Reference Detection for Off-road Self-Driving Vehicles using Deep Learning

Marcelo Eduardo Pederiva

School of Electrical and Computer Engineering
University of Campinas, Brazil
Email: *marcelopederiva@gmail.com*

Ely Carneiro de Paiva

School of Mechanical Engineering
University of Campinas, Brazil
Email: *elypaiva@fem.unicamp.br*

Reference Detection for Off-road Self-Driving Vehicles using Deep Learning

Marcelo Eduardo Pederiva holds a Bachelor's degree in Physics, from the Institute of Physics Gleb Wataghin - University of Campinas (Unicamp)/Brazil and a Master's degree in Mechanical Engineering - Unicamp/Brazil. At present, he has three years of experience in autonomous vehicles and the last two years in Machine Learning programming. The presenter, currently, is a Ph.D. student at the School of Electrical and Computer Engineering - Unicamp/Brazil, continuing the research and contributions in the autonomous vehicles field.

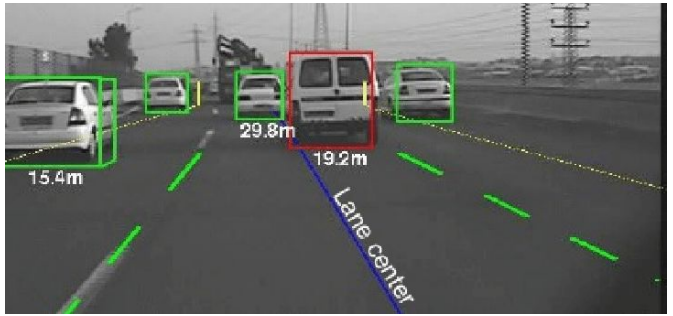
Marcelo Eduardo Pederiva

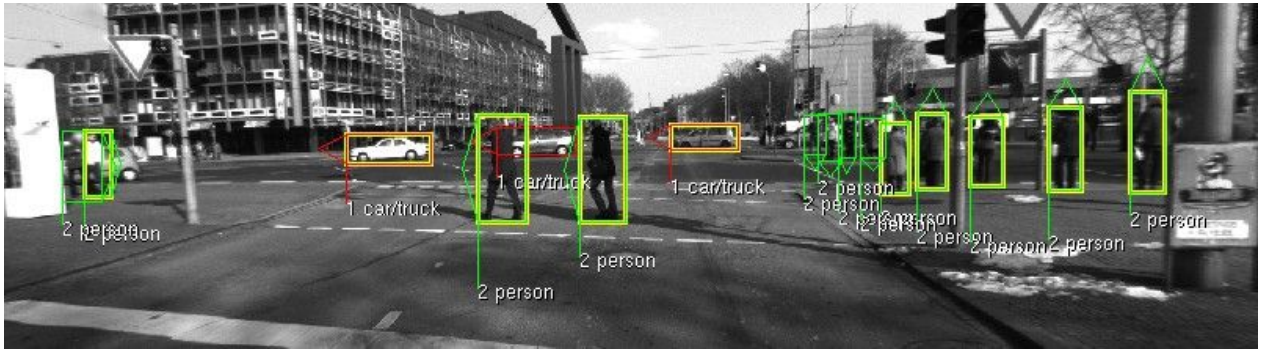
Introduction

Self-Driving Car Perception

Urban areas

- Traffic signs
- Traffic Light Colors
- Identification of pedestrians and cars
- Identification of lane lines





