Deep Learning Concepts and Datasets for Image Recognition: Overview 2019

Case Study: Pedestrian Detection

Dr. Karel Horak*[a,b], Prof. Robert Sablatnig[b]

[a]Brno University of Technology, Czech Republic

[b]Computer Vision Lab, Vienna University of Technology, Austria

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MŠMT
MINISTRY OF EDUCATION,
YOUTH AND SPORTS

Our goal is to detect pedestrians on images from on-board camera (Advanced driver-assistance systems)

SSD-Lite-Pedestrian-detection with MobileNet v2 as feature extractor on Mixed Dataset

Deep Learning Architectures

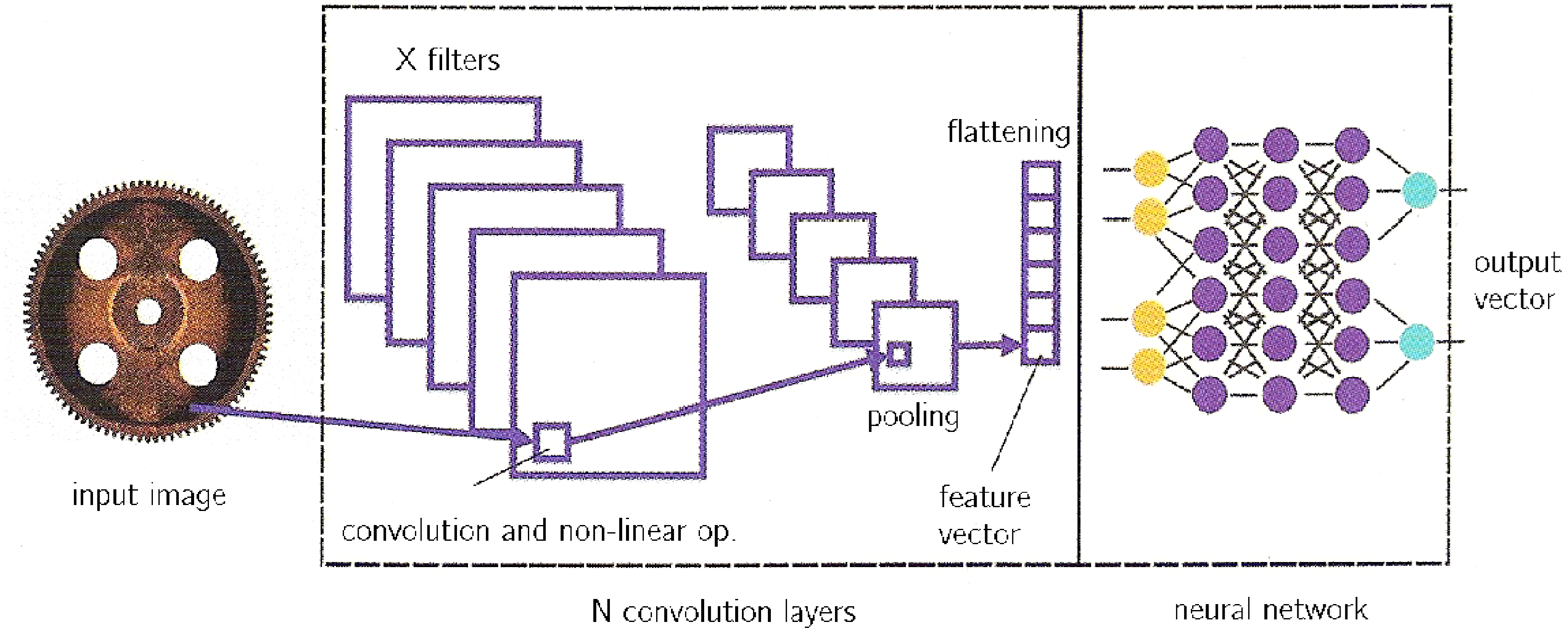
There is a lot of different architectures of Convolutional Neural Networks designed for image recognition:

Objects detectors:

- R-CNN family – Region-based CNN (R-CNN, Fast R-CNN, Faster R-CNN, R-FCN, Mask R-CNN)
- SSD – Single Shot MultiBox Detector
- YOLO – You Only Look Once
- RetinaNet – uses ResNet as backbone

Objects classifiers:

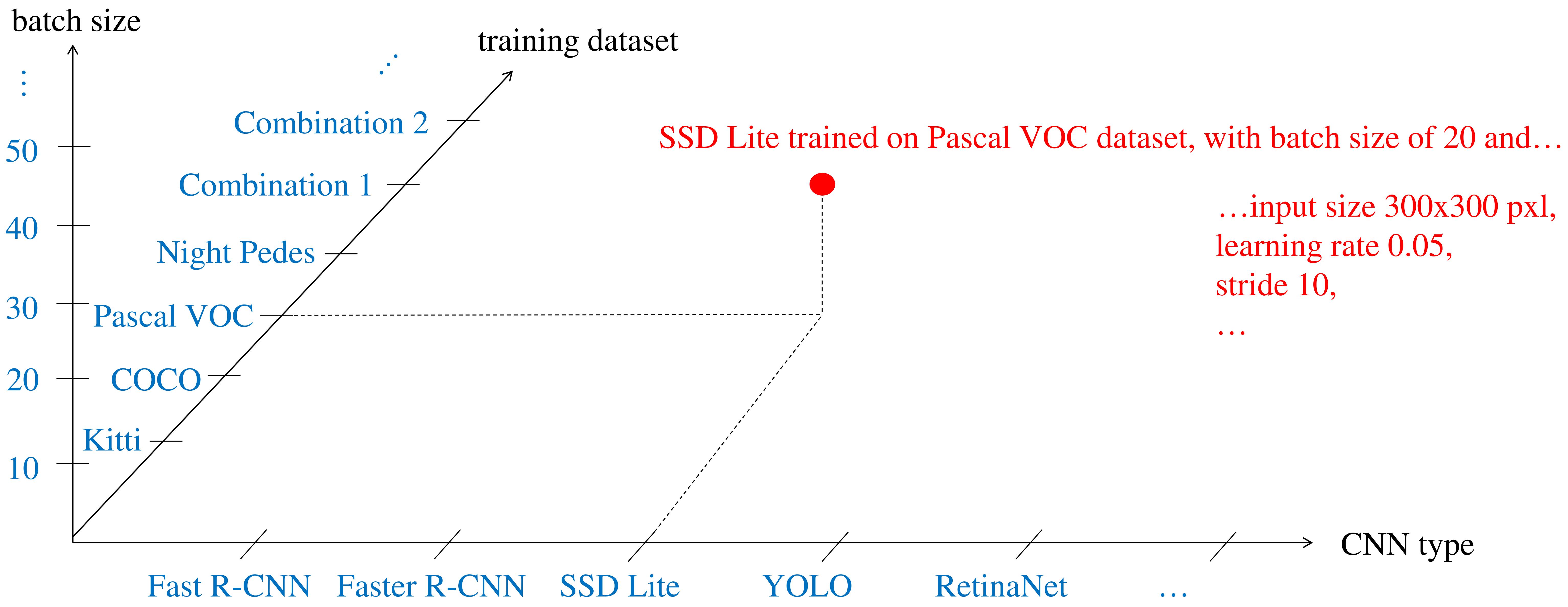
- AlexNet
- VGG16
- GoogleNet
- ResNet – Residual Neural Network



Deep Learning Architectures

Q: How to choose the proper one?

A: Find solution of **optimization problem** – searching of unknown parameters in **high-dimensional** space.



