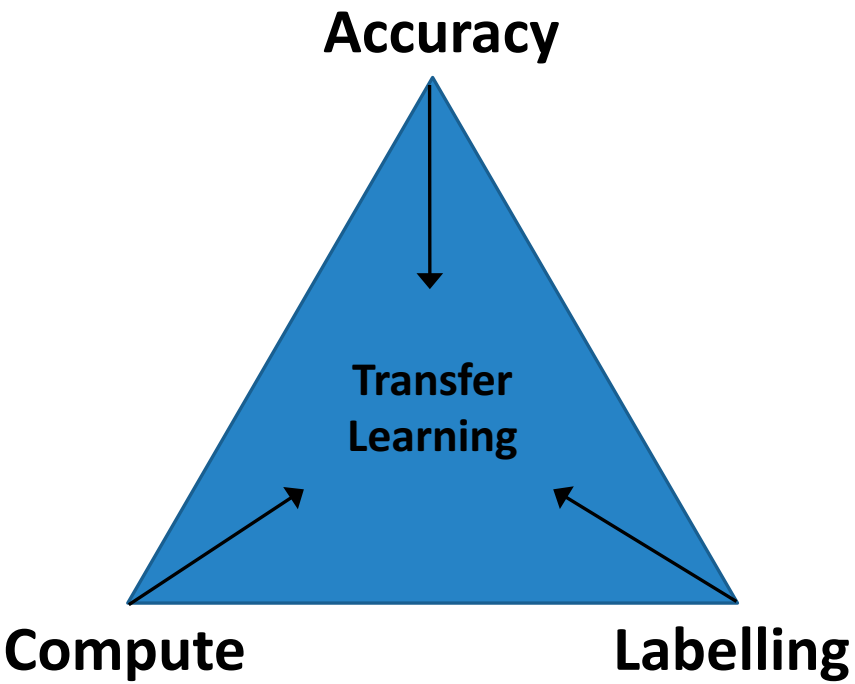


Transfer Learning

FOR FUN AND PROFIT

09.01.2018

Alexander Hirner
@cybertreiber

MoonVision.io
REALTIME OBJECT TRACKING

Intro

Problem

- Manual order control
  - Highly trusted individuals
  - Fatigued after 2h:
    (rotate or assume error rate)
  - “The 2nd most important people”
- Limited throughput

Conceived on a napkin in Lappland

DishTracker automates food checkout

Product

Stress tested at Oktoberfest 2017

- 98.2% agreement for frequent dishes

- >20 types of dishes
- 28h of sample video (first two days)
- 1 day later in operation

- < 1 sec. latency from network source to marked-up video stream
- 3x20fps with 20% GPU load

- High resonance from industry

Content

Stress tested at Oktoberfest 2017

1. What are the challenges

2. How to utilize transfer learning for a subset of these challenges

3. Wrap up and outlook

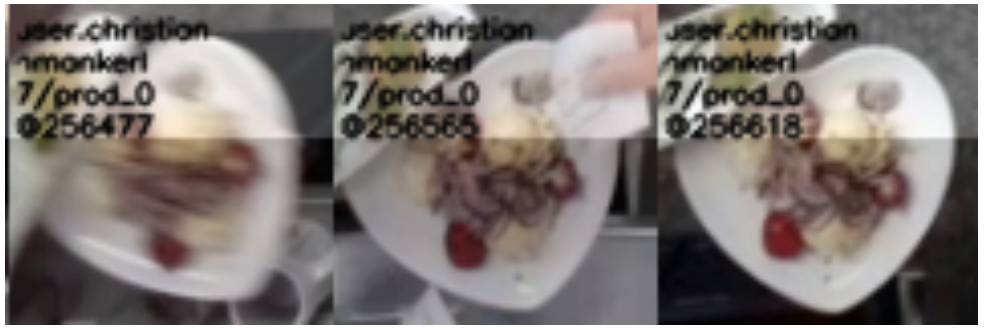
Core Challenges

From Video to Detection:

- Annotation time
- Label quality, taxonomy, completeness
- Class inbalance (fat-tail)

From Detection to Realtime Tracking:

- Blur
- Occlusion
- Noisy detections
- Compute Time [cf. “Tublets” approach Wang, CVPR 2017 https://youtu.be/pK6XAk95kUY?t=35m40s]



Architecture

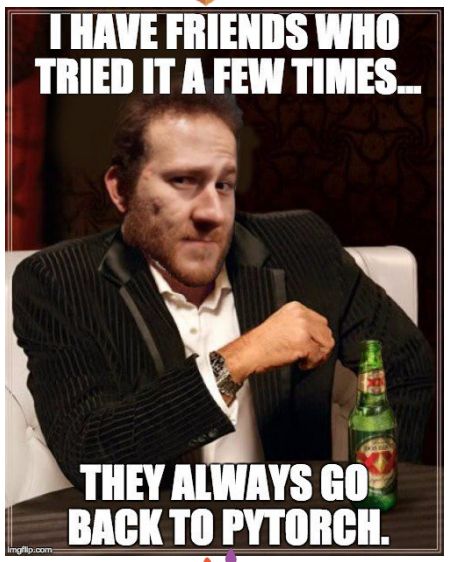
Stream

both

Offline

Data..

[embedded systems cam]



Flask

[Apache] **Superset**

Acquisition
40%

Science
20%

Management and Views
40%

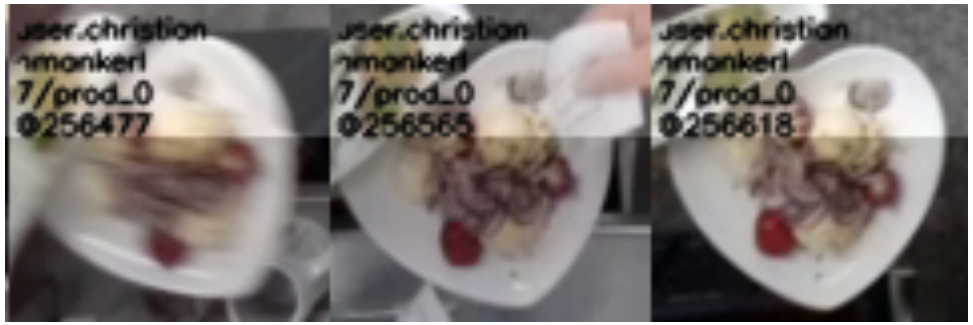
Core Solutions

From Video to Detection:

- Scene/Shot Extraction that maximizes pose variance
- Automated Labelling Tool:
  - Region Proposals
  - Label Proposal
  - Multi-tenant collaboration
- Training Strategy for incomplete and noisy labels

