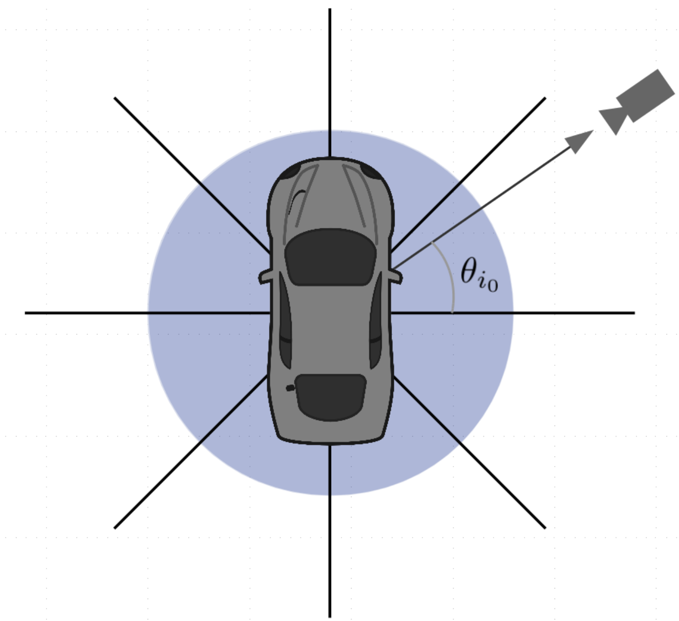
Viewpoint estimation



Discrete viewpoint inference

N_b **angle bins** $\Theta_i \dots \Theta_{N_b}$

$N_b = 8$



Training: $\theta_{i_0} \rightarrow \Theta_i$

$$\Theta_i = \left\{ \theta \in [0, 2\pi) \;\middle|\; \frac{2\pi}{N_b} \cdot i \leq \theta < \frac{2\pi}{N_b} \cdot (i+1) \right\}$$

Inference output: $\mathrm{r} \in \Delta^{N_b - 1}$

$$\Delta^N = \left\{ x \in \mathbb{R}^{N+1} \;\middle|\; \sum_{i=1}^{N+1} x_i = 1 \;\wedge\; \forall i\colon x_i \geq 0 \right\}$$

r

$$i^* = \arg\max_i (x_i)$$

Elements of r

Final estimation: $\Theta_{i^*} \rightarrow \hat{\theta}$

$$\hat{\theta} = \frac{\pi(2i^* + 1)}{N_b}$$

Joint detection and viewpoint estimation

CNN outputs

RPN

For each anchor:

- Objectness

  $a \in \{0,1\}$

- Predicted bounding box

  $b = (b_x, b_y, b_w, b_h)$

Fast R-CNN

For each proposal:

- Class

  $p = (p_0, \ldots, p_K)$

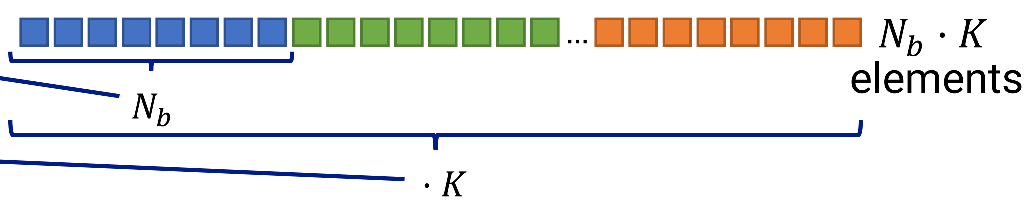
- Bounding box refinement

  $t^k = (t^k_x, t^k_y, t^k_w, t^k_h)$ for $k = 0, \ldots, K$

Per class

- Viewpoint

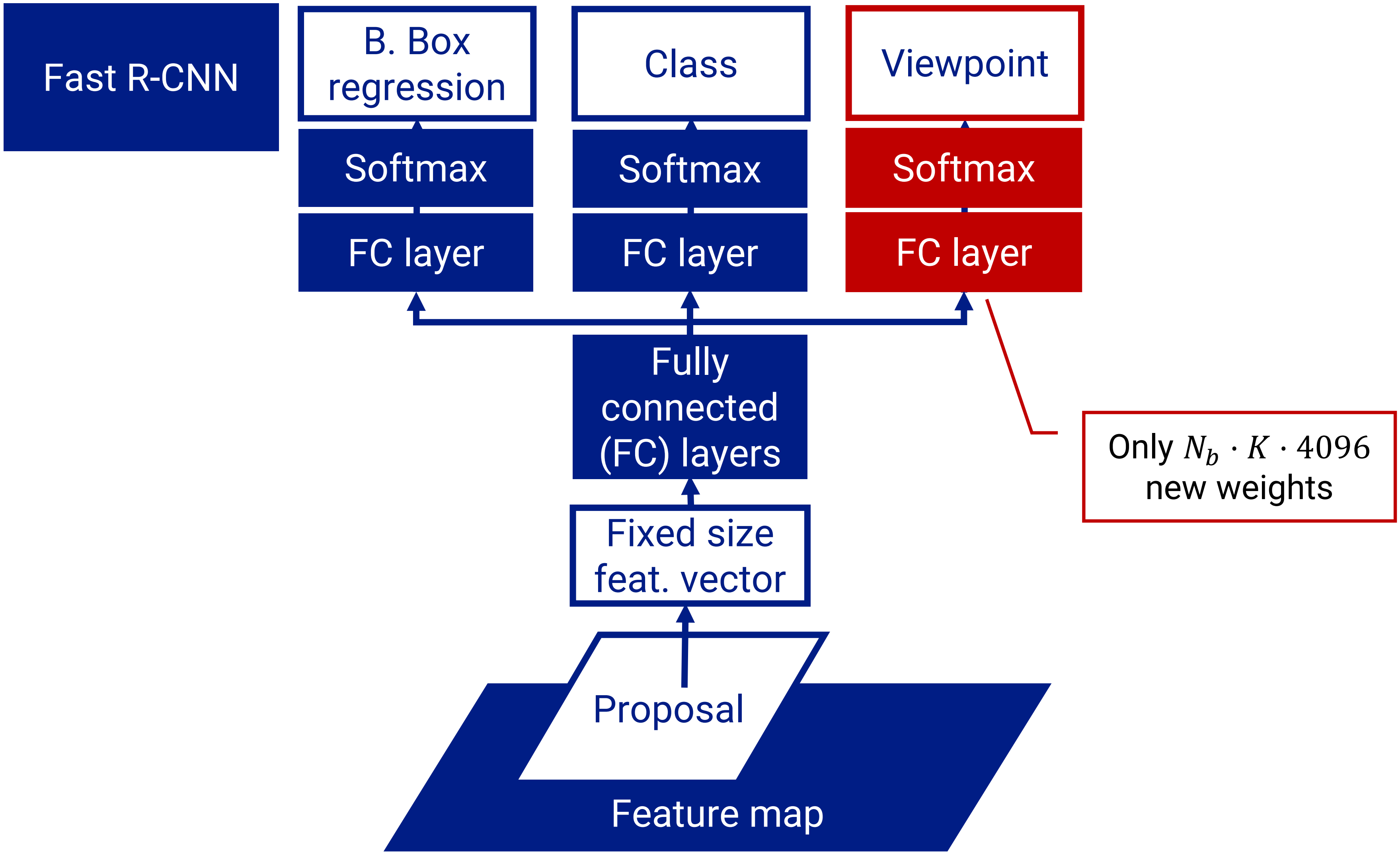
  $r^k = (r^k_0, \ldots, r^k_{N_b})$ for $k = 0, \ldots, K$



Number of angle bins

Number of classes

Joint detection and viewpoint estimation



Loss function and training

- **Approximate joint** training strategy
- Unweighted muli-task loss with **five** components

Logistic loss
for RPN objectness

Smooth-L1 loss
for RPN b.box regression

$$L = \frac{1}{N_{B_1}} \sum_{j \in B_1} L_{cls}(a_j, u_j) + \frac{1}{N_a} \sum_{j \in B_1} u_j L_{loc}(b_j, b_j^*) +$$
$$\frac{1}{N_{B_2}} \sum_{i \in B_2} L_{inf}(p_i, v_i) + \sum_{i \in B_2} [u \geq 1] \, L_{loc}(t_i^v, t_i^{v*}) +$$
$$\frac{1}{N_{B_2}} \sum_{i \in B_2} [u \geq 1] \, L_{cls}(r_i^u, \Theta_i)$$

