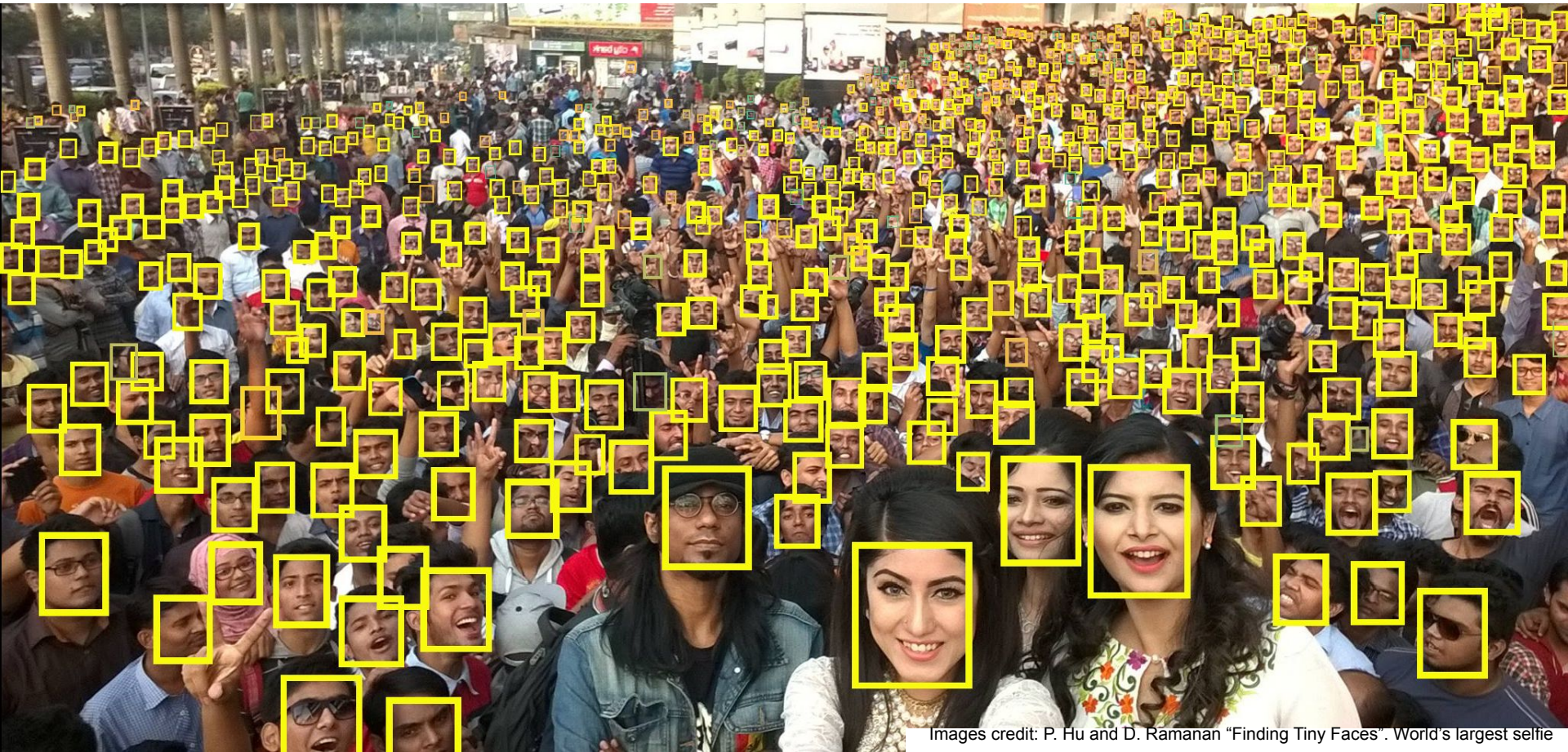
Xperience AI

Детектирование объектов: вчера, сегодня, завтра

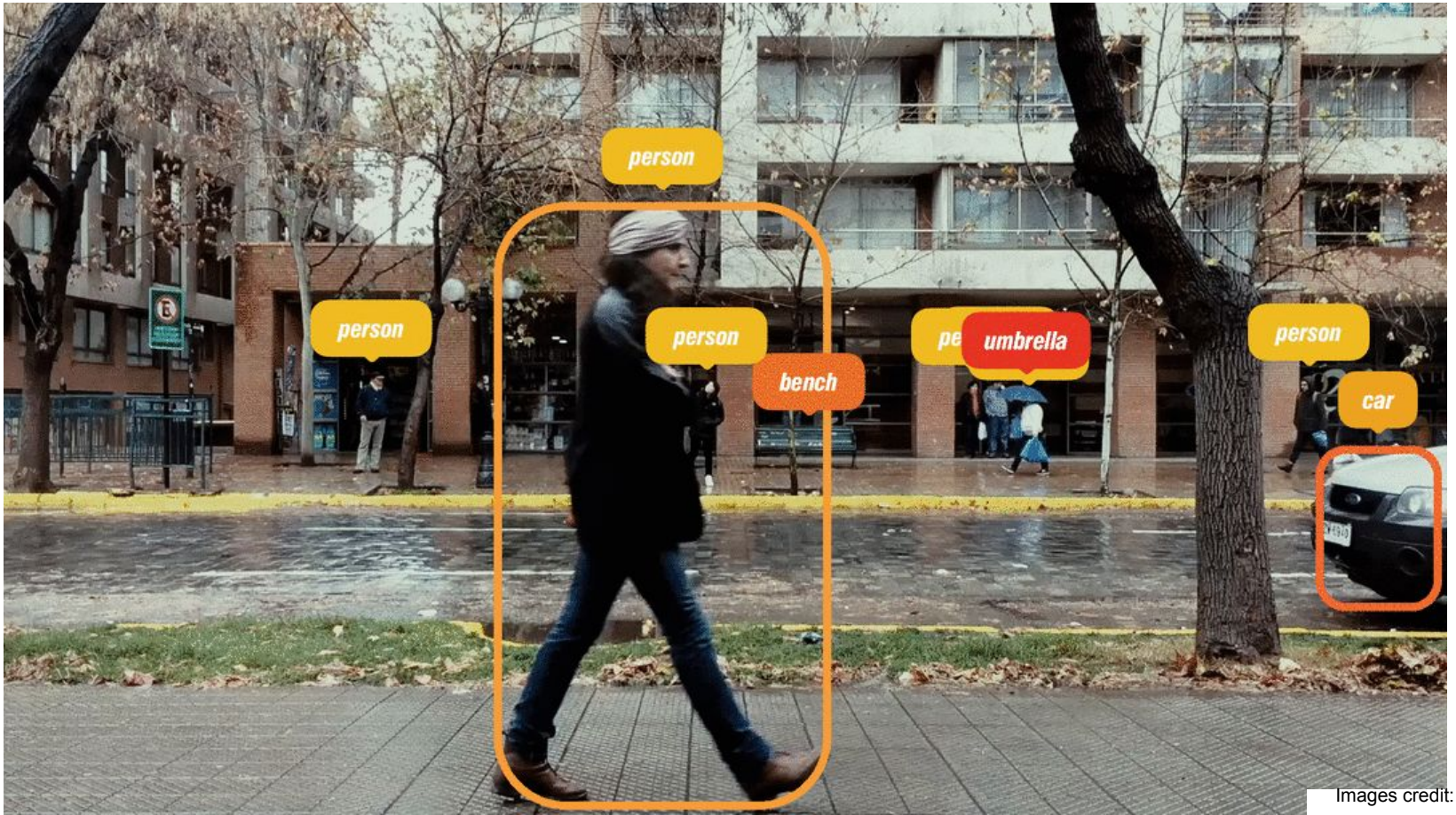
Даниил Осокин

Пример: детектирование лиц (один класс)



Images credit: P. Hu and D. Ramanan “Finding Tiny Faces”. World’s largest selfie

Пример: детектирование объектов (несколько классов)



Images credit: xperience.ai

Пример: 3D