Introduction

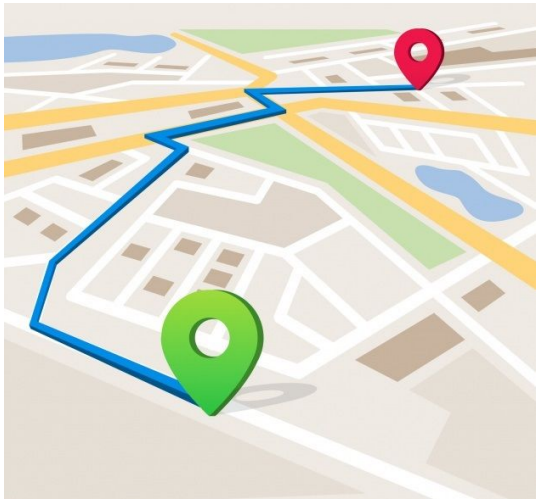
Self-Driving Car Perception

Urban areas

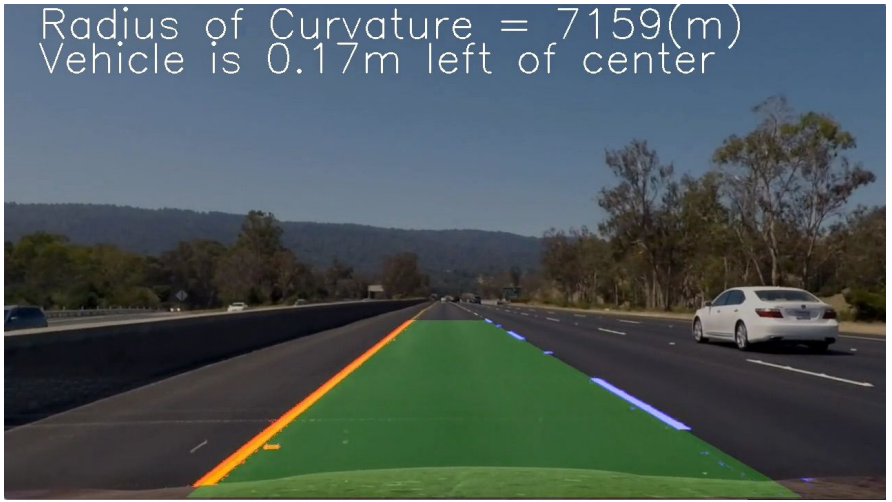
- Absence of traffic signs
- Imperfect terrain
- Eventual appearances of animals