Smooth-L1 loss
for b.box regression

Infogain loss
for class

$$L_{inf}(p_i v_i) = \sum_{n=1}^{N} H_{v_i,k} \log(p_{i,k})$$

↓ frequent → ↑ $H_{v_i,k}$

Logistic loss
for viewpoint estimation

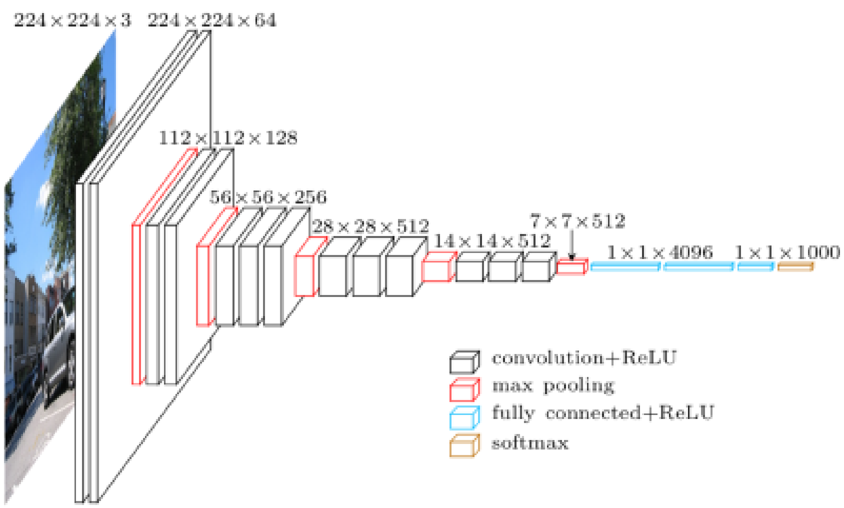
Only the N_b elements of the ground-truth class

Outline

- Introduction
- Object detection
- Viewpoint estimation
- **Results**
- Conclusion

Experiments

- On the KITTI Vision Benchmark Suite – Object detection set
- **Training parameters:**
  - **Scale**: 500 px. in height
  - 50k iter. $l_r = 0.001$ **+** 50k iter. $l_r = 0.0001$ **+** 50k iter. $l_r = 0.00001$
  - **VGG16** architecture, initialized with ImageNet weights.



KITTI: A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 3354–3361.

VGG16 Image: https://blog.heuritech.com/2016/02/29/a-brief-report-of-the-heuritech-deep-learning-meetup-5/

Experiments

- $N_b = 8$ (resolution: $\pi/4$ rad)
- Infogain matrix values:

$$H_{k,k} = 2 \cdot \left(\frac{f_{min}}{f_k}\right)^{\frac{1}{8}}$$

Number of ocurrences of the less frequent class

Number of instances of class k

Evaluation criteria

- Average precision
- Average orientation similarity (performance of **detection + orientation**)

$$AOS = \frac{1}{11} \sum_{r \in \{0, 0.1, .., 1\}} \max_{\tilde{r}:\tilde{r} \geq r} s(\tilde{r}) \qquad s(r) = \frac{1}{|\mathcal{D}(r)|} \sum_{i \in \mathcal{D}(r)} \frac{1 + \cos \Delta_\theta^{(i)}}{2} \delta_i$$

- Minimum overlaps established by the KITTI benchmark

KITTI submission

The KITTI Vision Benchmark Suite
A project of Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago

Karlsruhe Institute of Technology

home | setup | stereo | flow | scene flow | odometry | object | tracking | road | semantics | raw data | submit results | jobs

Andreas Geiger (MPI Tübingen) | Philip Lenz (KIT) | Christoph Stiller (KIT) | Raquel Urtasun (University of Toronto)

Method

Joint Object Detection and Viewpoint Estimation using CNN features [FRCNN+Or]

Submitted on 7 Jun. 2017 10:58 by
Carlos Guindel (Universidad Carlos III de Madrid)

Running time: 0.1 s
Environment: GPU @ 1.5 Ghz (Python + C/C++)

Method Description:
We enhance the well-known Faster R-CNN method by adding viewpoint inference capabilities. Convolutional features are computed and shared for use in the tasks of region proposal, classification, and orientation estimation, for efficiency reasons.

