



Введение в компьютерное зрение

19 ноября 2019, ННГУ

Сергей Носов, руководитель отдела разработки алгоритмов

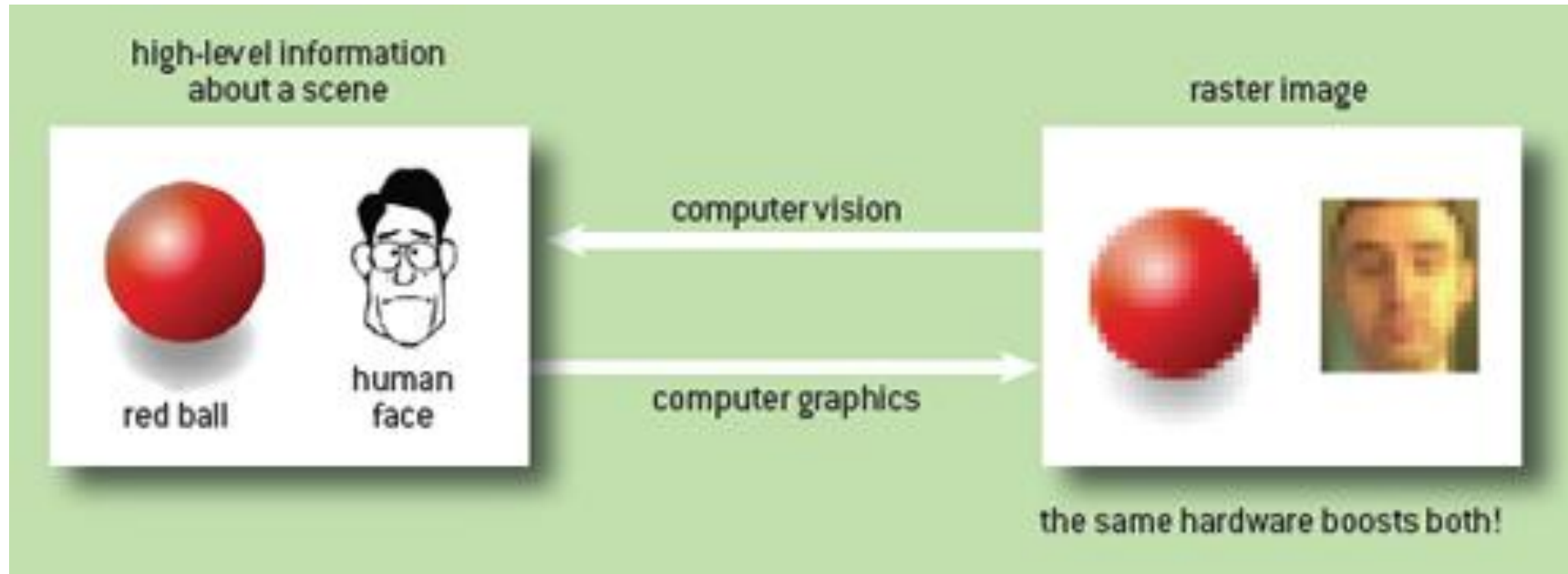
IOTG Computer Vision (ICV), Intel

Internet of Things Group

Обзор

- Решаемые задачи
- Современные бенчмарки
- Работа современного прикладного исследователя