Localization

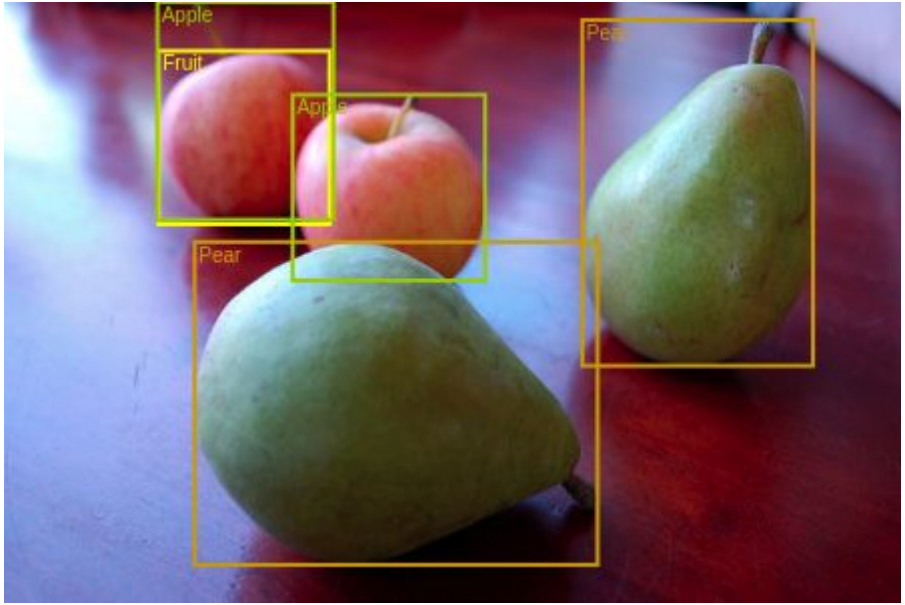
Global



Local



Object Detection



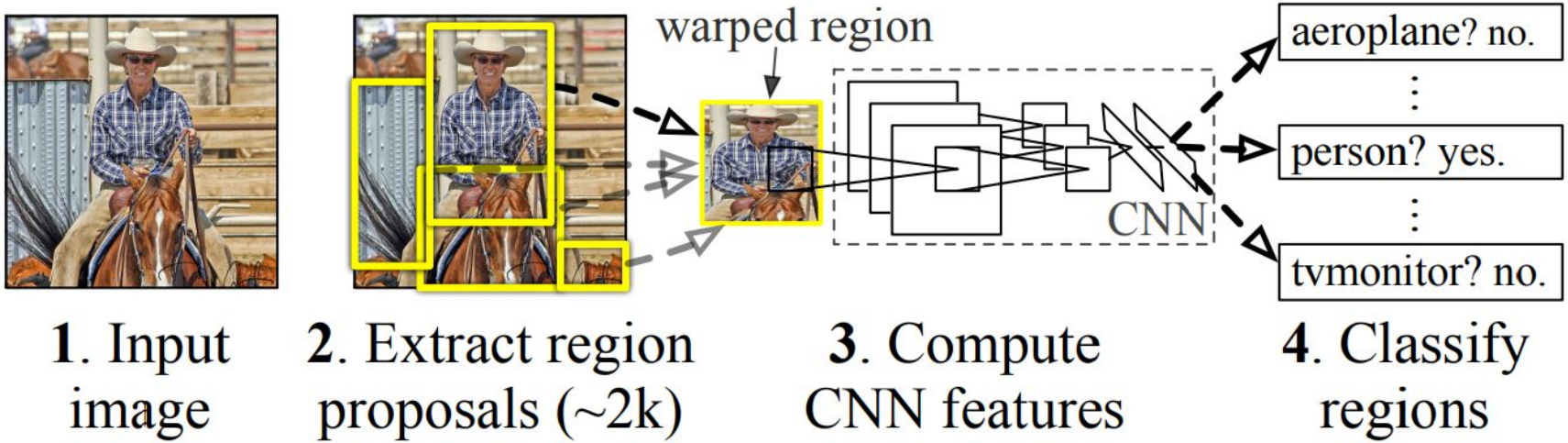
Detection Models

Model	Train	Test	mAP	FLOPS	FPS
SSD300	COCO trainval	test-dev	41.2	-	46
SSD500	COCO trainval	test-dev	46.5	-	19
YOLOv2 608x608	COCO trainval	test-dev	48.1	62.94 Bn	40
Tiny YOLO	COCO trainval	test-dev	23.7	5.41 Bn	244
SSD321	COCO trainval	test-dev	45.4	-	16
DSSD321	COCO trainval	test-dev	46.1	-	12
R-FCN	COCO trainval	test-dev	51.9	-	12
SSD513	COCO trainval	test-dev	50.4	-	8
DSSD513	COCO trainval	test-dev	53.3	-	6
FPN FRCN	COCO trainval	test-dev	59.1	-	6
Retinanet-50-500	COCO trainval	test-dev	50.9	-	14
Retinanet-101-500	COCO trainval	test-dev	53.1	-	11
Retinanet-101-800	COCO trainval	test-dev	57.5	-	5
YOLOv3-320	COCO trainval	test-dev	51.5	38.97 Bn	45
YOLOv3-416	COCO trainval	test-dev	55.3	65.86 Bn	35
YOLOv3-608	COCO trainval	test-dev	57.9	140.69 Bn	20
YOLOv3-tiny	COCO trainval	test-dev	33.1	5.56 Bn	220
YOLOv3-spp	COCO trainval	test-dev	60.6	141.45 Bn	20

Detection Frameworks	Train	mAP	FPS
Fast R-CNN [5]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[15]	2007+2012	73.2	7
Faster R-CNN ResNet[6]	2007+2012	76.4	5
YOLO [14]	2007+2012	63.4	45
SSD300 [11]	2007+2012	74.3	46
SSD500 [11]	2007+2012	76.8	19
YOLOv2 288 × 288	2007+2012	69.0	91
YOLOv2 352 × 352	2007+2012	73.7	81
YOLOv2 416 × 416	2007+2012	76.8	67
YOLOv2 480 × 480	2007+2012	77.8	59
YOLOv2 544 × 544	2007+2012	**78.6**	40