Images credit: X. Chen et al. “Multi-View 3D Object Detection Network for Autonomous Driving”

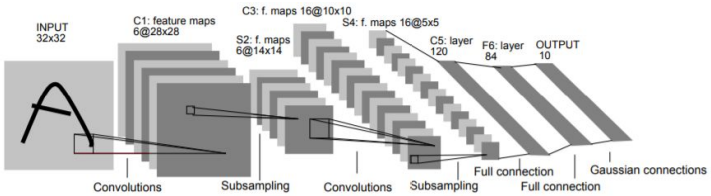
Детектирование Объектов

Локализация

- Порождение гипотез о местоположении объекта

Классификация

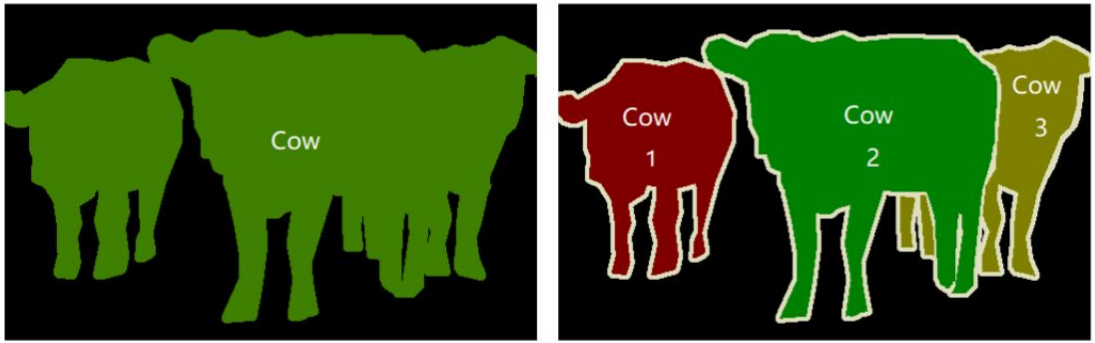
- Проверка гипотезы



Images credit: J.R.R. Uijlings et al. “Selective Search for Object Recognition”, Y. LeCun “Gradient-based learning applied to document recognition”