Parameters:
Scale=500, 300 RPN proposals, 3 anchor scales and 3 anchor ratios, \N_b=8, \l_r=0.001, 150k iterations. Running time given for a NVIDIA Titan X Pascal.

Latex Bibtex:
```
@inproceedings{GuindelICVES,
author = {Guindel, Carlos and Martin, David and Armingol, Jose
M.},
booktitle = {IEEE International Conference on Vehicular
Electronics and Safety (ICVES)},
title = {{Joint Object Detection and Viewpoint Estimation
using CNN features}},
year = {2017}
}
```

Results (KITTI submission: FRCNN+Or)

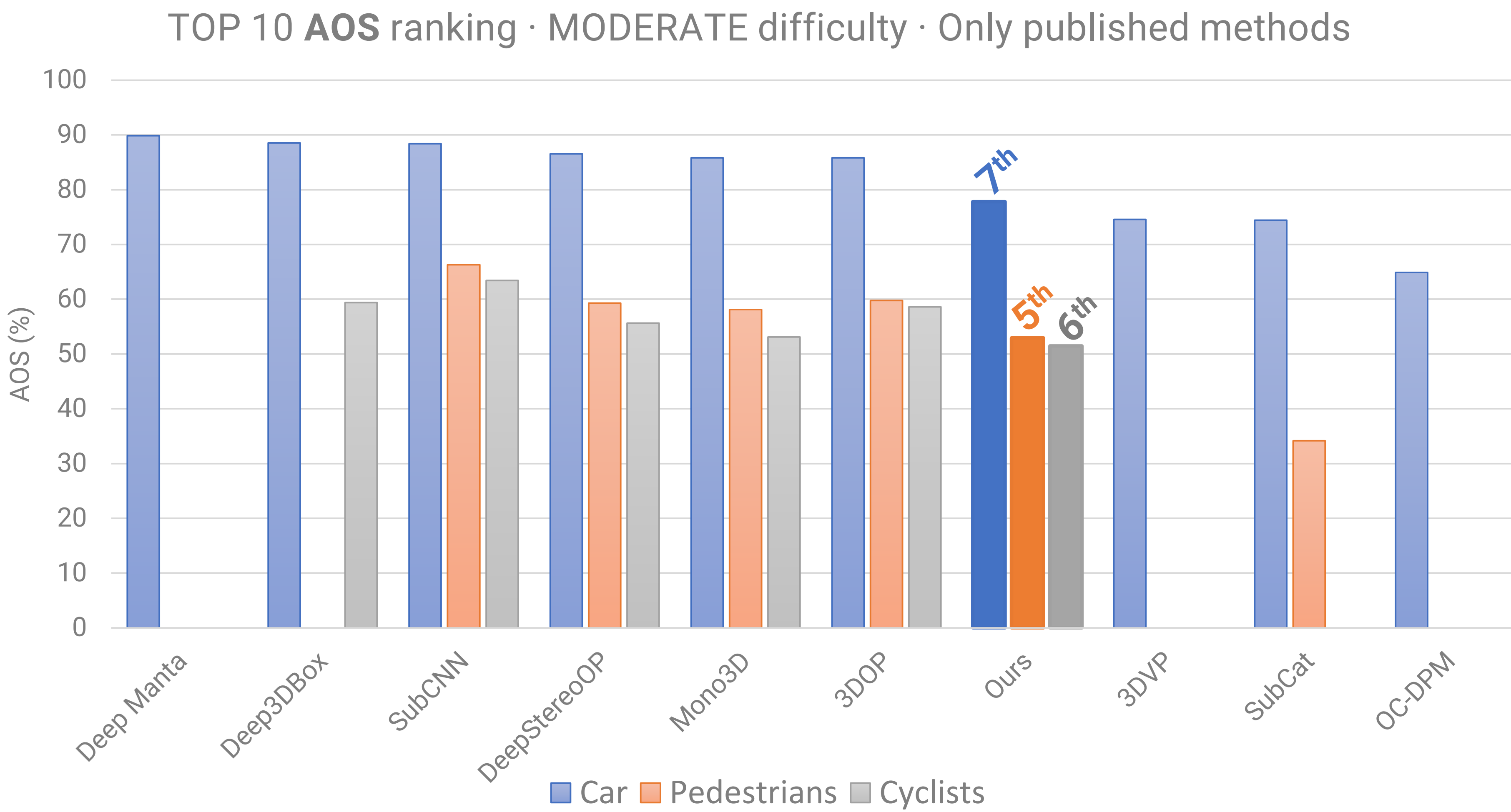
Detection (AP as %)	Easy	Moderate	Hard
Car	89.60	78.59	68.69
Pedestrian	72.21	56.99	53.72
Cyclist	68.81	55.80	50.52
mAP	**76.87**	**63.79**	**57,64**
SubCNN	*84.52*	*77.14*	*64.44*