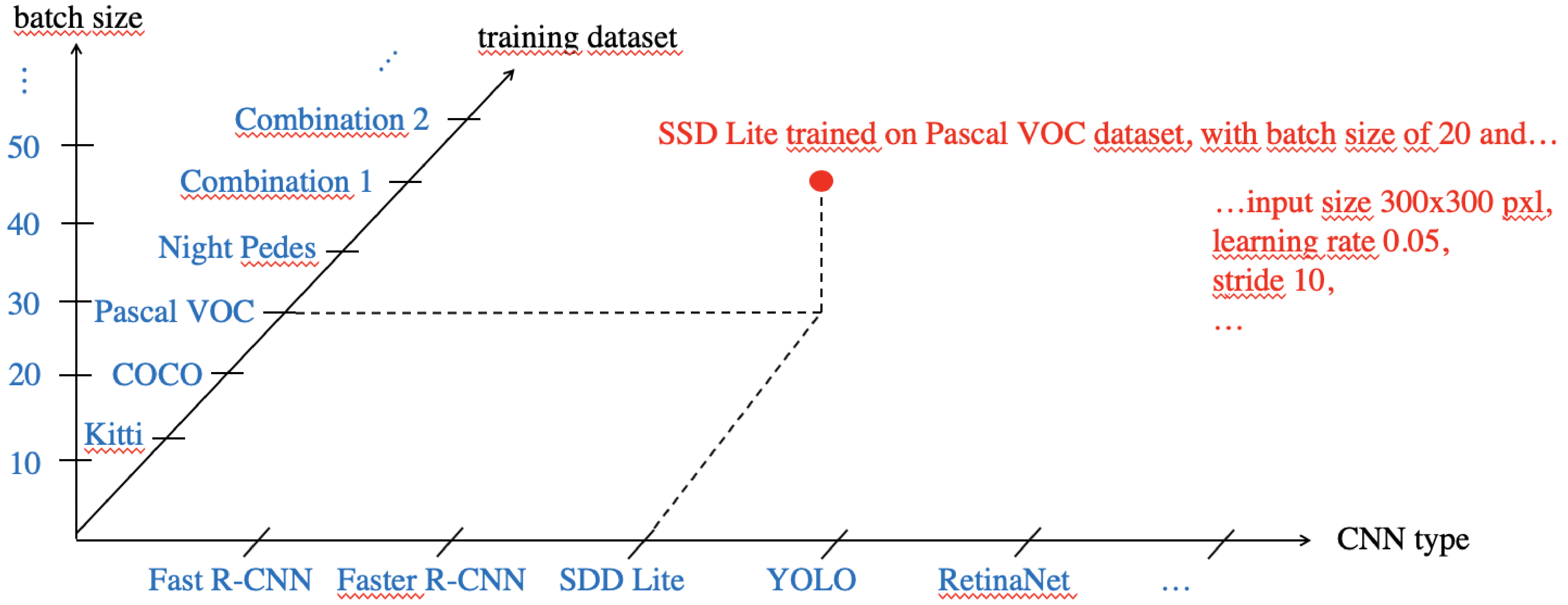
Only three dimensions are displayable here, nevertheless we have much more than only these to analyse!

Deep Learning Architectures

We know how to solve optimization tasks, but two problems arise:

1. Principal (theoretical) problem of solution: adjoining items on axes does not create neither sequential nor linear space (e.g. as ordinal numbers 1, 2, 3, …) => not measured values (combinations) can not be interpolated or estimated => only **brute force** optimization can be used.

Unfortunately, brute force solution necessarily creates the other one problem…



Deep Learning Architectures

We know how to solve optimization tasks, but two problems arise:

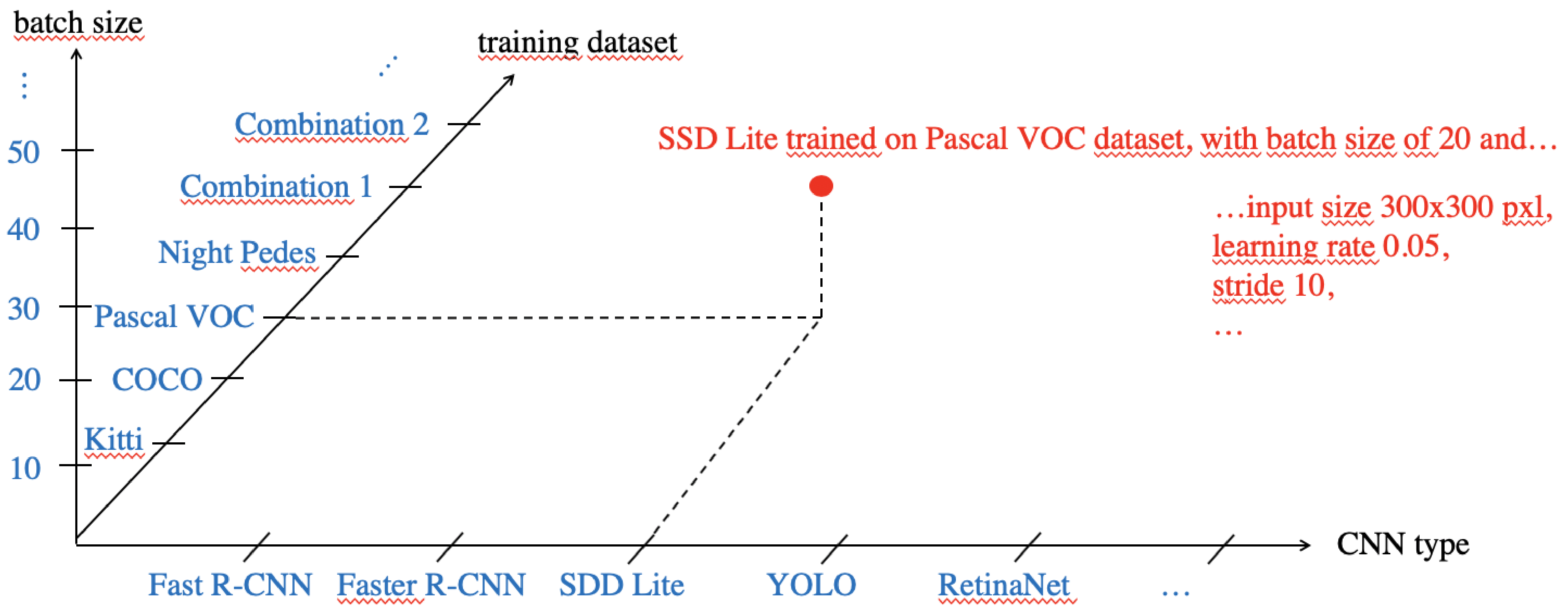
2. Practical problem of solution: measuring (= training) each type of the CNN on each available dataset (lots of variants and even more combinations of them) with each admissible value of each parameter (batch size, input size, learning rate, stride, networks structure,…) is not computable neither on any current hardware nor cloud!

Try to estimate number of all solution in this space and time needed to train:

tens of CNNs x hundreds of datasets x thousands of parameters values

=~ 10^8 solutions and above

=> almost 4000 years of computing!



SSD Lite trained on Pascal VOC dataset, with batch size of 20 and…

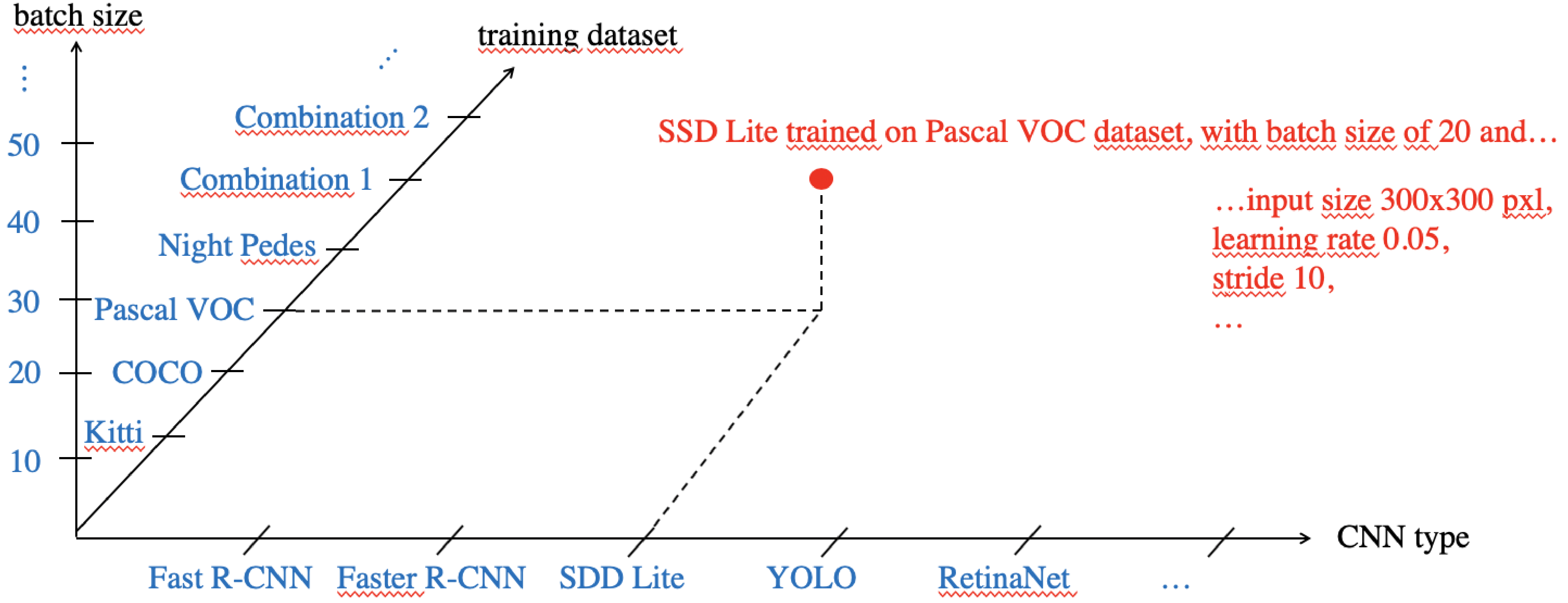
…input size 300x300 pxl,
learning rate 0.05,
stride 10,
…

Deep Learning Architectures

Chosen “enforced” solution:

Best practise = use pretrained network to fix some parameters at least (mainly CNNs structure and basic parameters as stride and learning rate).