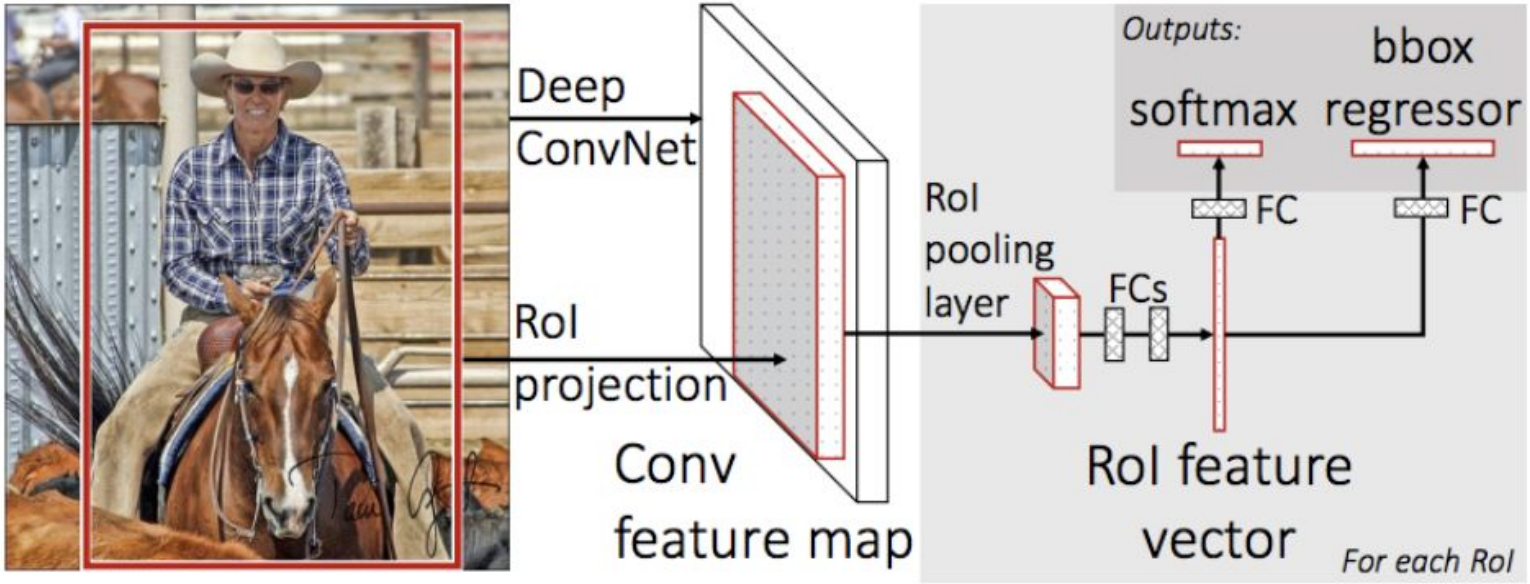
Faster R-CNN

R-CNN

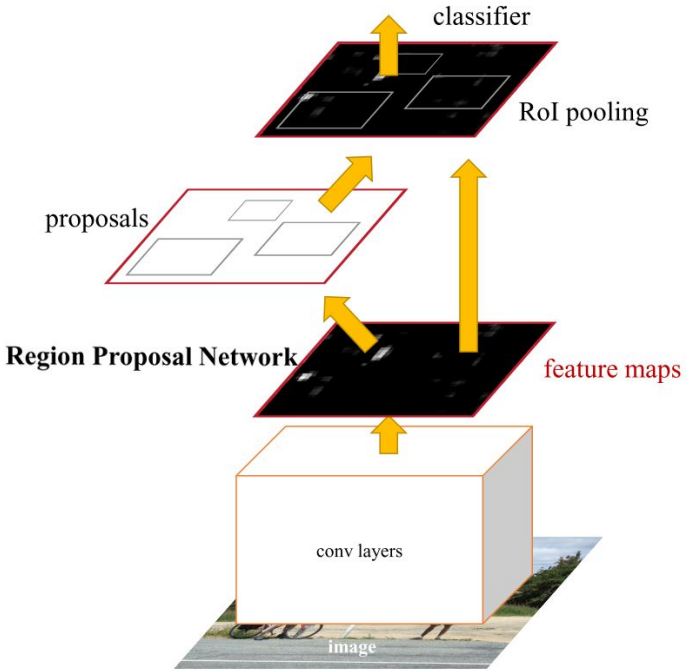


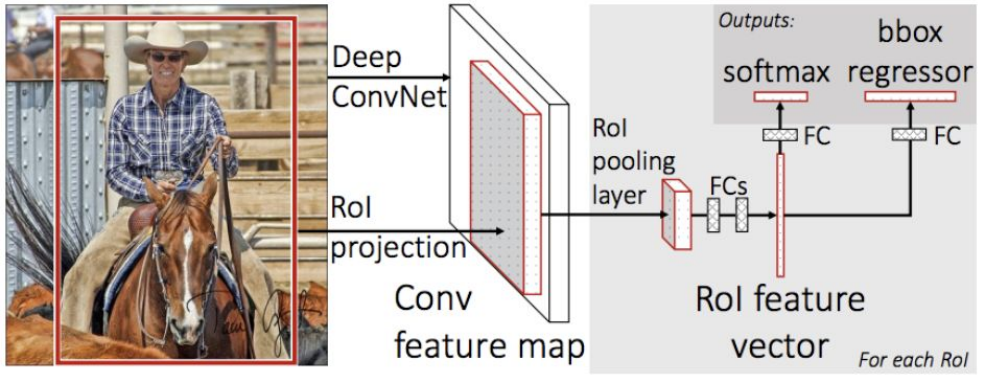
Faster R-CNN

Fast R-CNN



Faster R-CNN

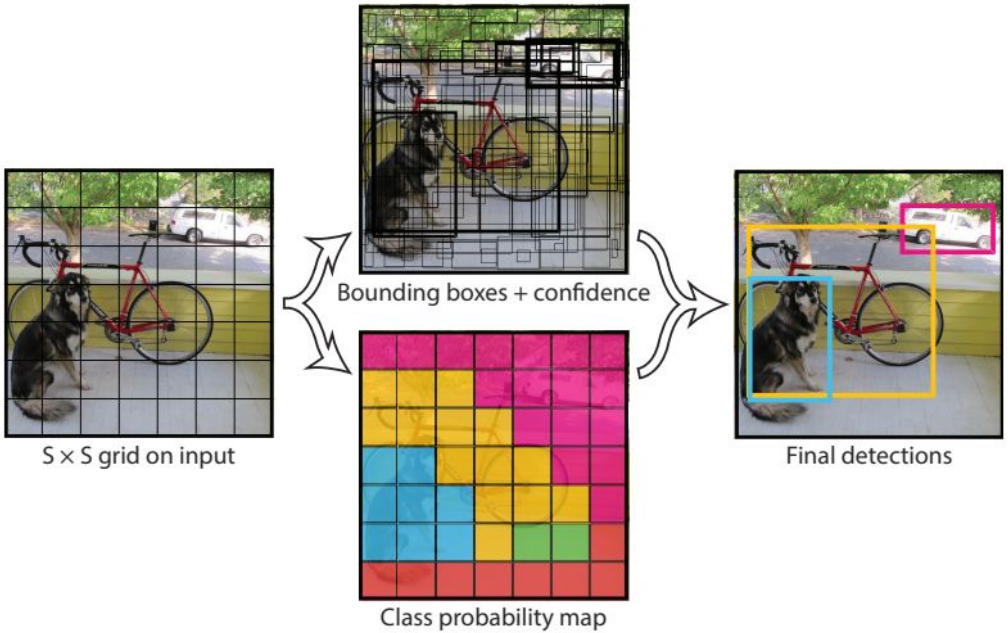

classifier
RoI pooling
proposals
Region Proposal Network
feature maps