- Slightly different (generally better) than the ones in the paper:
  - Used the whole KITTI training set
  - Trained only with Car, Pedestrian and Cyclist
  - Non-fixed weights (and bias) at the first convolutional layers

Det + Or (AOS as %)	Easy	Moderate	Hard
Car	88.93	77.8	67.87
Pedestrian	67.92	52.96	49.61
Cyclist	64.90	51.47	46.48
mAOS	**73.92**	**60.74**	**54,65**
SubCNN	*80.37*	*72.85*	*65.45*

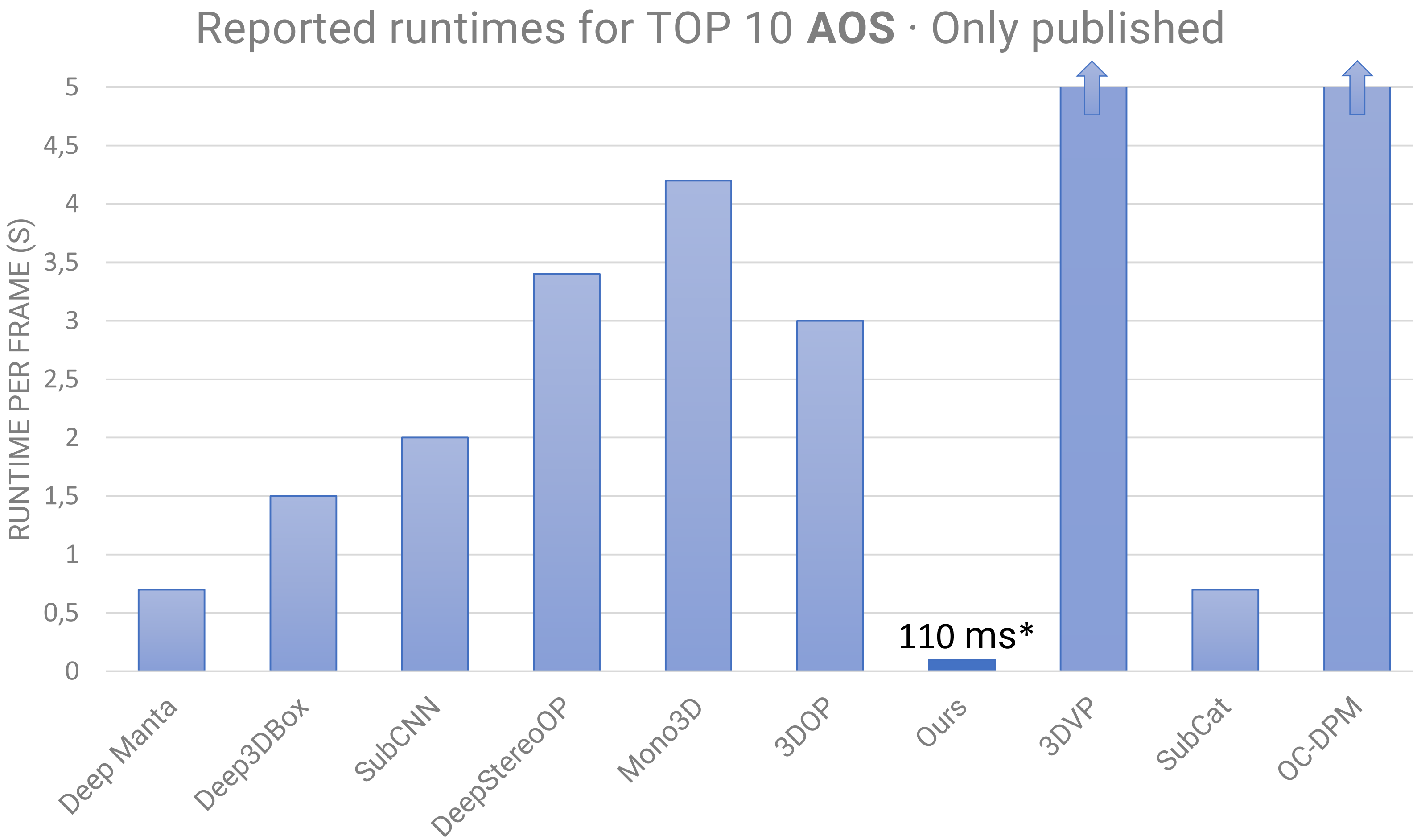
SubCNN: Y. Xiang, W. Choi, Y. Lin and S. Savarese, “Subcategory-aware Convolutional Neural Networks for Object Proposals and Detection,” in IEEE Winter Conference on Applications of Computer Vision (WACV), 2017, pp. 924-933.