Компьютерное зрение и Компьютерная графика



<https://queue.acm.org/detail.cfm?id=2206309>

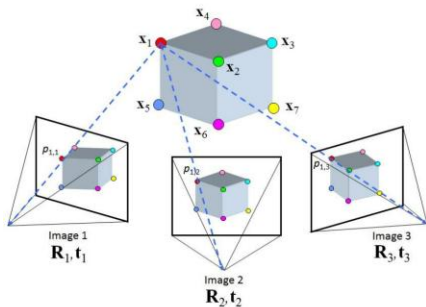
Компьютерное зрение

Анализ
движения

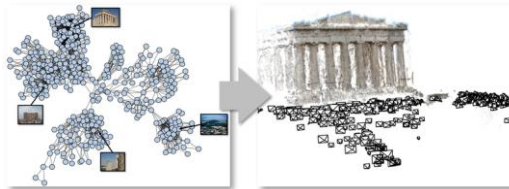
Трёхмерная
Реконструкция

Распознавание

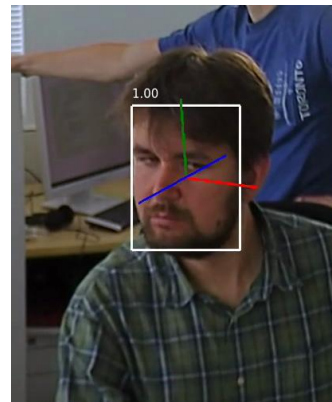
Восстановление
изображений



https://www.researchgate.net/publication/269327935_Stereo_and_kinect_fusion_for_continuous_3D_reconstruction_and_visual_odometry



<http://www.cs.cornell.edu/projects/bigsfm/>



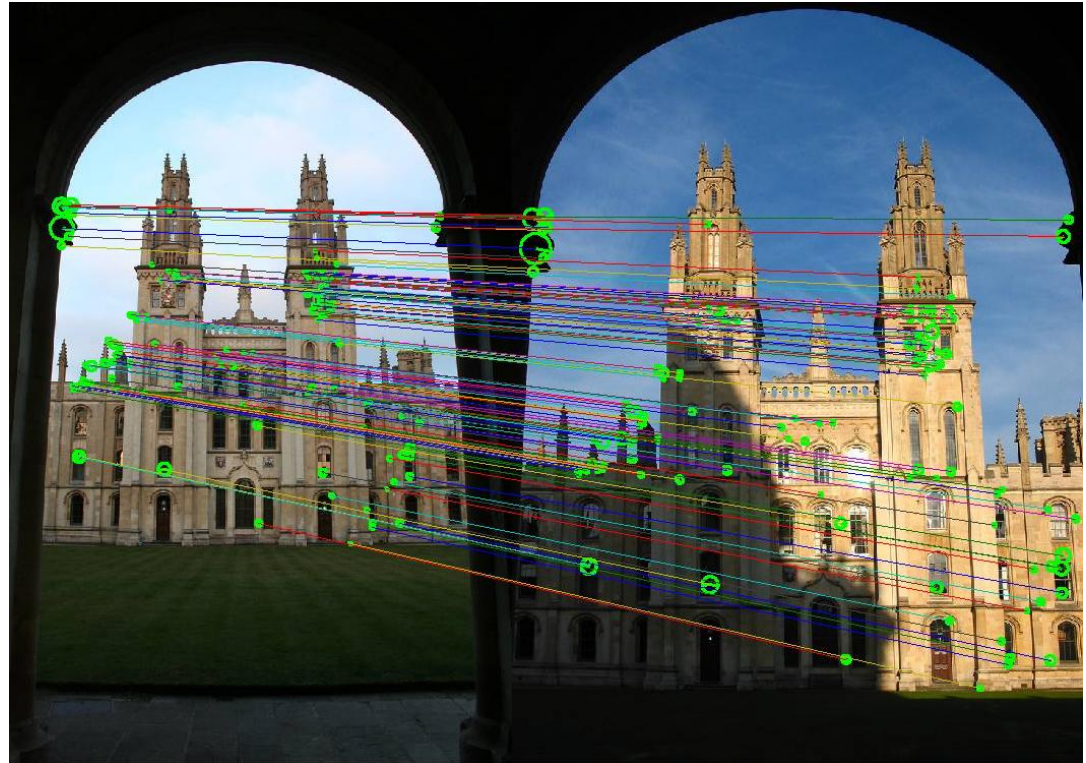
https://github.com/opencv/open_model_zoo



<https://towardsdatascience.com/how-to-perform-image-restoration-absolutely-dataset-free-d08da1a1e96d>

Ключевые точки

- Поиск и установление соответствия между ключевыми точками – одна из фундаментальных задач компьютерного зрения
- Несложные геометрические соображения позволяют в определённой степени решать задачи одометрии, трекинга, 3х-мерной реконструкции и др.