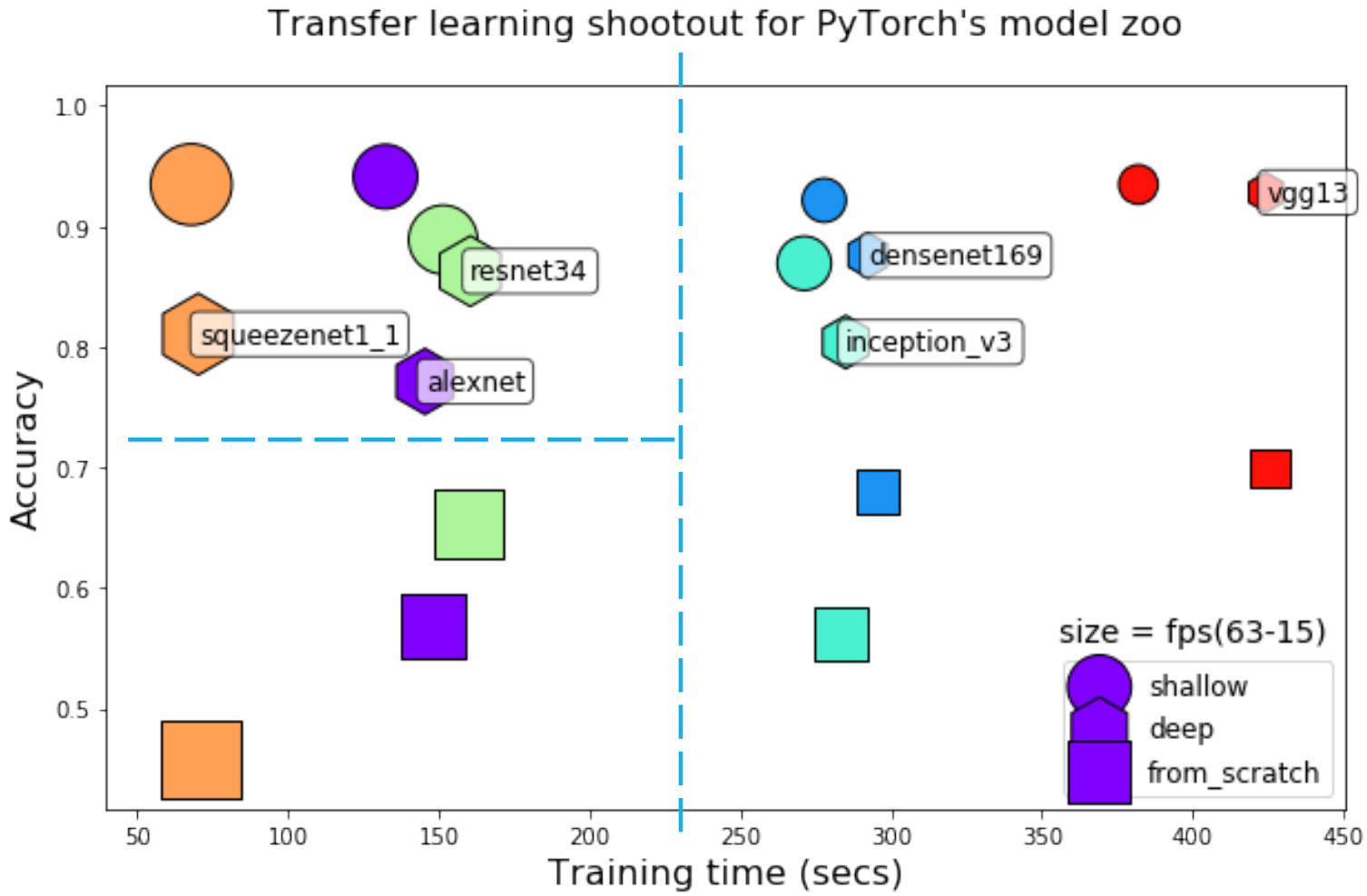
From Detection to Tracking in Realtime:

- Occlusion logic
- Aggregate state over object life-time
- Fusion with physical model, motion-flow