What dimensions remained to optimize? CNN type, datasets and their combinations, input size, dataset volume (we evaluated 10, 100, 500, 800, 2k and 5k images sets).



SSD Lite trained on Pascal VOC dataset, with batch size of 20 and…

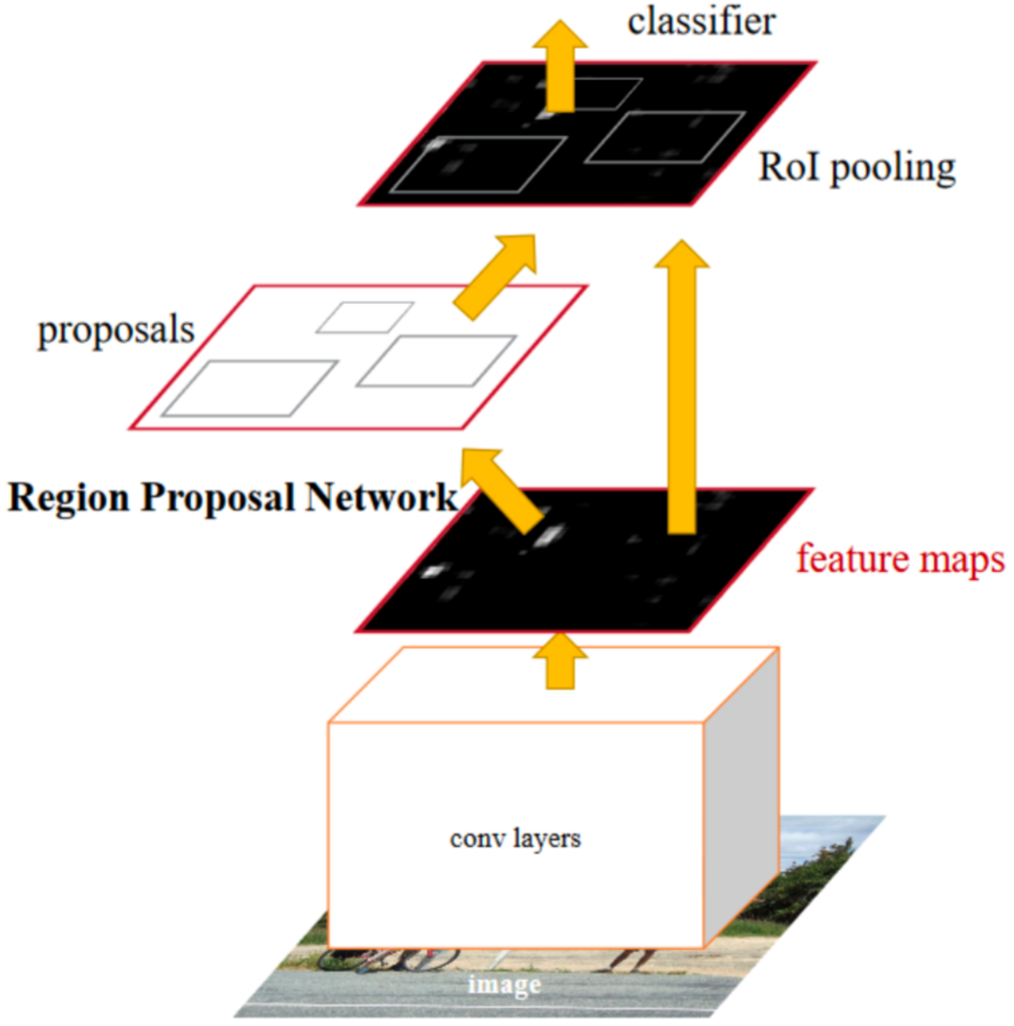
…input size 300x300 pxl,
learning rate 0.05,
stride 10,
…

Implementation Details

Two meta-architectures of objects detectors were selected for training:

a) Faster R-CNN with Resnet101 as a backbone network

b) SSD Lite with Mobilenet v2 as a backbone network

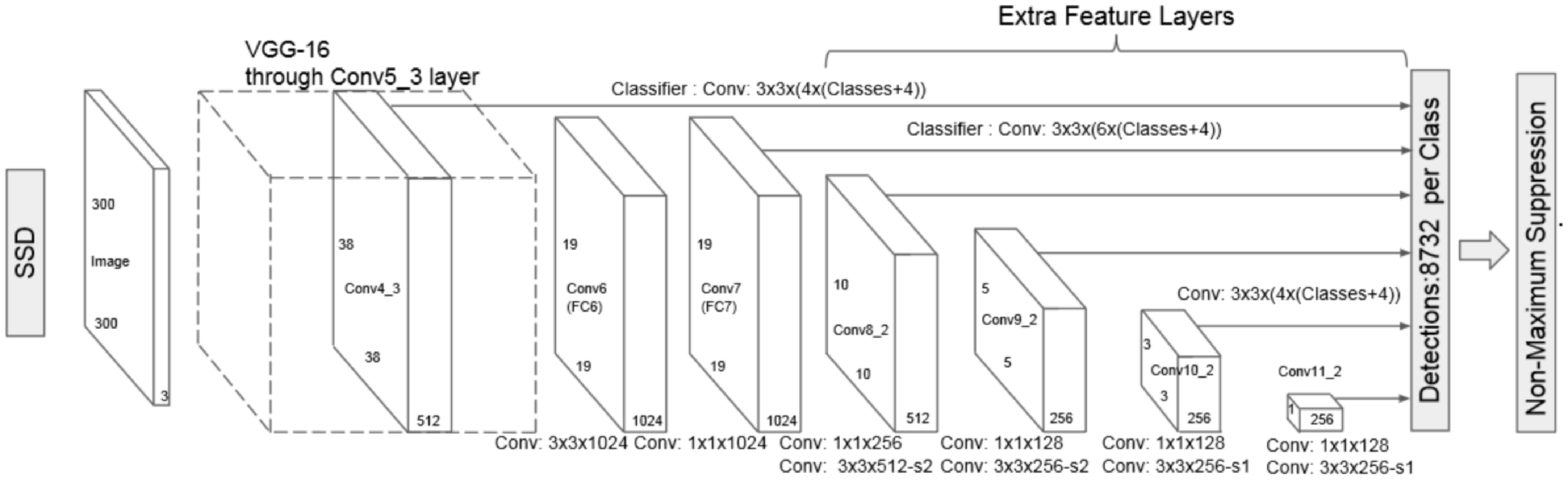


Libraries used: TensorFlow-gpu 1.12 + Object Detection API

GPU used for training phase:

Nvidia GeForce MX 150 v1/2 GB

Nvidia Titan X - 12 GB



Datasets Used – Publicly Available

COCO (used for testing, not training): contains 328k images in 80 categories, 250k labelled people

Kitti (complex dataset of 2D, 3D and Bird’s eye views): contains 15k images, 2k labelled pedestrians

Pascal VOC: 20 categories

CityShapes: contains 25k images in 30 categories, 3.5k labelled persons (+metadata: temperature and GPS)

Datasets Used – Proprietary Ad-hoc Dataset

Because of not any night images are present in the previous datasets at all, we have created the one:

Night Pedestrian: contains 227 images in one category of persons, 815 labelled pedestrians.

Note the LabelIMG tool has been used to manual annotations of pedestrians.

Detector Accuracy – object hit or missed and how much?