Задачи рядом

(a) Image Classification (b) Object Detection

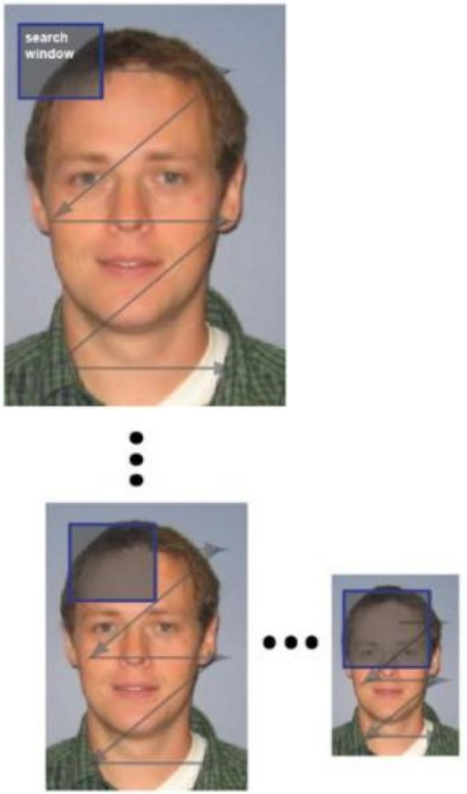


(c) Semantic Segmentation (d) Instance Segmentation

Images credit: X. Wu et al. “Recent Advances in Deep Learning for Object Detection”

Локализация: Скользящее Окно

Для нахождения координат объекта используется “*скользящее окно*”: классификатор применяется к каждому возможному расположению окна.



Для обнаружения объектов разного размера строится *пирамида изображений*.

Бинарная Классификация: Признаки

Тренировочная выборка: множество пар объект-метка: (x_i, y_i), $i=1..N$.

x_i - вектор признаков объекта (фичи) $\in R^n$, y_i - класс объекта $\in \{0,1\}$.

16x16 пикселей

x_i = (128, 128, 60, …, 0, 0), признаки - значения пикселей.

y_i = 1.

Вектор признаков объекта имеет один и тот же размер для всех объектов.

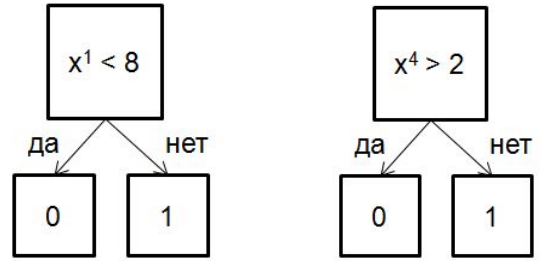
Бинарная Классификация: Дерево Решений

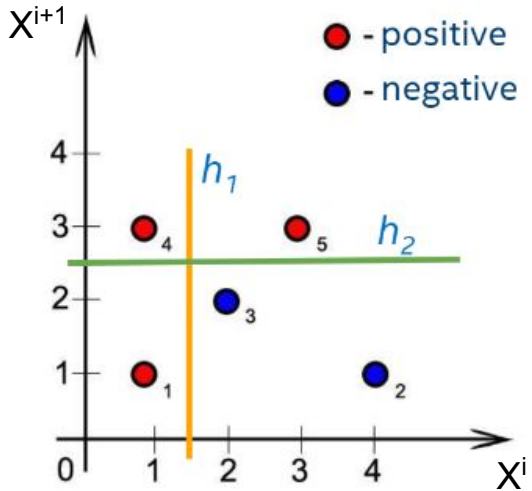
Разбивает пространство признаков по одной (нескольким) координатам.

Параметры разбиения (признак и порог) выбираются, чтобы минимизировать ошибку классификации:

$$\sum_{i=1}^{N} |label_{gt} - label_{predicted}|$$

Несколько “слабых” классификаторов комбинируются в один сильный (AdaBoost).

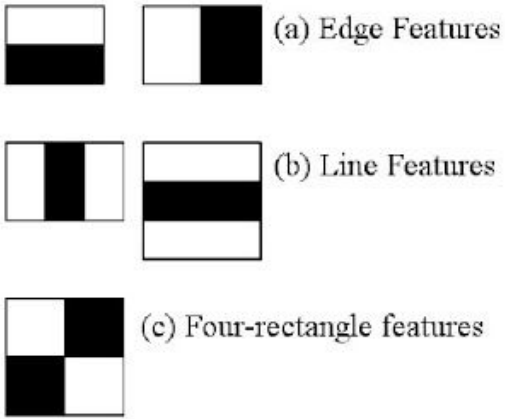




Признаки Хаара (2004)

Вектор признаков формируется из значений сверток с одним из заданных ядер, вычисленных в каждой точке изображения.

Ядра свертки (черный = -1, белый = 1):

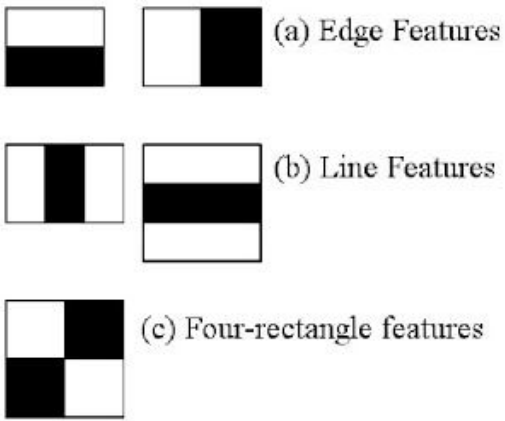


Images credit: https://www.youtube.com/watch?v=zLBAJ93-AEQ

Признаки Хаара (2004)

Вектор признаков формируется из значений сверток с одним из заданных ядер, вычисленных в каждой точке изображения.