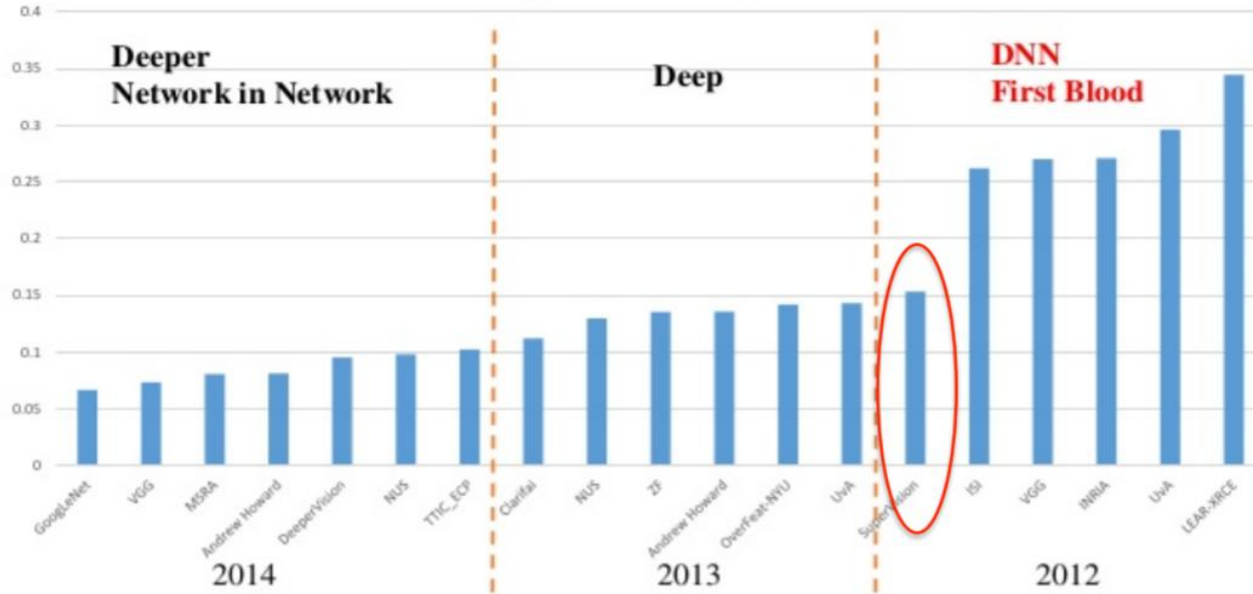
<http://www.robots.ox.ac.uk/~vgg/practicals/instance-recognition/index.html>

Поиск ключевых точек

- Фундаментально, нужно найти
 - «интересные» точки, которые будут **устойчиво** находиться на разных изображениях одной и той же сцены, и
 - их дескрипторы, которые будут, в некотором смысле, **уникальны**.
- Интуитивно, нужно искать углы, «мелкие» особенности и т.п. В простейшем случае можно опираться на величину градиента в точках изображения.
- В качестве дескриптора могут использоваться гистограммы локальных градиентов (HOG). Исторически, наиболее известны SIFT, SURF, ORB и др.

ILSVRC

ImageNet Classification error throughout years and groups



Li Fei-Fei: ImageNet Large Scale Visual Recognition Challenge, 2014 <http://image-net.org/>

http://vision.stanford.edu/teaching/cs231b_spring1415/slides/alexnet_tugce_kyunghee.pdf

***: The most criminally
underused tool in the potential
machine learning toolbox?

(Заметка за 2009 год)

Automatic Differentiation: The most criminally underused tool in the potential machine learning toolbox?