Problem: practically, any detector does not predict exact pixel position (caused by pooling, stride, pyramid scale, etc.) – difference between detectors (results in position) and classifiers (results in labels).

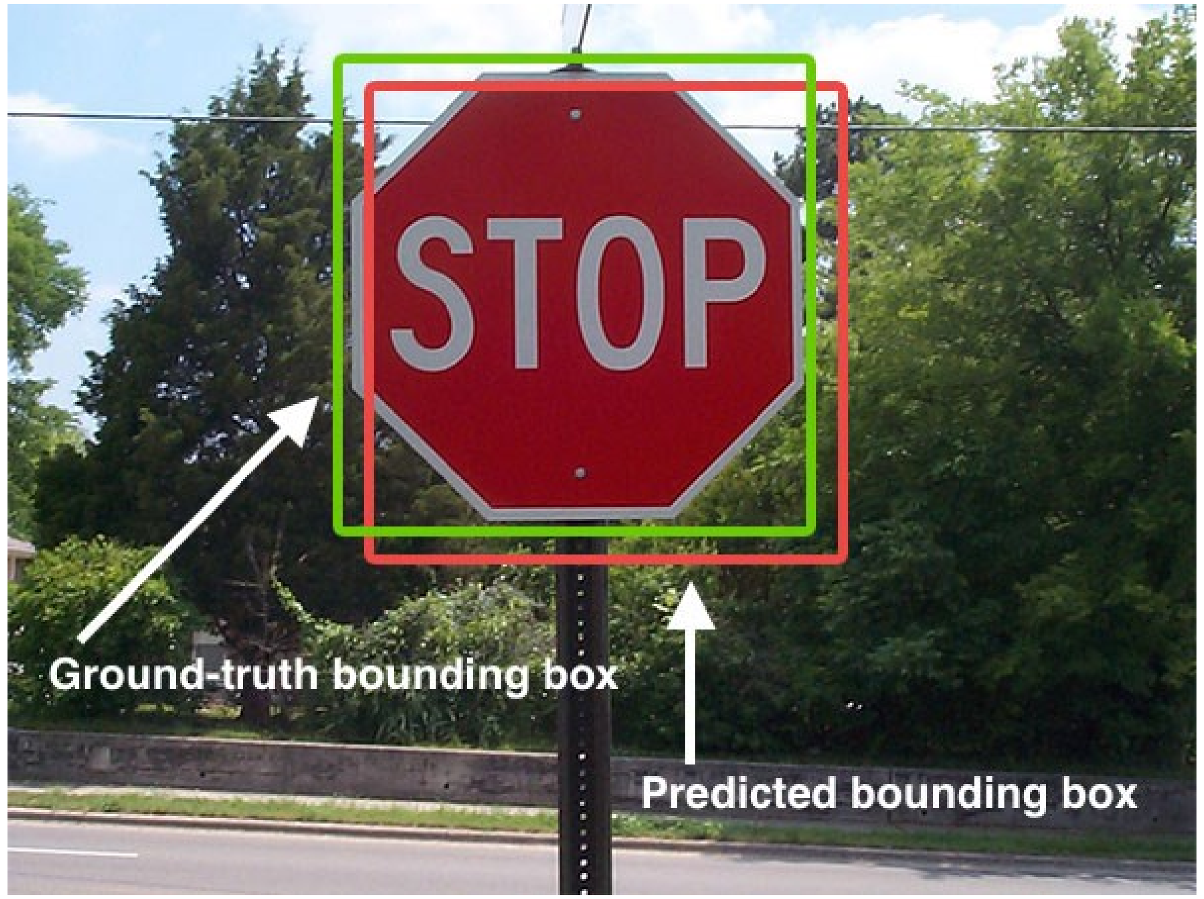
Q: How to evaluate an accuracy of the detector once it is trained?

A: Intersection of Union (IoU) method = an evaluation metric used to measure the accuracy of an object detector on a particular dataset.

Requirements: in order to apply IoU to evaluate the object detector we need:

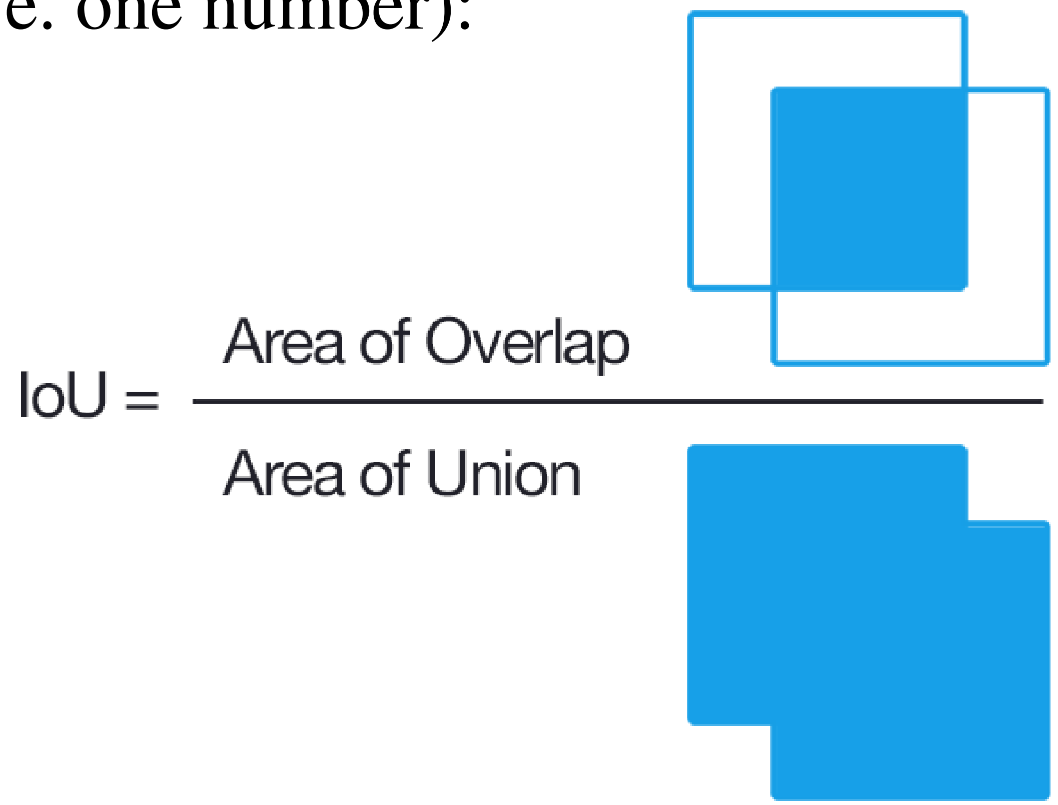
1. The ground-truth bounding boxes (i.e., the hand labelled bounding boxes from the testing dataset that specify where our object is in the image).

2. The predicted bounding boxes by the detector.



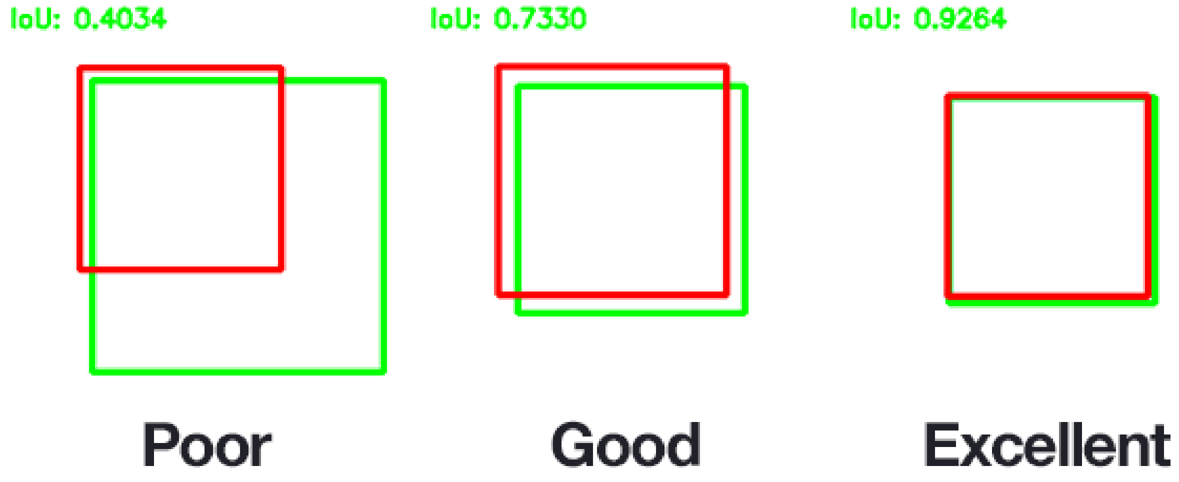
Detector Accuracy – object hit or missed and how much?