Ядра свертки (черный = -1, белый = 1):



Images credit: https://www.youtube.com/watch?v=zLBAJ93-AEQ

Чем один признак лучше другого?

Хорошие признаки обладают инвариантностью:

- к освещению
- к масштабу
- к повороту

Популярные признаки:

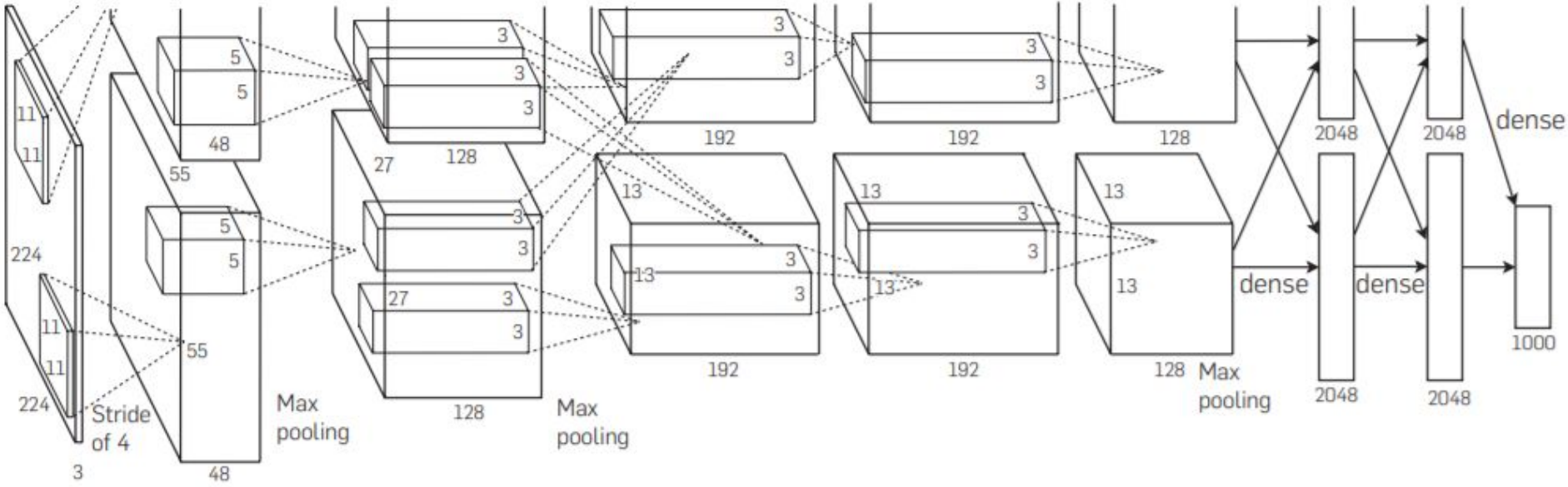
LBP: Local Binary Patterns (1994) - T. Ojala et al. “Performance evaluation of texture measures with classification based on Kullback discrimination of distributions”.

HOG: Histogram of Oriented Gradients (2005) - N. Dalal et al. “Histograms of Oriented Gradients for Human Detection”.

ICF: Integral Channel Features (2009) - P. Dollár et al. “Integral Channel Features”.

FCF: Filtered Channel Features (2015) - S. Zhang et al. “Filtered Channel Features for Pedestrian Detection”.

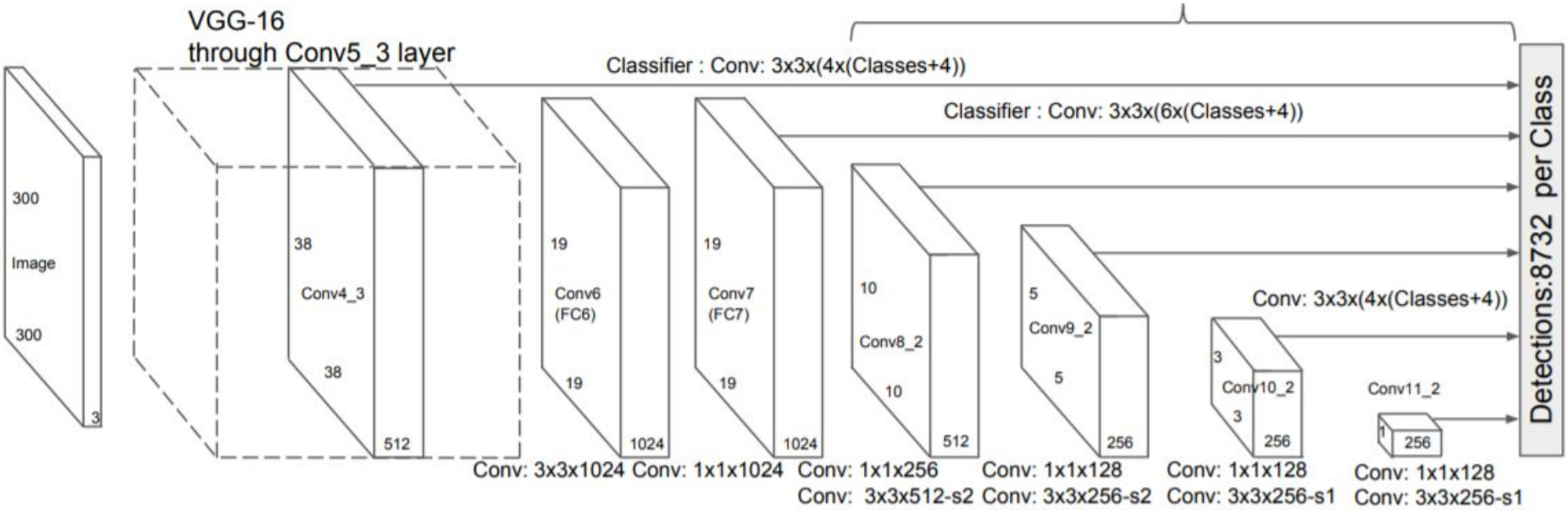
Чем поможет глубокое обучение?