Label Proposals: model pre-selection

Toy Dataset: two classes, scale variance

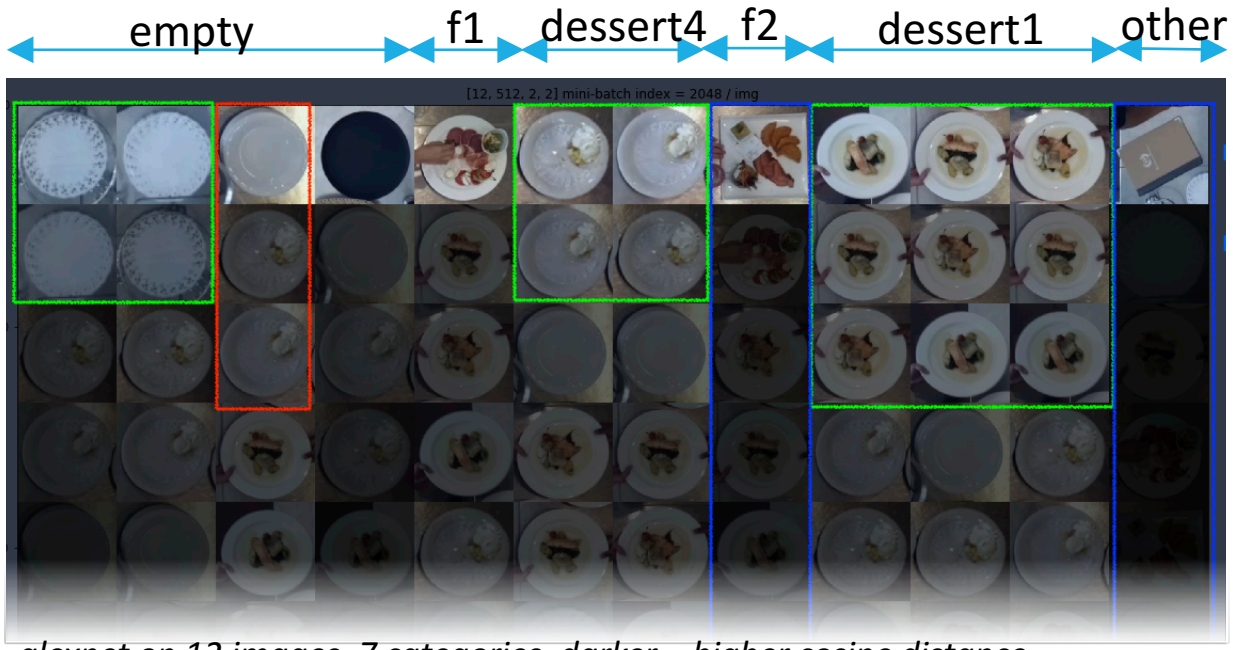


squeezenet, alexnet, (resnet34):

- Robust to retrain
- Quick to retrain
- Computationally feasible

Embedding Quality

Example



alexnet on 12 images, 7 categories, darker = higher cosine distance

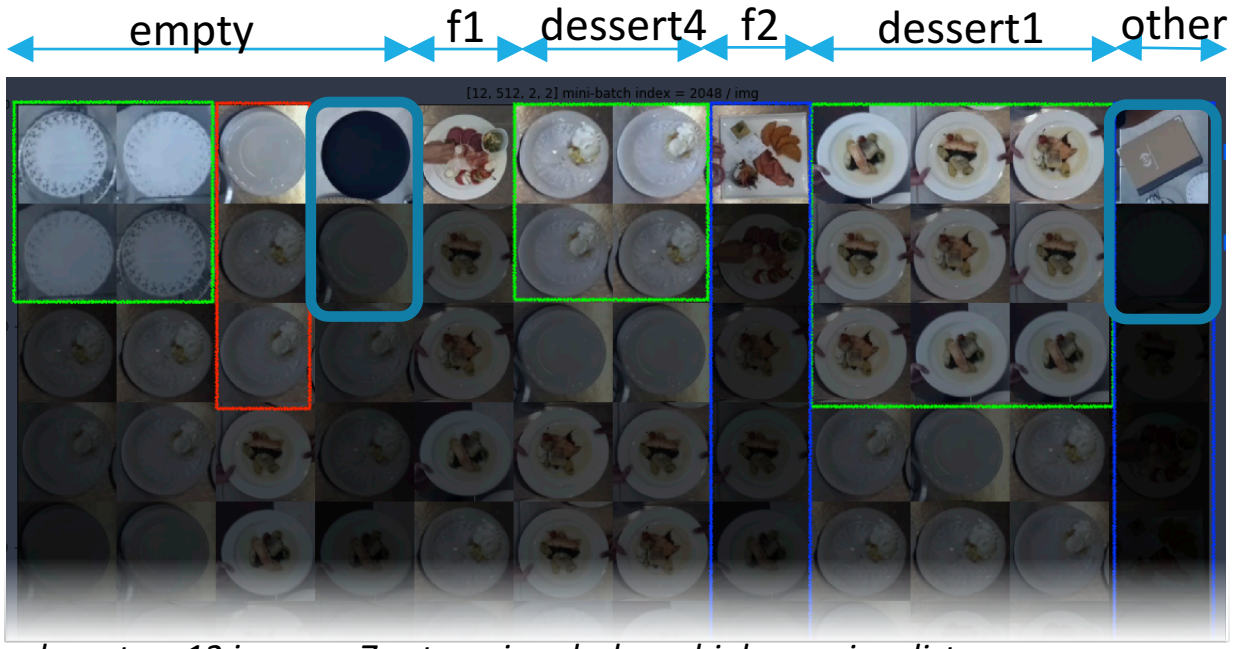
Process

1. **Model** [squeezenet, alexnet, resnet]
2. **Layers** [e.g. ‘features.1’, ‘features.2’]
3. **Reduction** to <2000 with avg_pool kernel size [3,4,5]
4. **Assessment:**
    1. NN-ranking
    2. Plausible false positives