Intersection over Union is simple overlapping ratio - **score** (i.e. one number):



$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Good practise: Intersection over Union score > 0.5 is normally considered a “good” prediction.



Detectors Efficiency

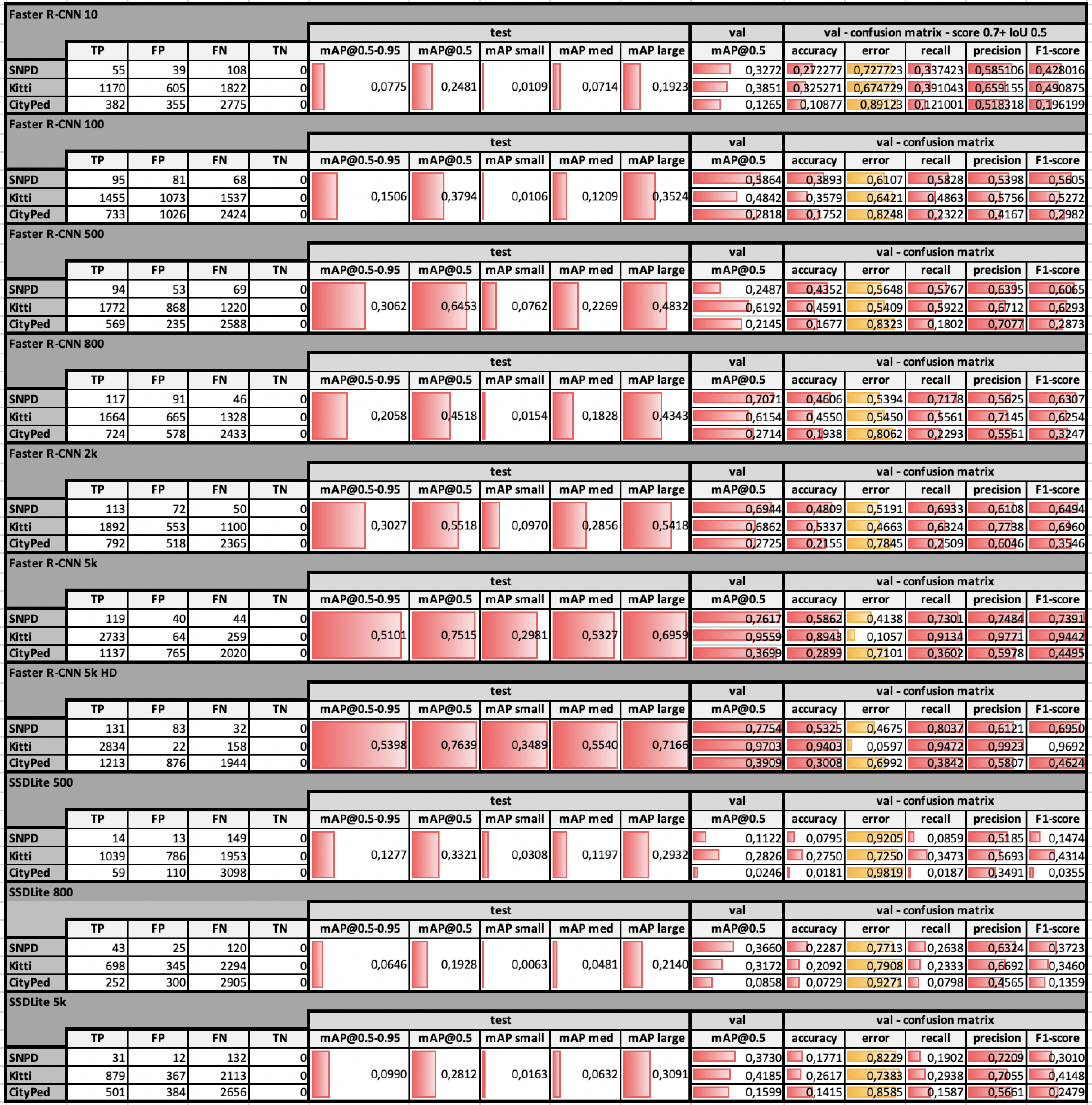
As soon as we know **score**, we know if prediction corresponds to the ground-truth box (by means of given threshold for score):

1. score > thr □ increase value of TP

2. score <= thr □ increase value of FP

3. for all ground-truth boxes never detected increase value of FN

Note TN does not apply: background detection



Faster R-CNN 10															
					test					val	val - confusion matrix - score 0.7+ IoU 0.5				
	TP	FP	FN	TN	mAP@0.5-0.95	mAP@0.5	mAP small	mAP med	mAP large	mAP@0.5	accuracy	error	recall	precision	F1-score
SNPD	55	39	108	0	0,0775	0,2481	0,0109	0,0714	0,1923	0,3272	0,272277	0,727723	0,337423	0,585106	0,428016
Kitti	1170	605	1822	0						0,3851	0,325271	0,674729	0,391043	0,659155	0,490875
CityPed	382	355	2775	0						0,1265	0,10877	0,89123	0,121001	0,518318	0,196199

Faster R-CNN 100															
					test					val	val - confusion matrix				
	TP	FP	FN	TN	mAP@0.5-0.95	mAP@0.5	mAP small	mAP med	mAP large	mAP@0.5	accuracy	error	recall	precision	F1-score
SNPD	95	81	68	0	0,1506	0,3794	0,0106	0,1209	0,3524	0,5864	0,3893	0,6107	0,5828	0,5398	0,5605
Kitti	1455	1073	1537	0						0,4842	0,3579	0,6421	0,4863	0,5756	0,5272
CityPed	733	1026	2424	0						0,2818	0,1752	0,8248	0,2322	0,4167	0,2982

Faster R-CNN 500															
					test					val	val - confusion matrix				
	TP	FP	FN	TN	mAP@0.5-0.95	mAP@0.5	mAP small	mAP med	mAP large	mAP@0.5	accuracy	error	recall	precision	F1-score
SNPD	94	53	69	0	0,3062	0,6453	0,0762	0,2269	0,4832	0,2487	0,4352	0,5648	0,5767	0,6395	0,6065
Kitti	1772	868	1220	0						0,6192	0,4591	0,5409	0,5922	0,6712	0,6293
CityPed	569	235	2588	0						0,2145	0,1677	0,8323	0,1802	0,7077	0,2873

Faster R-CNN 800															
					test					val	val - confusion matrix				
	TP	FP	FN	TN	mAP@0.5-0.95	mAP@0.5	mAP small	mAP med	mAP large	mAP@0.5	accuracy	error	recall	precision	F1-score
SNPD	117	91	46	0	0,2058	0,4518	0,0154	0,1828	0,4343	0,7071	0,4606	0,5394	0,7178	0,5625	0,6307
Kitti	1664	665	1328	0						0,6154	0,4550	0,5450	0,5561	0,7145	0,6254
CityPed	724	578	2433	0						0,2714	0,1938	0,8062	0,2293	0,5561	0,3247

Faster R-CNN 2k															
					test					val	val - confusion matrix				
	TP	FP	FN	TN	mAP@0.5-0.95	mAP@0.5	mAP small	mAP med	mAP large	mAP@0.5	accuracy	error	recall	precision	F1-score
SNPD	113	72	50	0	0,3027	0,5518	0,0970	0,2856	0,5418	0,6944	0,4809	0,5191	0,6933	0,6108	0,6494
Kitti	1892	553	1100	0						0,6862	0,5337	0,4663	0,6324	0,7738	0,6960
CityPed	792	518	2365	0						0,2725	0,2155	0,7845	0,2509	0,6046	0,3546

Faster R-CNN 5k															
					test					val	val - confusion matrix				
	TP	FP	FN	TN	mAP@0.5-0.95	mAP@0.5	mAP small	mAP med	mAP large	mAP@0.5	accuracy	error	recall	precision	F1-score
SNPD	119	40	44	0	0,5101	0,7515	0,2981	0,5327	0,6959	0,7617	0,5862	0,4138	0,7301	0,7484	0,7391
Kitti	2733	64	259	0						0,9559	0,8943	0,1057	0,9134	0,9771	0,9442
CityPed	1137	765	2020	0						0,3699	0,2899	0,7101	0,3602	0,5978	0,4495