Признаки обучаются сами, нужны только данные и оптимизируемая функция.

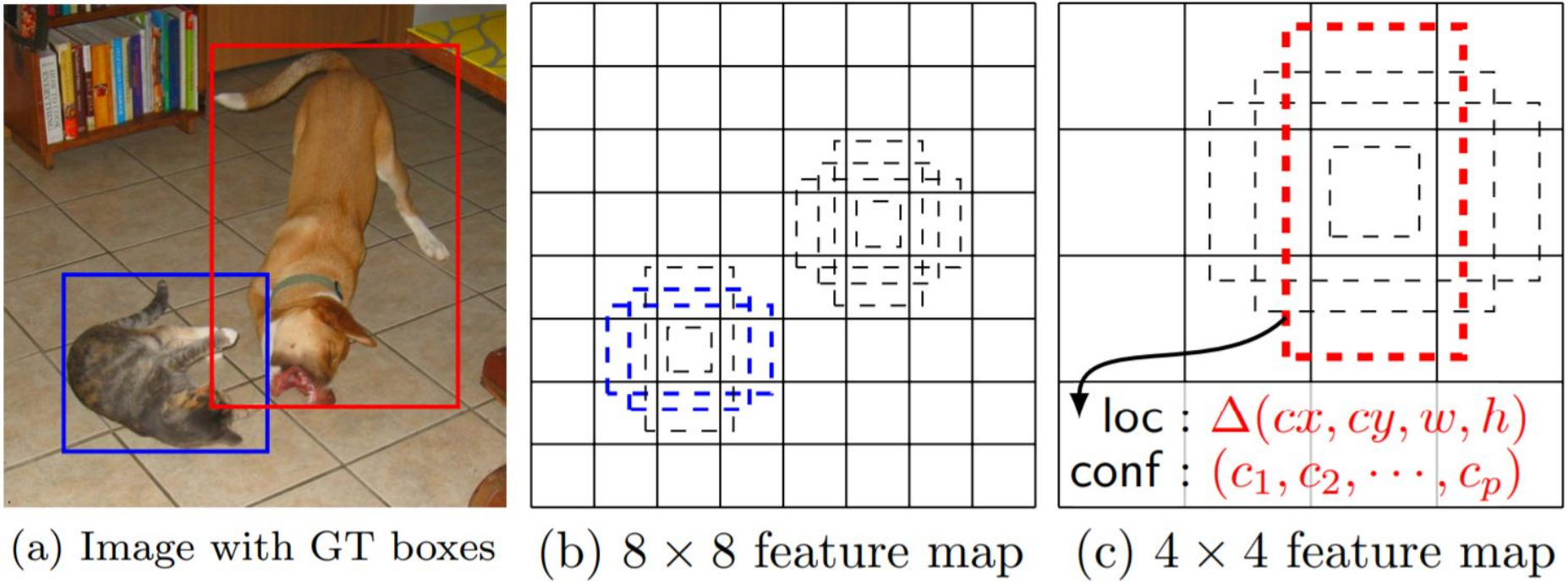
Images credit: A. Krizhevsky et al. “ImageNet classification with deep convolutional neural networks”

SSD: Single Shot MultiBox Detector. Пирамида



Images credit: W. Liu et al. “SSD: Single Shot MultiBox Detector”