[cf. Yosinski et al. 2014, https://arxiv.org/abs/1411.1792]

Embedding Quality

Example



alexnet on 12 images, 7 categories, darker = higher cosine distance

Result

- alexnet/resnet more accurate embedding than squeezenet
- alexnet additionally:
  - Most plausible false and true positives (Column 4)
  - Highest degree of separation (Last Column)

Process

1. **Model** [squeezenet, alexnet, resnet]
2. **Layers** [e.g. ‘features.1’, ‘features.2’]
3. **Reduction** to <2000 with avg_pool kernel size [3,4,5]
4. **Assessment:**
   1. NN-ranking
   2. Plausible false positives

Choice

alexnet

- Layer: ‘features’ (#1)
- Kernel size for dim reduction: 3
- Resulting dimensionality: 1024

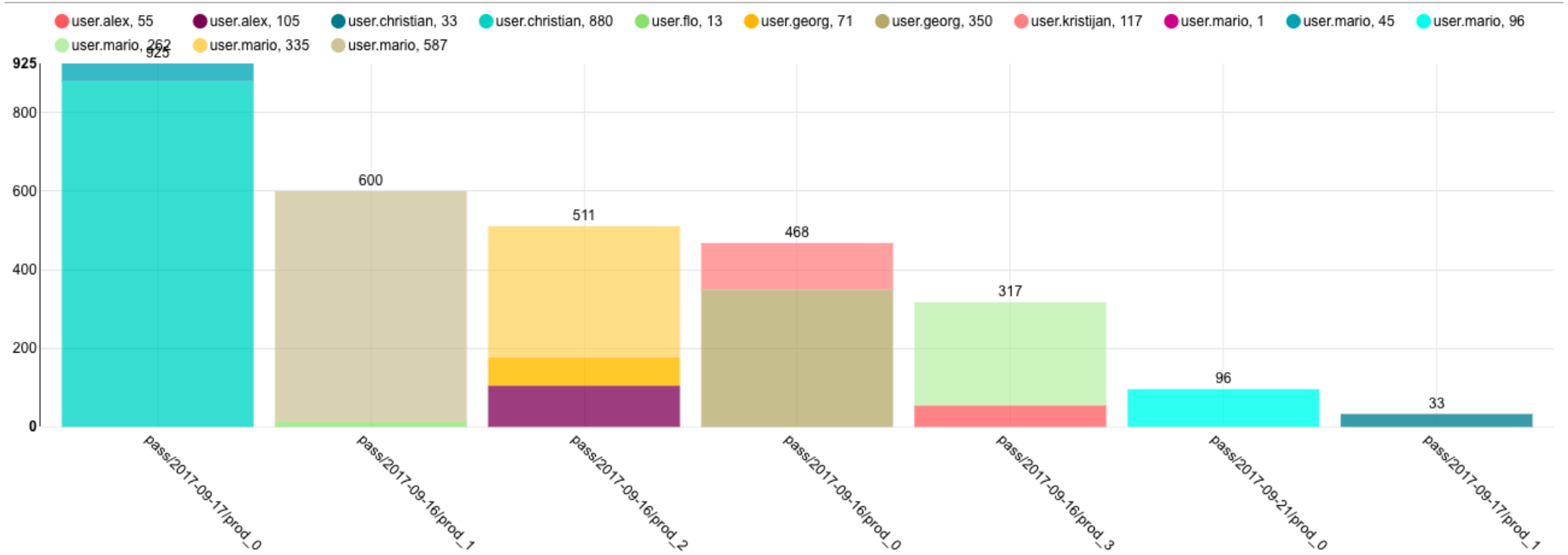
Labelling Tool - Effects

Instant feedback motivates, best practices emerge collaboratively

11:59

also dass das programm dann essen erkennt, leuchtet mir ein. aber body.shoulder! bin i beeindruckt