Update: (November 2015) In the almost seven years since writing this, there has been an explosion of great tools for automatic differentiation and a corresponding upsurge in its use. Thus, happily, this post is more or less obsolete.

I recently got back reviews of a paper in which I used [automatic differentiation](#). Therein, a reviewer clearly thought I was using finite difference, or “numerical” differentiation. This has led me to wondering: **Why don’t machine learning people use automatic differentiation more? Why don’t they use it...constantly?** Before recklessly speculating on the answer, let me briefly review what automatic differentiation (henceforth “autodiff”) is. Specifically, I will be talking about **reverse-mode autodiff**.

<https://justindomke.wordpress.com/2009/02/17/automatic-differentiation-the-most-criminally-underused-tool-in-the-potential-machine-learning-toolbox/>

Автодифференцирование: взгляд алгебраиста

Построим гиперкомплексные числа, заменив вещественный x на

$x + x'\varepsilon$, где $\varepsilon^2 = 0$. Тогда в этой алгебре справедливы выражения:

$$\langle u, u' \rangle + \langle v, v' \rangle = \langle u + v, u' + v' \rangle$$

$$\langle u, u' \rangle - \langle v, v' \rangle = \langle u - v, u' - v' \rangle$$

$$\langle u, u' \rangle * \langle v, v' \rangle = \langle uv, u'v + uv' \rangle$$

$$\langle u, u' \rangle / \langle v, v' \rangle = \left\langle \frac{u}{v}, \frac{u'v - uv'}{v^2} \right\rangle \quad (v \neq 0)$$

Что же произошло в 2012 году?

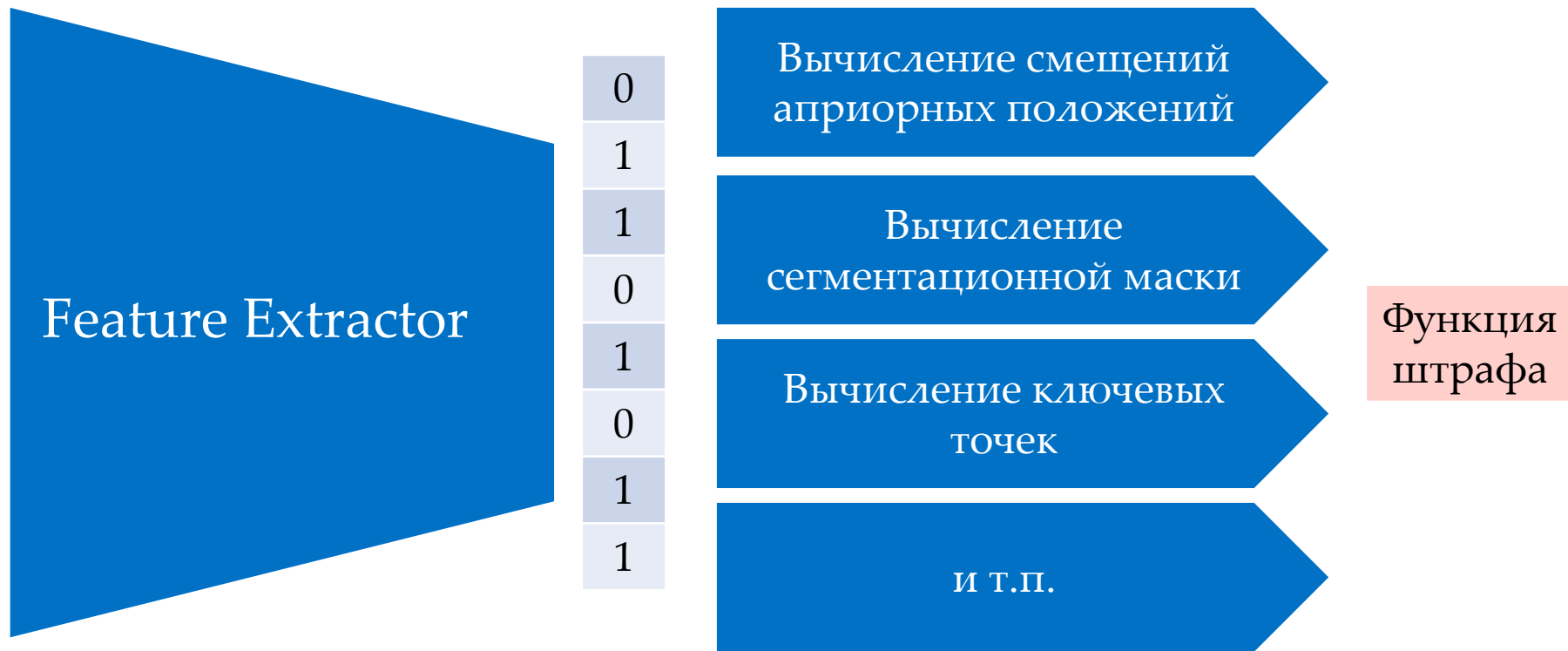
Критическая масса исследователей в области машинного обучения и, в частности, компьютерного зрения, вдохновлённая успехом Алекса Крижевского, взяла на вооружение современный подход к решению задач:

1. Для формализации некорректно-поставленных задач нужно использовать большие датасеты с надёжной разметкой
2. Функцию штрафа нужно строить из «простых» функций при помощи «простых» операторов
3. Бояться размера датасета и сложности функции не надо, потому что при помощи автодифференцирования и современного железа (GPU) её можно эффективно оптимизировать.

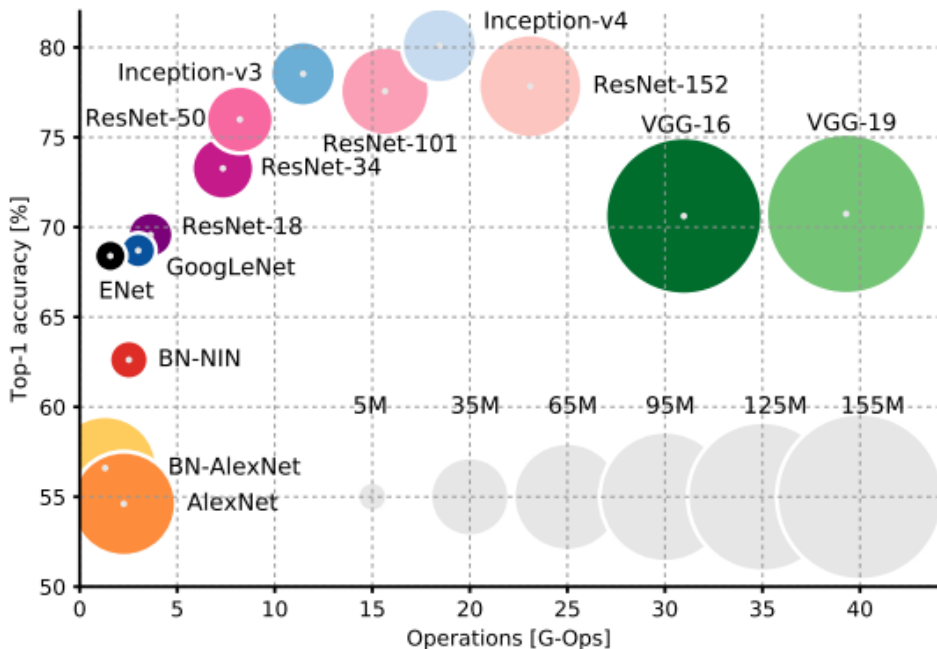
Задачи распознавания

- Классификация – image -> label (кошка, собака, кружка, банан, и т.д.)
- Идентификация – image -> embedding (unique)
 - Лиц, QR-кодов, отпечатков пальцев, людей и т.д.
- Детектирование: image -> bounding boxes
- Сегментация: image -> (instance) segmentation mask
- Ключевые точки: image -> key points
 - Общего назначения, человека, животного и др.
- Распознавание текста: image -> text