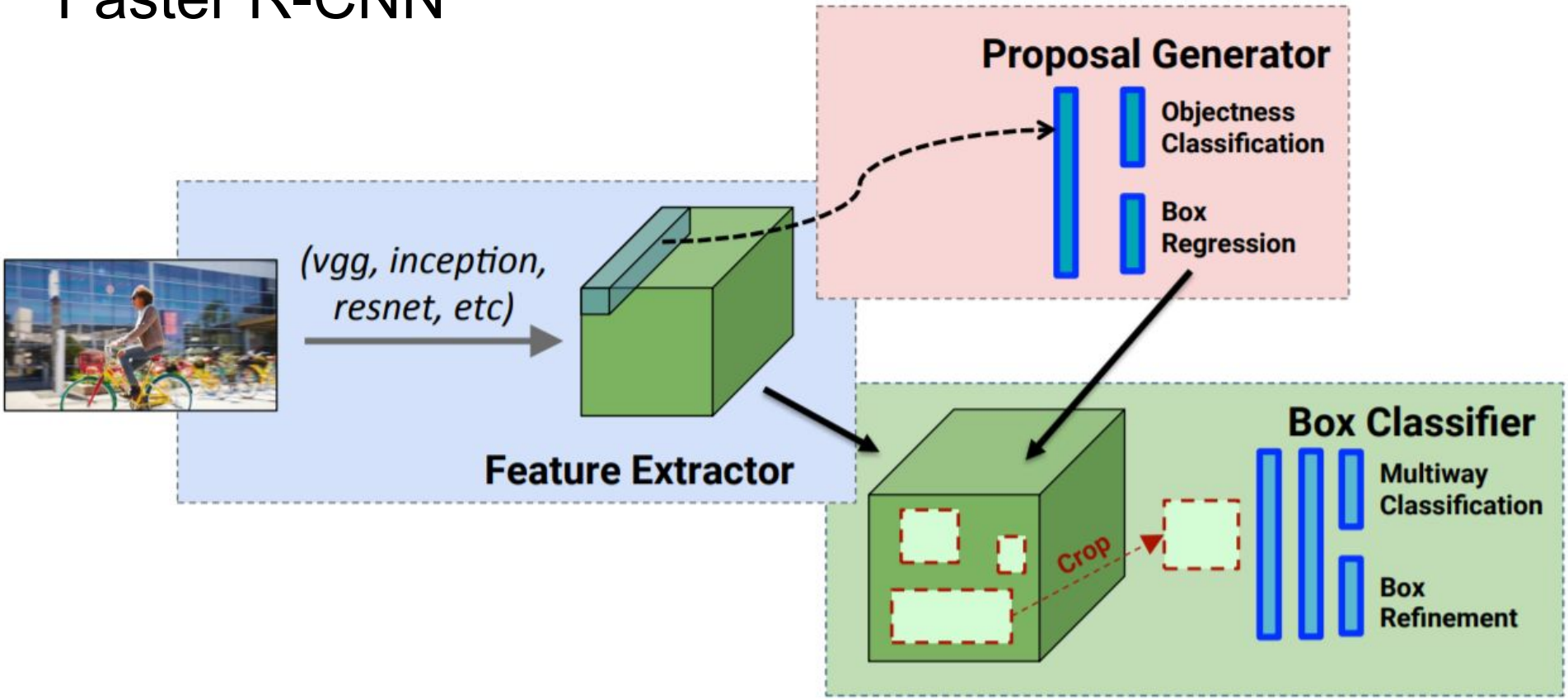
SSD: Single Shot MultiBox Detector. Скользящее Окно (Anchors).



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

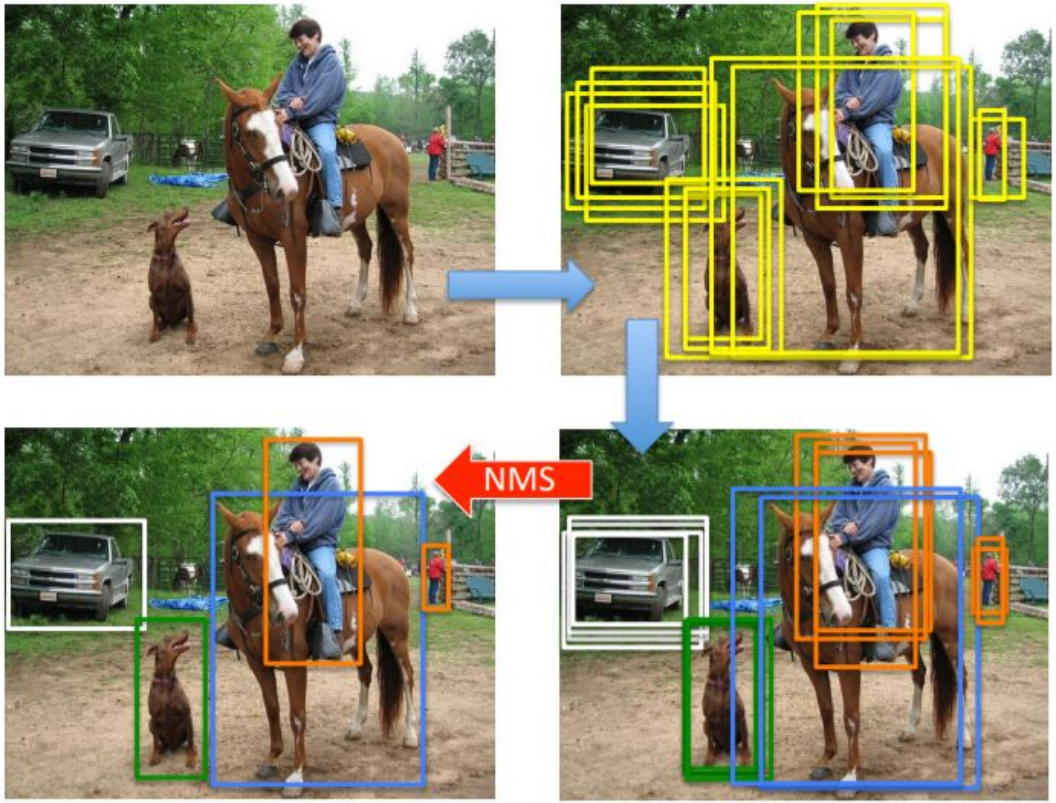
Images credit: W. Liu et al. “SSD: Single Shot MultiBox Detector”