Comparison with published KITTI submissions



Global rankings (including unpublished methods): Car: 11st · Pedestrian: 9th · Cyclist: 10th

Comparison with published KITTI submissions



*Average running time using an implementation based on py-faster-rcnn (Python & Caffe) and a NVIDIA **Titan Xp** donated by NVIDIA Corporation

Outline

- Introduction
- Object detection
- Viewpoint estimation
- Results
- **Conclusion**

Conclusion

- Monocular approach for object detection focused on traffic environments.
- Based on a state-of-the-art CNN and adding viewpoint inference.
- Results comparable with non-real-time sophisticated approaches.
- Orientation is a step towards a complete scene understanding.

Future work

- Fine-grained orientation inference using the **cross-entropy logistic loss**

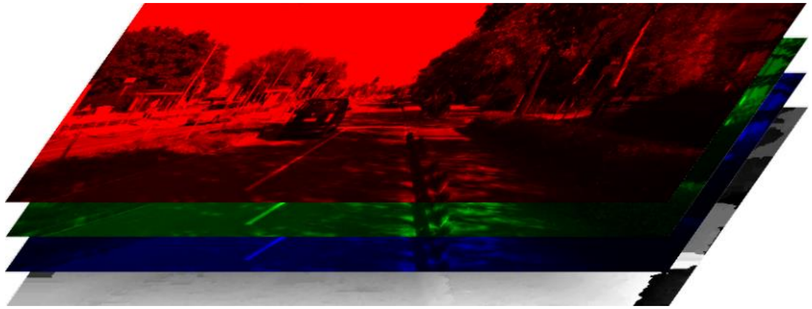
$$L = -\frac{1}{n}\sum_{n=1}^{n}[p_n \log \hat{p}_n + (1 - p_n)\log(1 - \hat{p}_n)]$$

- Improvements:
  - Network architecture
  - Methods to overcome the fixed-size receptive field

Other Developments

Additional data for the CNN

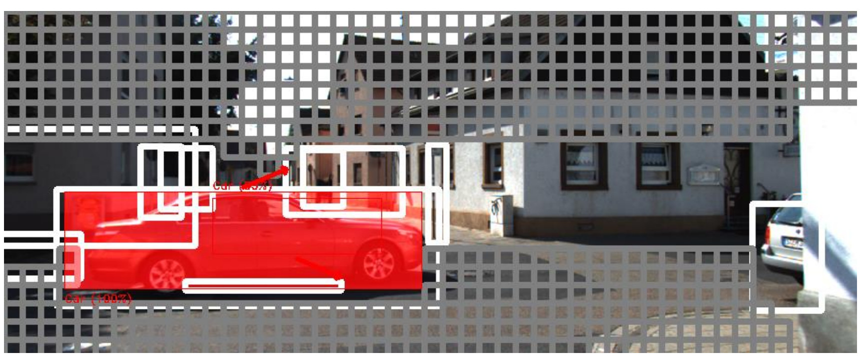
- Stereo. Using the disparity map as a fourth channel