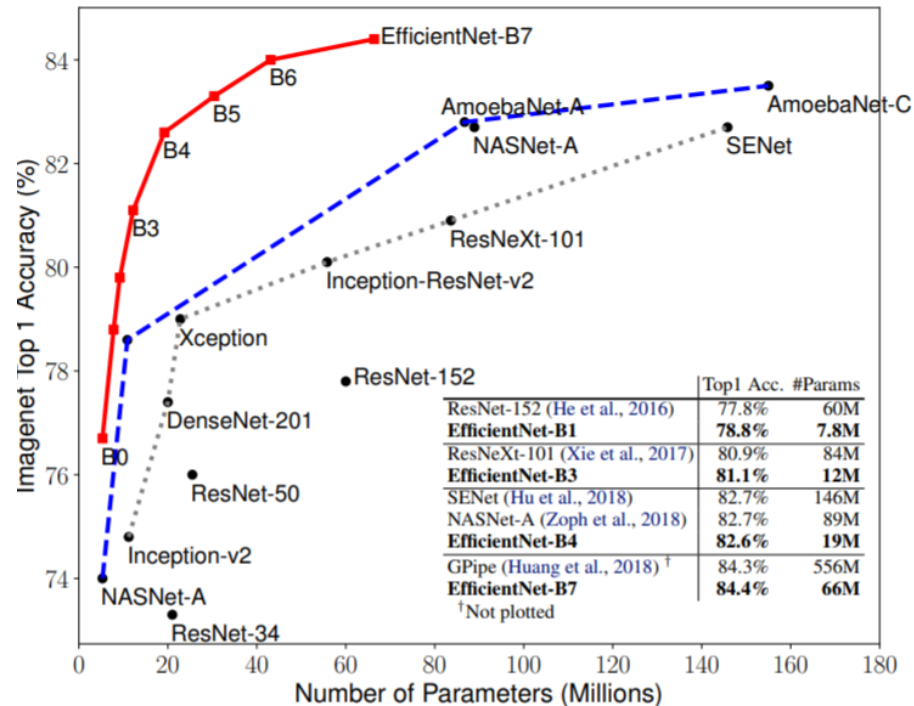
Архитектура сети в задачах распознавания



Поиск лучших решений



<https://arxiv.org/pdf/1605.07678.pdf>



<https://arxiv.org/pdf/1905.11946.pdf>

Современные бенчмарки

- MS COCO <http://cocodataset.org/> ~200K размеченных изображений
 - Детектирование объектов (80 классов)
 - Детектирование ключевых точек человека (19 точек)
 - Семантическая сегментация (things & stuff)
- ImageNet <http://image-net.org/> ~1M размеченных изображений
 - Классификация изображений (1000 классов)
- KITTI <http://www.cvlibs.net/datasets/kitti/>
 - Большое количество бенчмарков для automotive алгоритмов

Современные бенчмарки

- Детектирование лиц
 - WIDER FACE <http://shuoyang1213.me/WIDERFACE/> ~32К изображений
 - FDDB <http://vis-www.cs.umass.edu/fddb/> ~3К изображений
- Распознавание лиц
 - MegaFace <http://megaface.cs.washington.edu/> ~5М изображений
 - FRVT NIST <https://www.nist.gov/programs-projects/face-recognition-vendor-test-frvt-ongoing>
 - LFW <http://vis-www.cs.umass.edu/lfw/> ~13К изображений







Detection Leaderboard

BBOX: [Dev](#) [Standard15](#) [Chal15](#) [Chal16](#) [Chal17](#)

SEGM: [Dev](#) [Standard15](#) [Chal15](#) [Chal16](#) [Chal17](#) [Chal18](#)

[Copy to Clipboard](#)
[Export to CSV](#)

Search:

	AP	AP ⁵⁰	AP ⁷⁵	AP ^S	AP ^M	AP ^L	AR ¹	AR ¹⁰	AR ¹⁰⁰	AR ^S	AR ^M	AR ^L	date
 Megvii (Face++)	0.526	0.730	0.585	0.343	0.556	0.660	0.391	0.645	0.689	0.513	0.727	0.827	2017-10-05
 CM-CV AR	0.525	0.717	0.578	0.352	0.550	0.642	0.392	0.647	0.689	0.528	0.720	0.821	2019-07-26
 Night owl	0.519	0.704	0.570	0.342	0.548	0.647	0.391	0.640	0.680	0.498	0.713	0.824	2019-08-05
 Alibaba Turing Lab	0.514	0.694	0.563	0.336	0.540	0.639	0.388	0.638	0.679	0.499	0.711	0.818	2019-08-05
 UCenter	0.510	0.705	0.558	0.326	0.539	0.648	0.392	0.640	0.678	0.497	0.720	0.829	2017-10-05
 MSRA	0.507	0.717	0.566	0.343	0.529	0.627	0.379	0.638	0.690	0.524	0.720	0.824	2017-10-05

AP AP⁵⁰ AP⁷⁵ AP^S AP^M AP^L AR¹ AR¹⁰ AR¹⁰⁰ AR^S AR^M AR^L date

 Megvii (Face++)	0.526	0.730	0.585	0.343	0.556	0.660	0.391	0.645	0.689	0.513	0.727	0.827	2017-10-05
---	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	------------

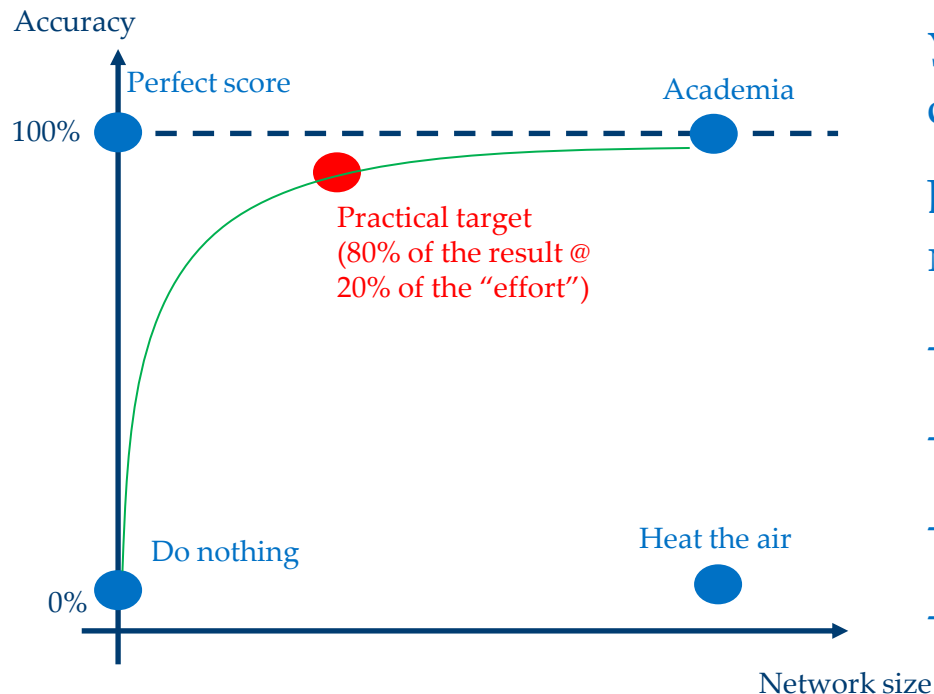
Team: Chao Peng*, Tete Xiao*, Zeming Li*, Yuning Jiang, Xiangyu Zhang, Kai Jia, Gang Yu, Jian Sun (* indicates equal contribution); Megvii Research