Faster R-CNN 5k HD															
					test					val	val - confusion matrix				
	TP	FP	FN	TN	mAP@0.5-0.95	mAP@0.5	mAP small	mAP med	mAP large	mAP@0.5	accuracy	error	recall	precision	F1-score
SNPD	131	83	32	0	0,5398	0,7639	0,3489	0,5540	0,7166	0,7754	0,5325	0,4675	0,8037	0,6121	0,6950
Kitti	2834	22	158	0						0,9703	0,9403	0,0597	0,9472	0,9923	0,9692
CityPed	1213	876	1944	0						0,3909	0,3008	0,6992	0,3842	0,5807	0,4624

SSDLite 500															
					test					val	val - confusion matrix				
	TP	FP	FN	TN	mAP@0.5-0.95	mAP@0.5	mAP small	mAP med	mAP large	mAP@0.5	accuracy	error	recall	precision	F1-score
SNPD	14	13	149	0	0,1277	0,3321	0,0308	0,1197	0,2932	0,1122	0,0795	0,9205	0,0859	0,5185	0,1474
Kitti	1039	786	1953	0						0,2826	0,2750	0,7250	0,3473	0,5693	0,4314
CityPed	59	110	3098	0						0,0246	0,0181	0,9819	0,0187	0,3491	0,0355

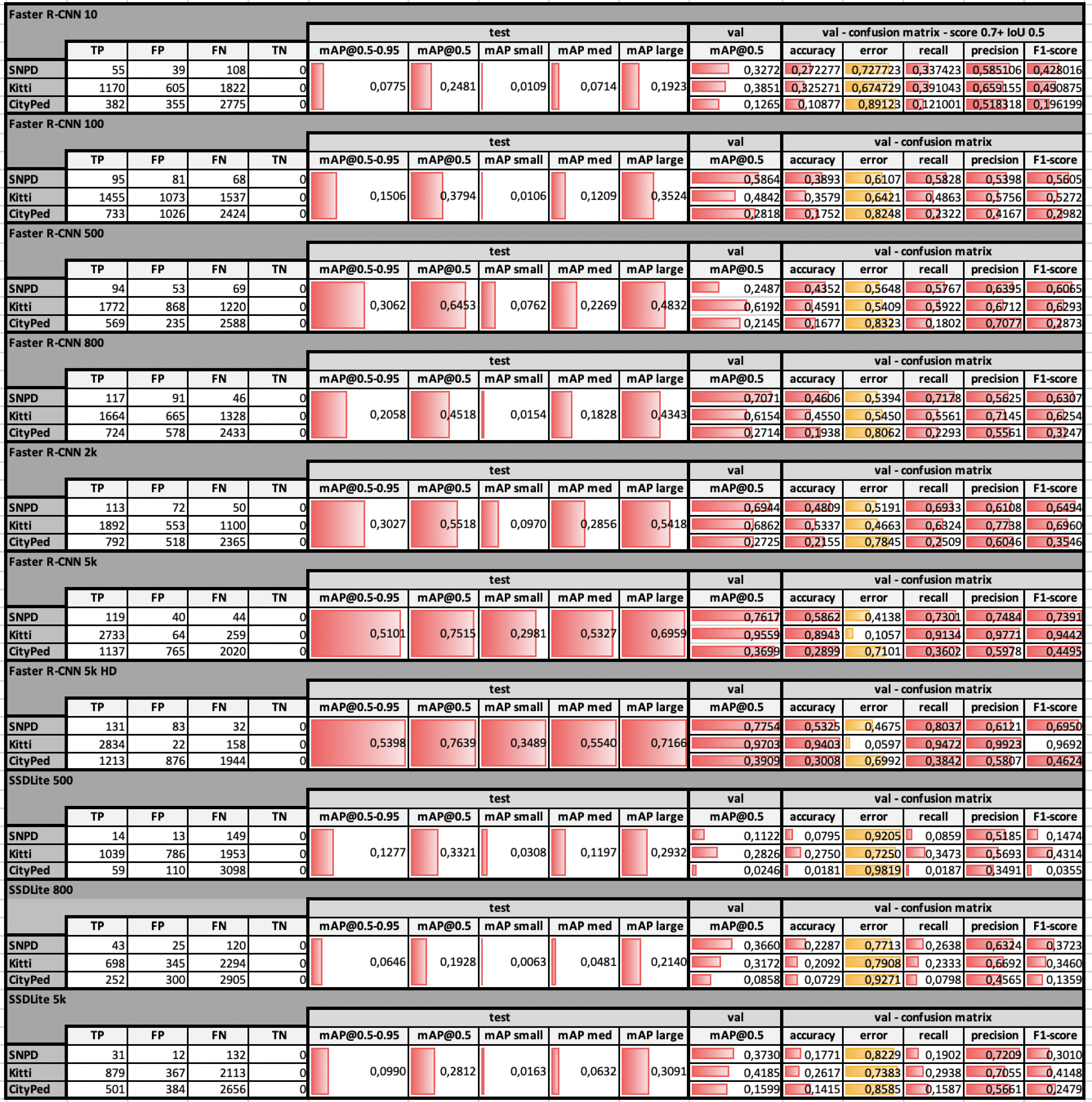
SSDLite 800															
					test					val	val - confusion matrix				
	TP	FP	FN	TN	mAP@0.5-0.95	mAP@0.5	mAP small	mAP med	mAP large	mAP@0.5	accuracy	error	recall	precision	F1-score
SNPD	43	25	120	0	0,0646	0,1928	0,0063	0,0481	0,2140	0,3660	0,2287	0,7713	0,2638	0,6324	0,3723
Kitti	698	345	2294	0						0,3172	0,2092	0,7908	0,2333	0,6692	0,3460
CityPed	252	300	2905	0						0,0858	0,0729	0,9271	0,0798	0,4565	0,1359

SSDLite 5k															
					test					val	val - confusion matrix				
	TP	FP	FN	TN	mAP@0.5-0.95	mAP@0.5	mAP small	mAP med	mAP large	mAP@0.5	accuracy	error	recall	precision	F1-score
SNPD	31	12	132	0	0,0990	0,2812	0,0163	0,0632	0,3091	0,3730	0,1771	0,8229	0,1902	0,7209	0,3010
Kitti	879	367	2113	0						0,4185	0,2617	0,7383	0,2938	0,7055	0,4148
CityPed	501	384	2656	0						0,1599	0,1415	0,8585	0,1587	0,5661	0,2479

Detectors Efficiency

As soon as we know TP and FP values, the **mAP** (mean Average Precision) serves as a Detector efficiency indicator:

$$mAP = \frac{1}{|classes|} \sum_{c \in classes} \frac{\#TP(c)}{\#TP(c) + \#FP(c)}$$



Faster R-CNN 10

					test					val	val - confusion matrix - score 0.7+ IoU 0.5				
	TP	FP	FN	TN	mAP@0.5-0.95	mAP@0.5	mAP small	mAP med	mAP large	mAP@0.5	accuracy	error	recall	precision	F1-score
SNPD	55	39	108	0	0,0775	0,2481	0,0109	0,0714	0,1923	0,3272	0,272277	0,727723	0,337423	0,585106	0,428016
Kitti	1170	605	1822	0						0,3851	0,325271	0,674729	0,391043	0,659155	0,490875
CityPed	382	355	2775	0						0,1265	0,10877	0,89123	0,121001	0,518318	0,196199

Faster R-CNN 100

					test					val	val - confusion matrix				
	TP	FP	FN	TN	mAP@0.5-0.95	mAP@0.5	mAP small	mAP med	mAP large	mAP@0.5	accuracy	error	recall	precision	F1-score
SNPD	95	81	68	0	0,1506	0,3794	0,0106	0,1209	0,3524	0,5864	0,3893	0,6107	0,5828	0,5398	0,5605
Kitti	1455	1073	1537	0						0,4842	0,3579	0,6421	0,4863	0,5756	0,5272
CityPed	733	1026	2424	0						0,2818	0,1752	0,8248	0,2322	0,4167	0,2982