conv layers
image


Deep
ConvNet
RoI
projection
Conv
feature map
RoI
pooling
layer
FCs
Outputs:
softmax
bbox
regressor
FC
FC
RoI feature
vector
For each RoI

YOLO (You Only Look Once)

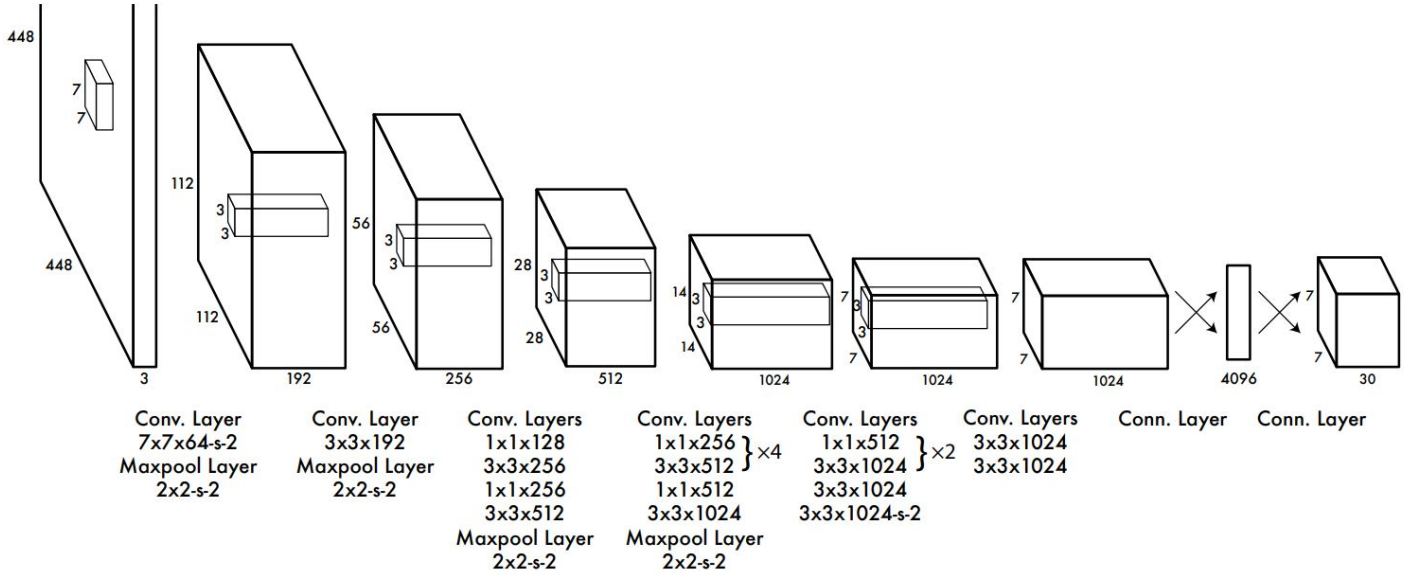
- Grid S X S
- B Bounding boxes in each cell
- C Number of classes