Faster R-CNN



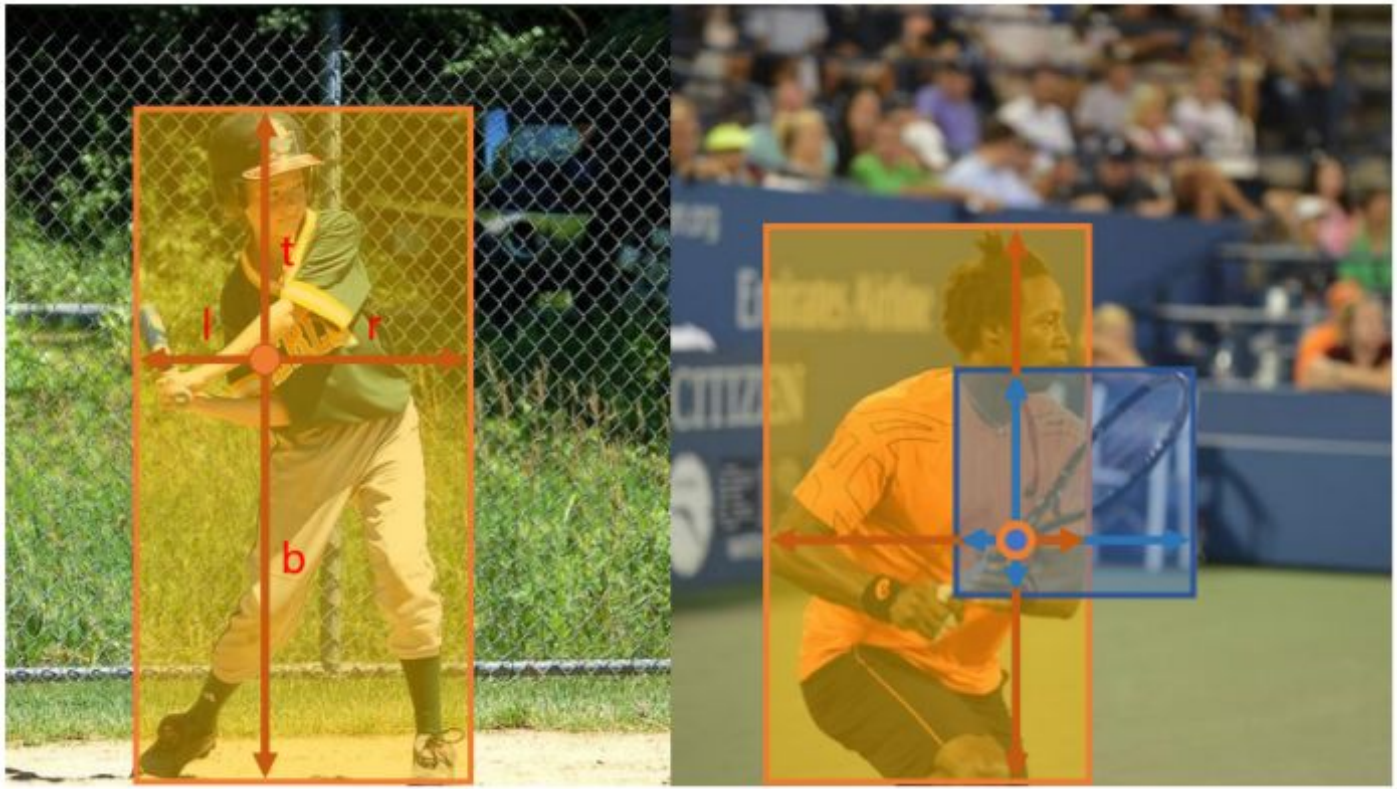
Images credit: S. Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, J. Huang et al. “Speed/accuracy trade-offs for modern convolutional object detectors”