Faster R-CNN 500

					test					val	val - confusion matrix				
	TP	FP	FN	TN	mAP@0.5-0.95	mAP@0.5	mAP small	mAP med	mAP large	mAP@0.5	accuracy	error	recall	precision	F1-score
SNPD	94	53	69	0	0,3062	0,6453	0,0762	0,2269	0,4832	0,2487	0,4352	0,5648	0,5767	0,6395	0,6065
Kitti	1772	868	1220	0						0,6192	0,4591	0,5409	0,5922	0,6712	0,6293
CityPed	569	235	2588	0						0,2145	0,1677	0,8323	0,1802	0,7077	0,2873

Faster R-CNN 800

					test					val	val - confusion matrix				
	TP	FP	FN	TN	mAP@0.5-0.95	mAP@0.5	mAP small	mAP med	mAP large	mAP@0.5	accuracy	error	recall	precision	F1-score
SNPD	117	91	46	0	0,2058	0,4518	0,0154	0,1828	0,4343	0,7071	0,4606	0,5394	0,7178	0,5625	0,6307
Kitti	1664	665	1328	0						0,6154	0,4550	0,5450	0,5561	0,7145	0,6254
CityPed	724	578	2433	0						0,2714	0,1938	0,8062	0,2293	0,5561	0,3247

Faster R-CNN 2k

					test					val	val - confusion matrix				
	TP	FP	FN	TN	mAP@0.5-0.95	mAP@0.5	mAP small	mAP med	mAP large	mAP@0.5	accuracy	error	recall	precision	F1-score
SNPD	113	72	50	0	0,3027	0,5518	0,0970	0,2856	0,5418	0,6944	0,4809	0,5191	0,6933	0,6108	0,6494
Kitti	1892	553	1100	0						0,6862	0,5337	0,4663	0,6324	0,7738	0,6960
CityPed	792	518	2365	0						0,2725	0,2155	0,7845	0,2509	0,6046	0,3546

Faster R-CNN 5k

					test					val	val - confusion matrix				
	TP	FP	FN	TN	mAP@0.5-0.95	mAP@0.5	mAP small	mAP med	mAP large	mAP@0.5	accuracy	error	recall	precision	F1-score
SNPD	119	40	44	0	0,5101	0,7515	0,2981	0,5327	0,6959	0,7617	0,5862	0,4138	0,7301	0,7484	0,7391
Kitti	2733	64	259	0						0,9559	0,8943	0,1057	0,9134	0,9771	0,9442
CityPed	1137	765	2020	0						0,3699	0,2899	0,7101	0,3602	0,5978	0,4495

Faster R-CNN 5k HD

					test					val	val - confusion matrix				
	TP	FP	FN	TN	mAP@0.5-0.95	mAP@0.5	mAP small	mAP med	mAP large	mAP@0.5	accuracy	error	recall	precision	F1-score
SNPD	131	83	32	0	0,5398	0,7639	0,3489	0,5540	0,7166	0,7754	0,5325	0,4675	0,8037	0,6121	0,6950
Kitti	2834	22	158	0						0,9703	0,9403	0,0597	0,9472	0,9923	0,9692
CityPed	1213	876	1944	0						0,3909	0,3008	0,6992	0,3842	0,5807	0,4624

SSDLite 500

					test					val	val - confusion matrix				
	TP	FP	FN	TN	mAP@0.5-0.95	mAP@0.5	mAP small	mAP med	mAP large	mAP@0.5	accuracy	error	recall	precision	F1-score
SNPD	14	13	149	0	0,1277	0,3321	0,0308	0,1197	0,2932	0,1122	0,0795	0,9205	0,0859	0,5185	0,1474
Kitti	1039	786	1953	0						0,2826	0,2750	0,7250	0,3473	0,5693	0,4314
CityPed	59	110	3098	0						0,0246	0,0181	0,9819	0,0187	0,3491	0,0355

SSDLite 800

					test					val	val - confusion matrix				
	TP	FP	FN	TN	mAP@0.5-0.95	mAP@0.5	mAP small	mAP med	mAP large	mAP@0.5	accuracy	error	recall	precision	F1-score
SNPD	43	25	120	0	0,0646	0,1928	0,0063	0,0481	0,2140	0,3660	0,2287	0,7713	0,2638	0,6324	0,3723
Kitti	698	345	2294	0						0,3172	0,2092	0,7908	0,2333	0,6692	0,3460
CityPed	252	300	2905	0						0,0858	0,0729	0,9271	0,0798	0,4565	0,1359

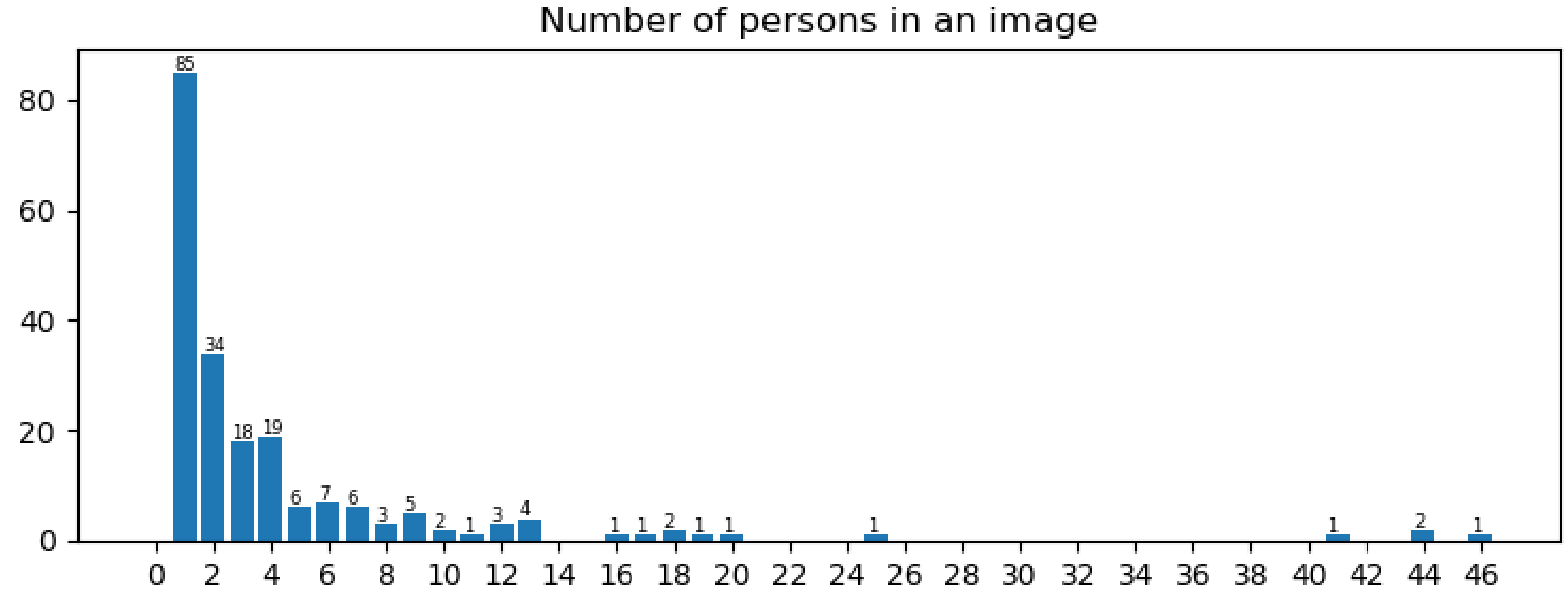
SSDLite 5k

					test					val	val - confusion matrix				
	TP	FP	FN	TN	mAP@0.5-0.95	mAP@0.5	mAP small	mAP med	mAP large	mAP@0.5	accuracy	error	recall	precision	F1-score
SNPD	31	12	132	0	0,0990	0,2812	0,0163	0,0632	0,3091	0,3730	0,1771	0,8229	0,1902	0,7209	0,3010
Kitti	879	367	2113	0						0,4185	0,2617	0,7383	0,2938	0,7055	0,4148
CityPed	501	384	2656	0						0,1599	0,1415	0,8585	0,1587	0,5661	0,2479

Training Phase Example

Measured parameters of training phase on limited dataset (204 images) for Faster R-CNN and SSD Lite:

Architecture	**Faster R-CNN**					**SSD Lite**	
Speed [FPS]	7.52					47.3	
Dataset volume	10	100	800	2000	5000	800	5000
mAP [%] **{small,medium,large}**	24.81 {1, 7, 19}	37.94 {1, 12, 35}	64.53 {8, 22, 48}	55.18 {10, 29, 54}	75.15 {30, 53, 70}	19.28 {1, 5, 21}	28.12 {2, 6, 31}
Learning time [min]	351	411	378	(677)	(4946)	737	(1692)