$S \times S \times (B*5 + C)$



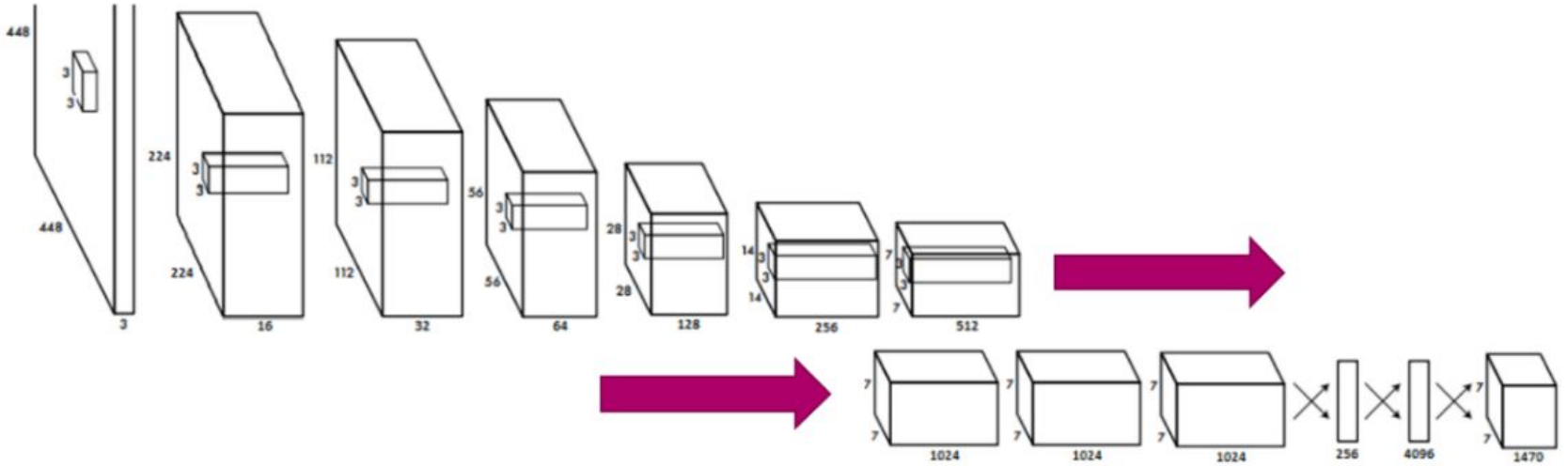
YOLO (You Only Look Once)

Regular YOLO



YOLO (You Only Look Once)

Fast (Tiny) YOLO

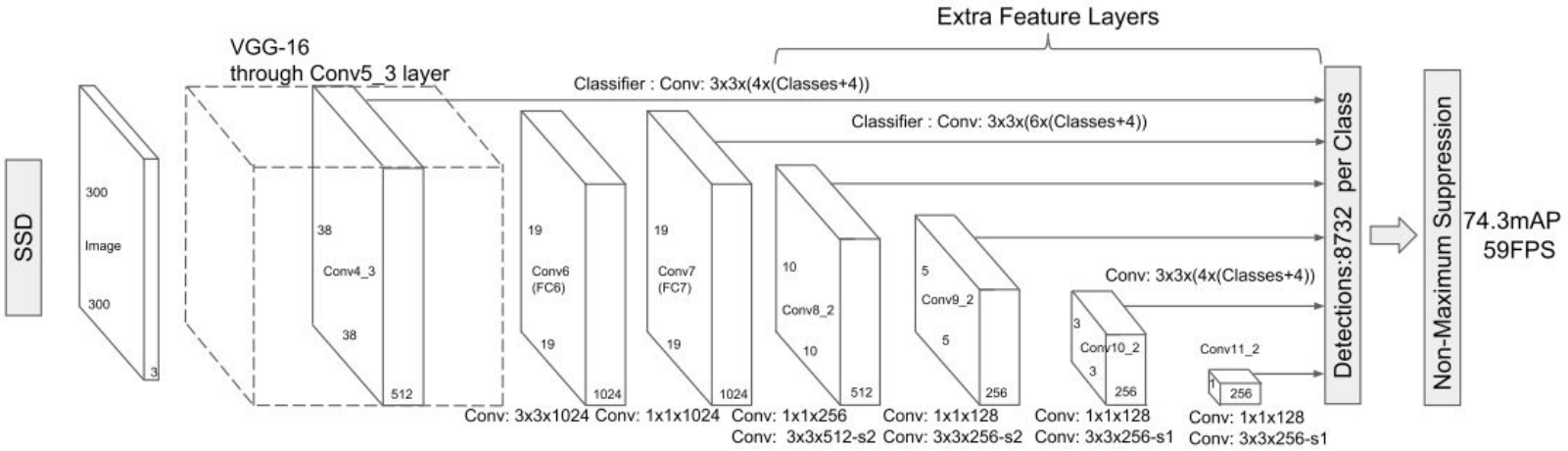


YOLO (You Only Look Once)