NMS: Non-maximum Suppression



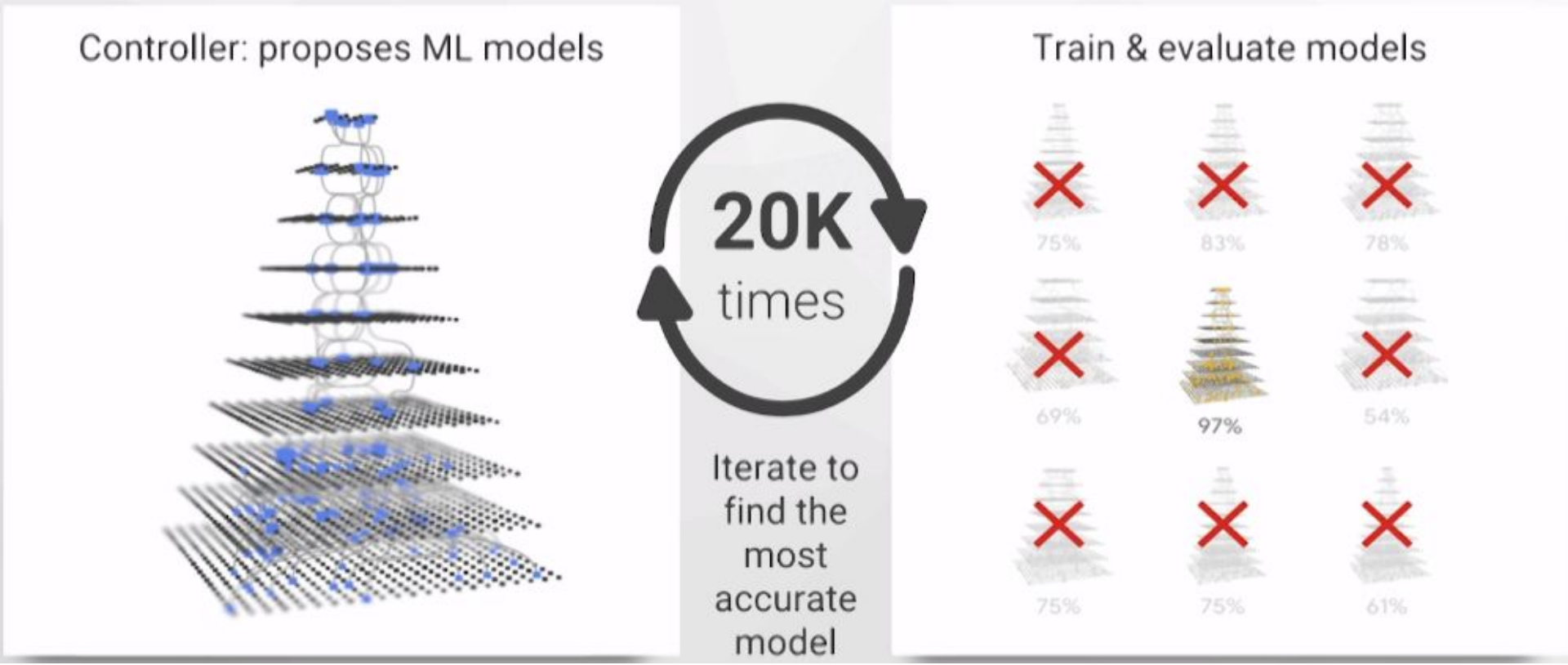
Images credit: N. Bodla et al. “Soft-NMS - Improving Object Detection With One Line of Code”

FCOS: Anchor-free Детектор



Images credit: Z. Tian et al. “FCOS: Fully Convolutional One-Stage Object Detection”

NAS: Neural Architecture Search



Images credit: https://www.youtube.com/watch?v=kSa3UObNS6o

NASNet

Model	image size	# parameters	Mult-Adds	Top 1 Acc. (%)	Top 5 Acc. (%)
Inception V2 [29]	224×224	11.2 M	1.94 B	74.8	92.2
NASNet-A (5 @ 1538)	**299×299**	**10.9 M**	**2.35 B**	**78.6**	**94.2**
Inception V3 [60]	299×299	23.8 M	5.72 B	78.8	94.4
Xception [9]	299×299	22.8 M	8.38 B	79.0	94.5
Inception ResNet V2 [58]	299×299	55.8 M	13.2 B	80.1	95.1
NASNet-A (7 @ 1920)	**299×299**	**22.6 M**	**4.93 B**	**80.8**	**95.3**
ResNeXt-101 (64 x 4d) [68]	320×320	83.6 M	31.5 B	80.9	95.6
PolyNet [69]	331×331	92 M	34.7 B	81.3	95.8
DPN-131 [8]	320×320	79.5 M	32.0 B	81.5	95.8
SENet [25]	**320×320**	**145.8 M**	**42.3 B**	**82.7**	**96.2**
NASNet-A (6 @ 4032)	**331×331**	**88.9 M**	**23.8 B**	**82.7**	**96.2**

Images credit: B. Zoph et al. “Learning Transferable Architectures for Scalable Image Recognition”

NASNet

Model	image size	# parameters	Mult-Adds	Top 1 Acc. (%)	Top 5 Acc. (%)
Inception V2 [29]	224×224	11.2 M	1.94 B	74.8	92.2
NASNet-A (5 @ 1538)	**299×299**	**10.9 M**	**2.35 B**	**78.6**	**94.2**
Inception V3 [60]	299×299	23.8 M	5.72 B	78.8	94.4
Xception [9]	299×299	22.8 M	8.38 B	79.0	94.5
Inception ResNet V2 [58]	299×299	55.8 M	13.2 B	80.1	95.1
NASNet-A (7 @ 1920)	**299×299**	**22.6 M**	**4.93 B**	**80.8**	**95.3**
ResNeXt-101 (64 x 4d) [68]	320×320	83.6 M	31.5 B	80.9	95.6
PolyNet [69]	331×331	92 M	34.7 B	81.3	95.8
DPN-131 [8]	320×320	79.5 M	32.0 B	81.5	95.8
SENet [25]	**320×320**	**145.8 M**	**42.3 B**	**82.7**	**96.2**
NASNet-A (6 @ 4032)	**331×331**	**88.9 M**	**23.8 B**	**82.7**	**96.2**

[1] In particular, we note that previous architecture search [71] used 800 GPUs for 28 days resulting in 22,400 GPU-hours. The method in this paper uses 500 GPUs across 4 days resulting in 2,000 GPU-hours. The former effort used Nvidia K40 GPUs, whereas the current efforts used faster NVidia P100s. Discounting the fact that the we use faster hardware, we estimate that the current procedure is roughly about 7× more efficient.

Images credit: B. Zoph et al. “Learning Transferable Architectures for Scalable Image Recognition”

Резюме

Детектирование объектов:

- Локализация (скользящее окно)
- Классификация (деревья решений)

Признаки Хаара

Популярные архитектуры с автоматически обучаемыми признаками:

- SSD, Faster R-CNN

Q&A

daniil.osokin@xperience.ai