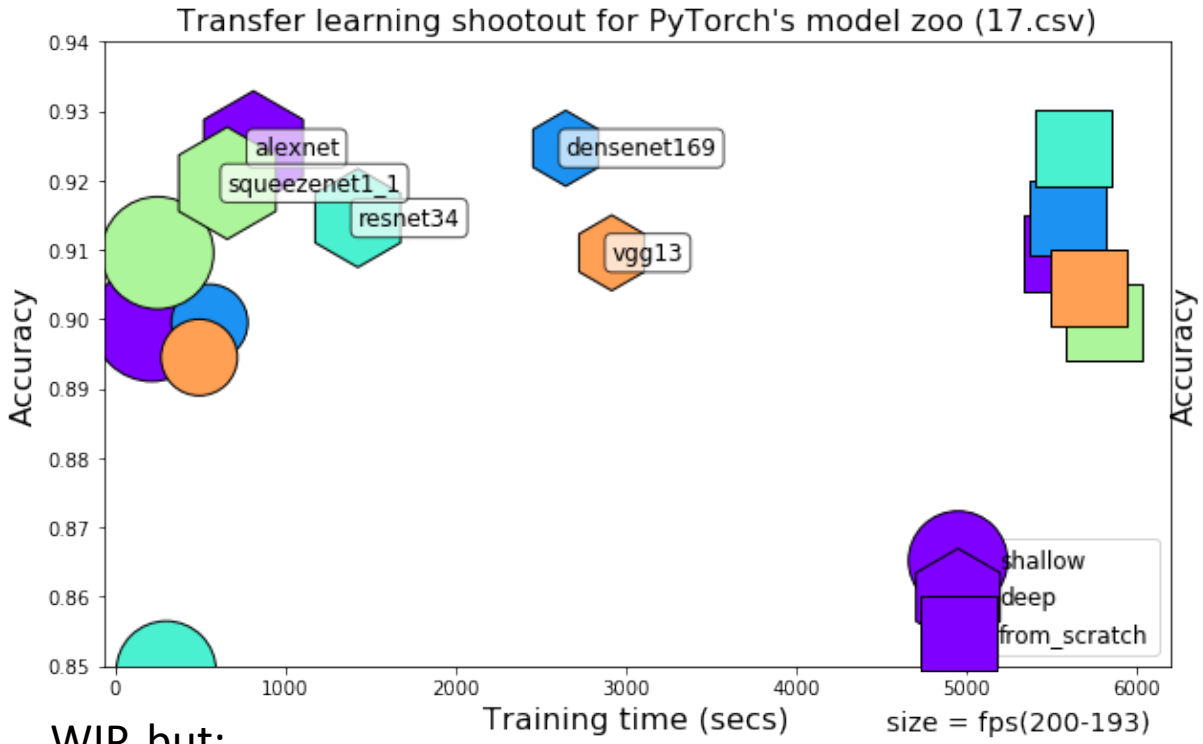
“--that the program then recognizes dishes is clear. But [parts of the body]… I‘m impressed