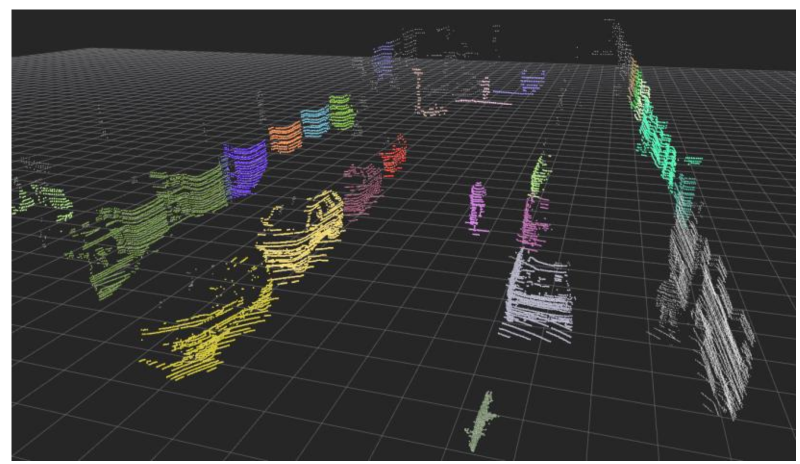


Input	Easy	Moderate	Hard
RGB	79.98	66.46	57.53
RGB+SGM	+0.30	+0.01	+2.52
RGB+DispNet	**+1.03**	**+1.14**	**+3.57**

C. Guindel, D. Martín, and J. M. Armingol, “Stereo Vision-Based Convolutional Networks for Object Detection in Driving Environments,” in EUROCAST 2017 - Extended Abstracts, 2017, pp. 288–289.

- Laser. Classifying object proposals coming from the Velodyne

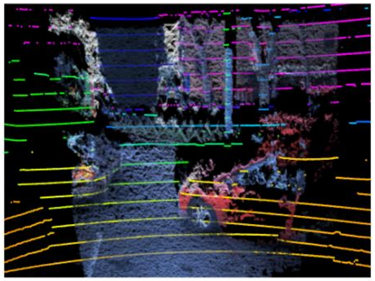




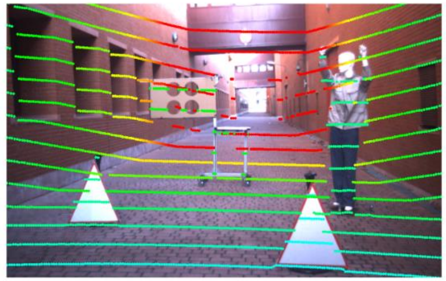
Work in progress!

Teamwork

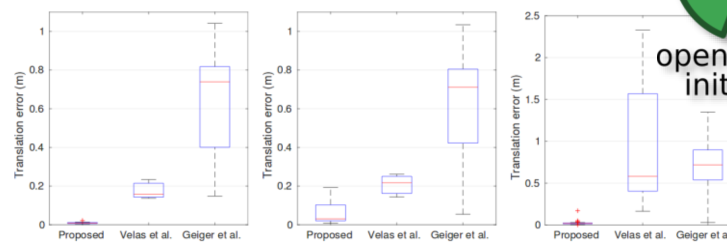
- Lidar-camera calibration. Solving a recurrent problem.



(a)



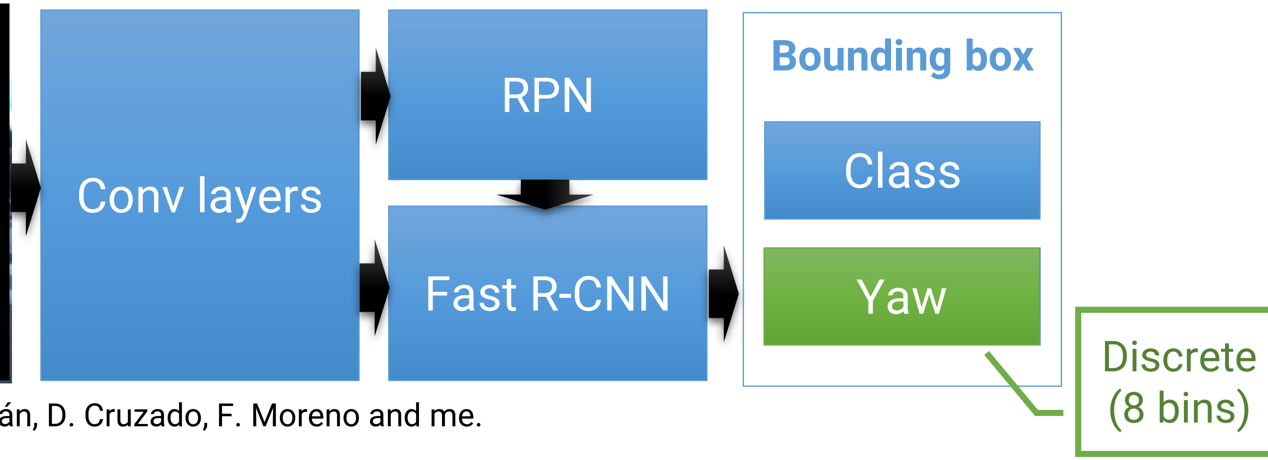
(b)