YOLOv2

	YOLO								YOLOv2
batch norm?		✓	✓	✓	✓	✓	✓	✓	✓
hi-res classifier?			✓	✓	✓	✓	✓	✓	✓
convolutional?				✓	✓	✓	✓	✓	✓
anchor boxes?				✓	✓				
new network?					✓	✓	✓	✓	✓
dimension priors?						✓	✓	✓	✓
location prediction?						✓	✓	✓	✓
passthrough?							✓	✓	✓
multi-scale?								✓	✓
hi-res detector?									✓
VOC2007 mAP	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8	**78.6**

SSD (Single Shot Detection)



SSD (Single Shot Detection)

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
GoogleNet	69.8%	1550	6.8
VGG 16	71.5%	15300	138

Network	Top 1	Params	MAdds	CPU
MobileNetV1	70.6	4.2M	575M	113ms
ShuffleNet (1.5)	71.5	**3.4M**	292M	-
ShuffleNet (x2)	73.7	5.4M	524M	-
NasNet-A	74.0	5.3M	564M	183ms
MobileNetV2	**72.0**	**3.4M**	**300M**	**75ms**
MobileNetV2 (1.4)	**74.7**	6.9M	585M	**143ms**

SSD (Single Shot Detection)

Method	mAP	FPS	# Boxes	Input resolution
SSD300	74.3	46	8732	300×300
SSD512	76.8	19	24564	512×512