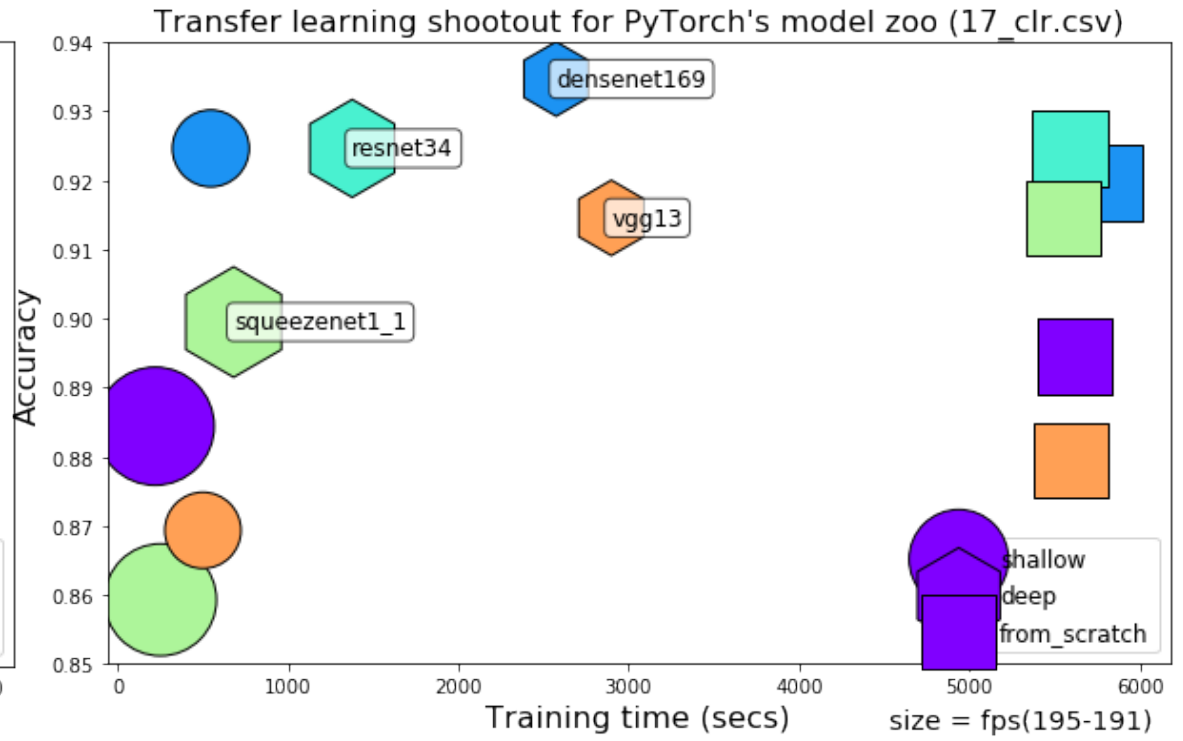
Label Proposals: model **re**-selection

Dish and body parts: many classes, <u>rotation and blur variance</u>

Constant LR w/ momentum



Cyclical Learning Rate [Smith 2017, arxiv.org/abs/1506.01186]



WIP, but:

- Deep retraining wins over shallow given now available real-world data
- Warrants new qualitative assessment along the Pareto curve
- Cyclical LR helps some models (resnet, densenet)

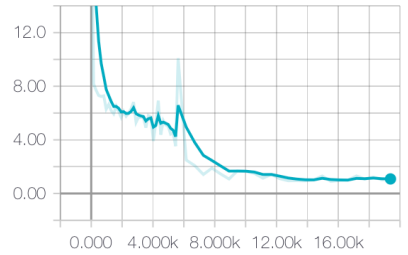
[https://github.com/ahirner/pytorch-retraining]
[https://medium.com/towards-data-science/transfer-learning-with-pytorch-72a052297c51]

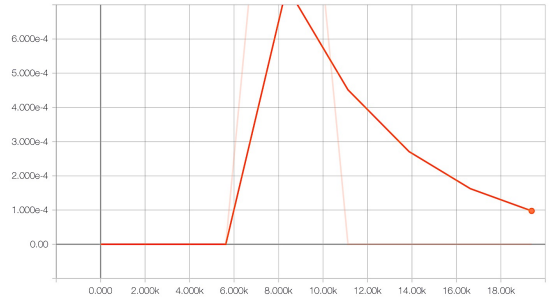
One more thing (Training Process)

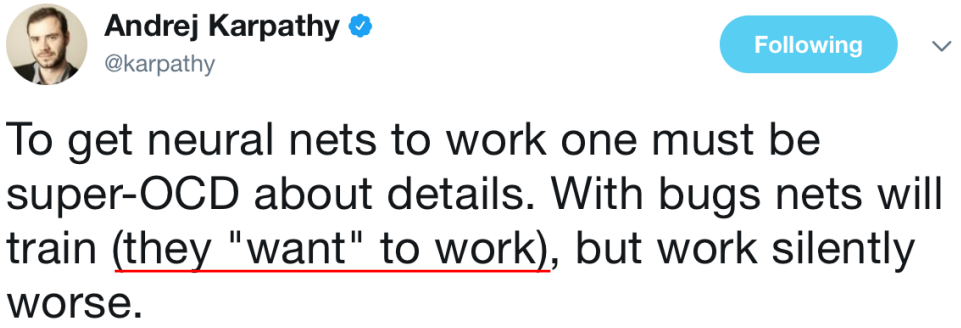
Loss: decreasing monotonically
(almost)