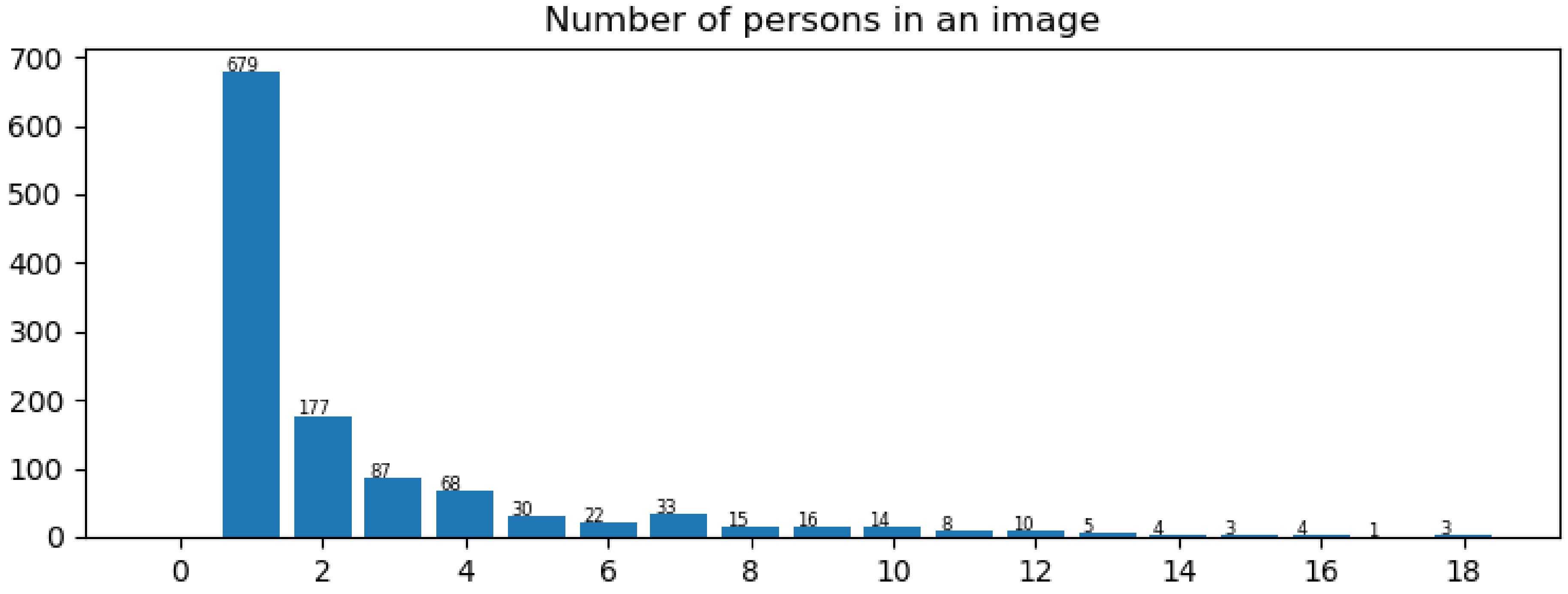


Images learned:	204
Pedestrians/image:	4.38
Average object width:	43.95 pxl
Average object height:	111.23 pxl
Small objects:	68
Medium objects:	420
Large objects:	405

Testing Phase – Kitti Dataset Validation

Now, we are interested much more in precision instead of in learning time:

	TP	FP	FN	TN	Precision	Recall	F1-score
F.R-CNN 800	1664	665	1328	0	0.71	0.56	0.63
F.R-CNN 5k	2733	64	259	0	**0.98**	**0.91**	**0.94**
SSD Lite 800	698	345	2294	0	0.67	0.23	0.35
SSD Lite 5k	501	384	2656	0	0.57	0.16	0.25

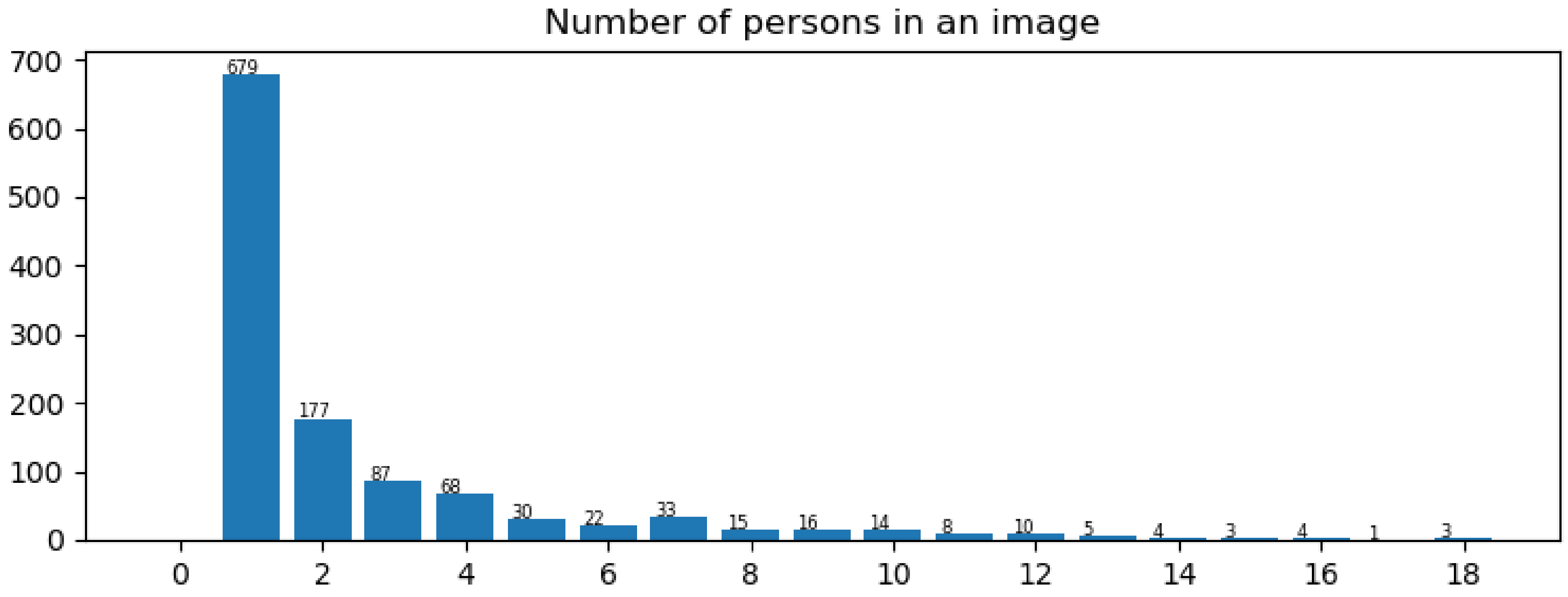


Images classified:	1179
Pedestrians/image:	2.54
Average object width:	43.66 pxl
Average object height:	103.21 pxl
Small objects:	142
Medium objects:	1495
Large objects:	1355

Testing Phase – CityPersons Dataset Validation

Now, we are interested much more in precision instead of in learning time:

	TP	FP	FN	TN	Precision	Recall	F1-score
F.R-CNN 800	724	578	2433	0	0.56	0.23	0.32
F.R-CNN 5k	1137	765	2020	0	0.60	0.36	0.45
SSD Lite 800	252	300	2905	0	0.46	0.08	0.14
SSD Lite 5k	501	284	2656	0	0.57	0.16	0.25



Images classified:	398
Pedestrians/image:	7.93
Average object width:	47.47 pxl
Average object height:	117.23 pxl
Small objects:	210
Medium objects:	1457
Large objects:	1490

Results Evaluation

Conclusions:

- **Mixed** training dataset (Kitti + Pascal VOC + CityPersons + Night Pedes) results in **lower false positive** detection in comparison with **single** Kitti dataset (video).
- Approx. 500 images are needed to train a model to **basic usable** level and 5k+ images for robust detection.
- Detection speed on laptop with low-end GPU Nvidia GeForce MX 150 v1/2 GB:

  a) Faster R-CNN **1.2 FPS**

  b) SSD Lite **12 FPS** (can be considered as real-time for ADAS)
- Dataset volume: 10, 100 and 500 are not enough => negligible mAP

Again, our goal was to detect pedestrians on images from on-board camera (Advanced driver-assistance systems)