Description: We trained a large-batch object detector, MegDet, by Megvii (Face++)'s large-scale deep learning framework called MegBrain, in parallel, on 128 GPUs. Thanks to MegBrain, we were able to finish the whole training within ~1 day and got improved performance due to the large batch size. The design of our detectors follows the idea of FPN[1], whose feature extractors are a series of ReNeXt-like models pre-trained on ImageNet only. GCN modules[2] and instance-blind segmentation supervision[3] were also applied in the detector. We did not use the unlabeled images provided by MSCOCO or other training data. On the test-dev, our best single detector had obtained mAP 50.5, and the ensemble of four detectors had achieved mAP 52.6. [1] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, et al. Feature pyramid networks for object detection. CVPR 2017. [2] Chao Peng, Xiangyu Zhang, Gang Yu, et al. Large Kernel Matters--Improve Semantic Segmentation by Global Convolutional Network. CVPR 2017. [3] Jiayuan Mao, Tete Xiao, Yuning Jiang, et al. What Can Help Pedestrian Detection? CVPR 2017.

Link:

More: [Per Category Results](#)

«Реальная» сложность задачи

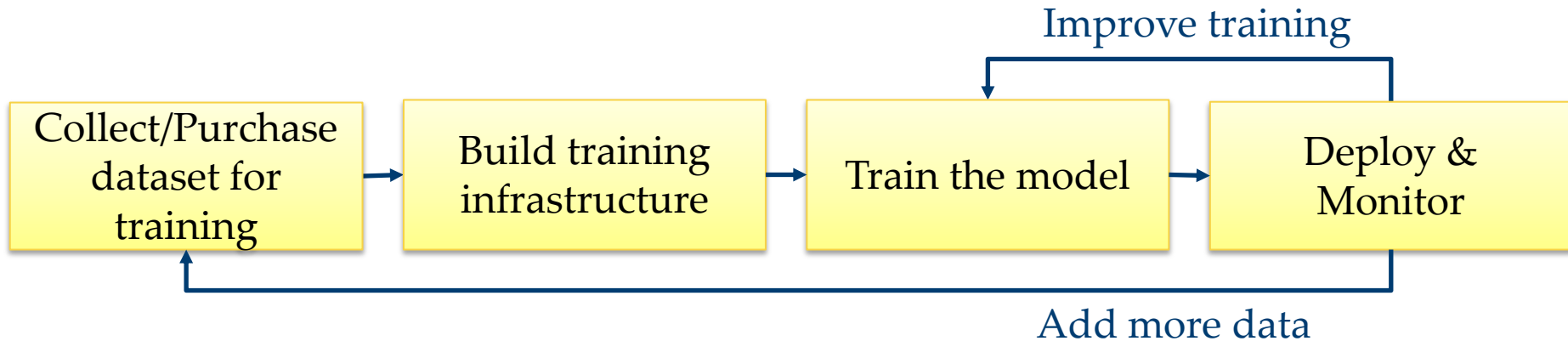


У любой вычислительной задачи есть своя «реальная» сложность.

В детектировании объектов на текущий момент существуют такие оценки:

- Регистрационный номер: 0.35 GFLOPs
- Автомобиль: 0.5-1.0 GFLOPs
- Лицо: 1.0-2.0 GFLOPs
- Человек: >3.5 GFLOPs

R&D Lifecycle in Numbers



- \$0.01-30 per image
- \$1-100 per hour
- \$0.03-1 per annotated object

- 10-100 computational nodes
- 100-1000 GBs of storage and network transfers

- 1-10 Data Scientists
- 3-12 months

Литература

- Richard Szeliski, “Computer Vision: Algorithms and Applications”, 2010, <http://szeliski.org/Book/>
- <http://opencv.org/>
- <https://paperswithcode.com/>
- Ian Goodfellow, Yoshua Bengio, Aaron Courville, “Deep Learning”, 2016, <http://www.deeplearningbook.org/>



<https://en.wikipedia.org/wiki/StyleGAN>