(a) 16-layer, trans. error (b) 32-layer, trans. error (c) 64-layer, trans. error

C. Guindel, J. Beltrán, D. Martín and F. García. "Automatic Extrinsic Calibration for Lidar-Stereo Vehicle Sensor Setups." arXiv:1705.04085 [cs.CV], 2017.

- Didi Challenge. Faster R-CNN applied over image-like inputs (bird-view).

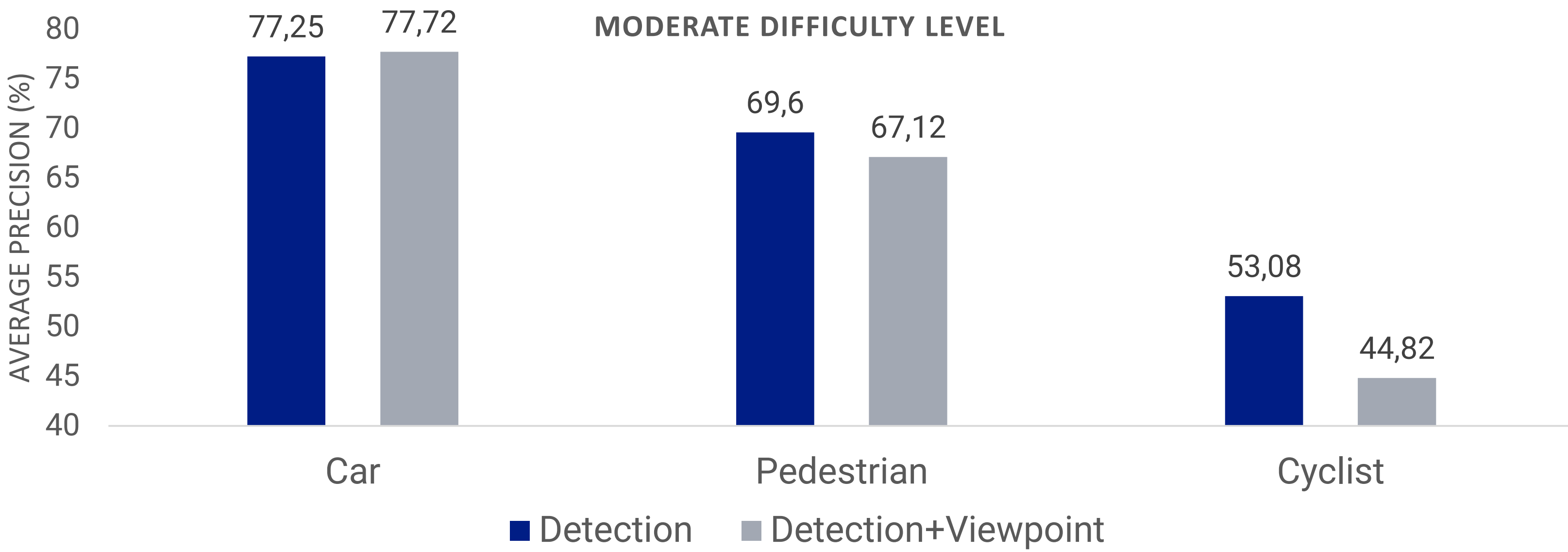


Proudly brought to you by J. Beltrán, D. Cruzado, F. Moreno and me.

Thank you for your attention!

Questions?

Precision change when introducing viewpoint



Detection (ΔAP as %)	Easy	Moderate	Hard
Car	+0,72	+0,47	+0,26
Pedestrian	-1,85	-2,47	-3,00
Cyclist	-12,16	-8,25	-8,11