Training Step

- Faster R-CNN
- Fast YOLOv2
- SSD300

Accuracy

- Localization Loss
- Video test

Response time

- Mean of the time process in 10 images

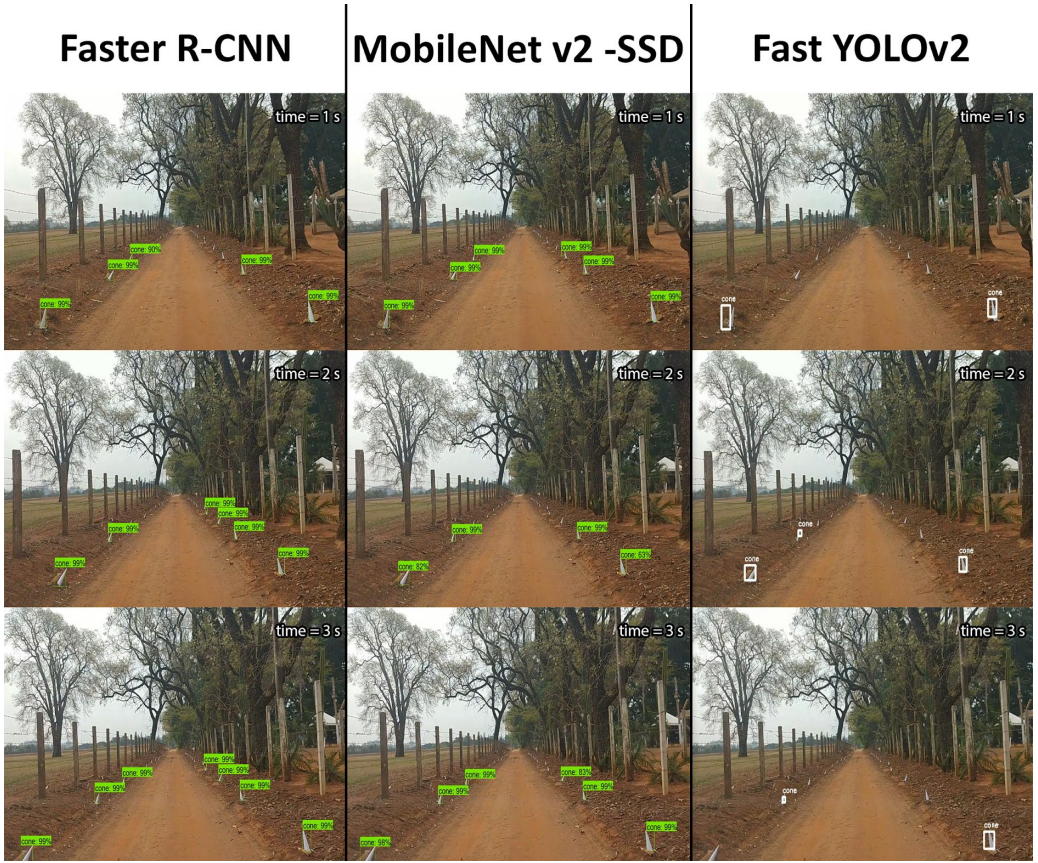
Training Step

- Gpu Nvidia Geforce 1060 3GB
- CPU Intel i5-8500u
- Windows 10

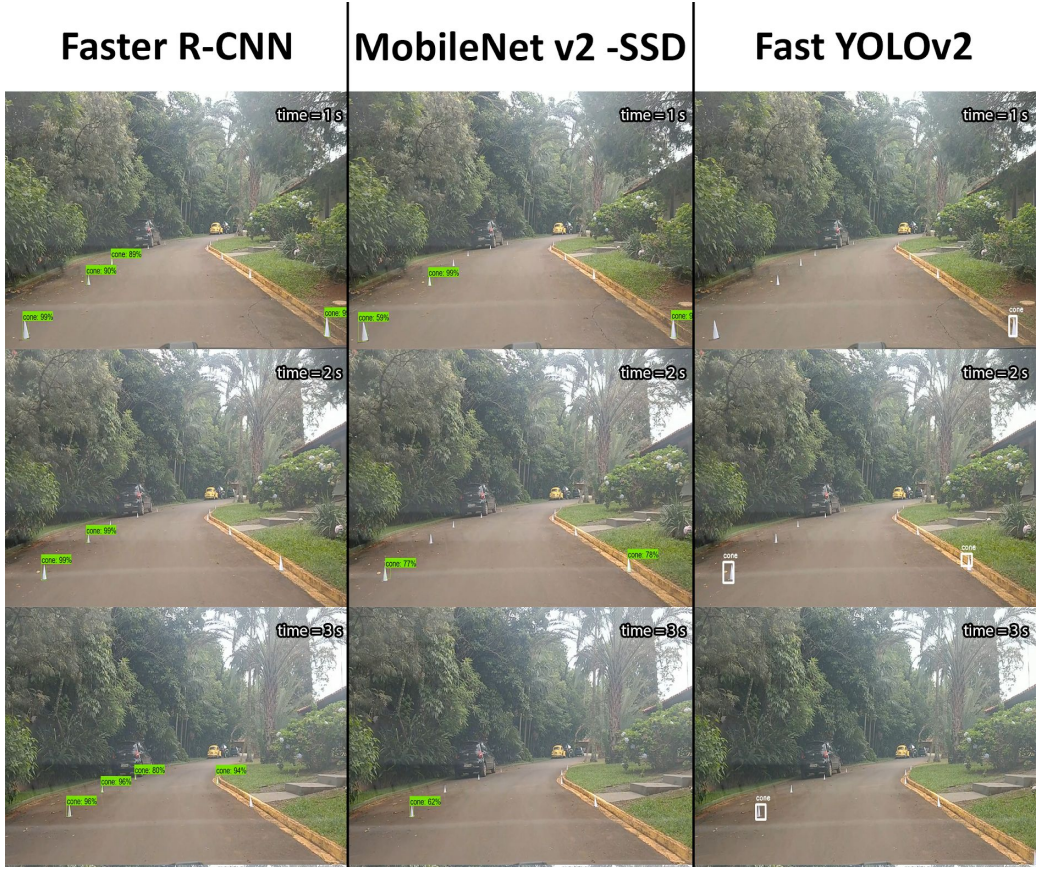
Training Step

Models	Training	Localization Loss
Faster R-CNN	296 images	0.017
Fast YOLOv2	296 images	1.500
MobileNetv2 SSD300	296 images	0.258

Experiment 1



Experiment 2



Results

Models	Training (images)	Localization Loss	Mean Process Time (seconds)	FPS
Faster R-CNN	296	0.017	3.14	00.3
Fast YOLOv2	296	1.500	0.07	14.3
MobileNetv2 SSD300	296	0.258	1.41	00.7

Conclusion

Faster R-CNN

Mobilenetv2 SSD300

Fast YOLOv2

- Slow detection, but accurate (Faster R-CNN)
- Fast detection with bad precision (Fast YOLOv2)
- Accurate detection with a intermediary response time (SSD300)
- Merge two methods for applications (Fast YOLOv2 e SSD300)

Thank You