Different Eval

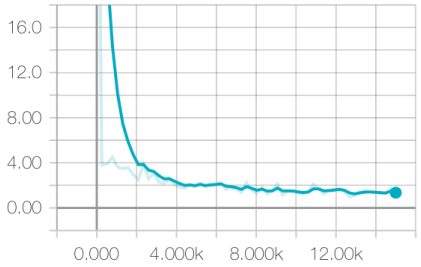
Bug

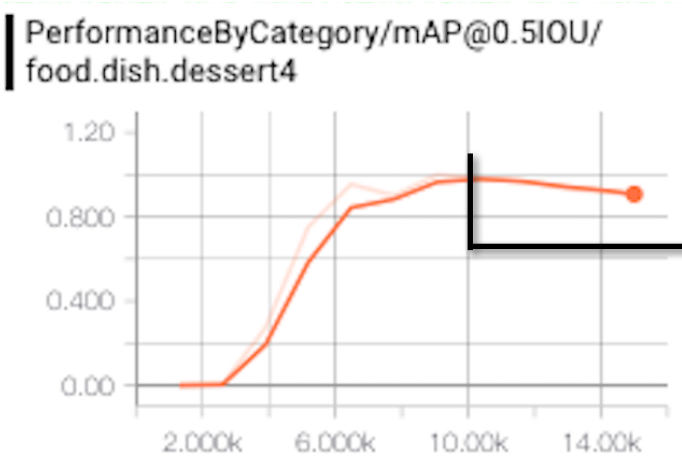






No Bug

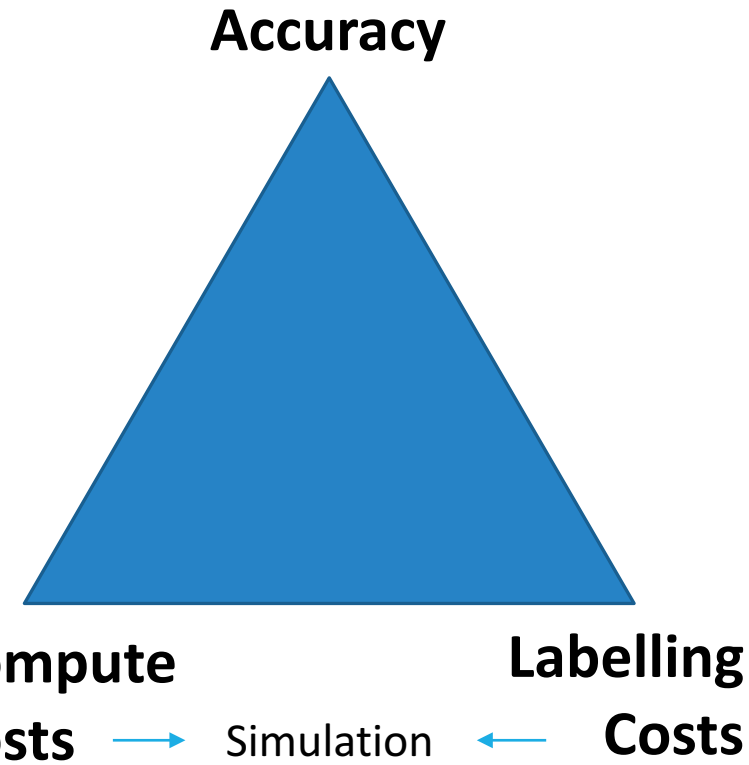




Overfitting = Unit Test of Machine Learning

Transfer (not) all the things

Tradeoff without… → Win/Win with **Transfer Learning**



E.g.: Learning 2 Learn:
$2 Mio. of compute
https://news.ycombinator.com/item?id=14950122

Have all three!

+ Partial Confidentiality
+ Stepping Stone for composable AI
+ Technology transfer between industry and academia

Share not necessarily	Predictions / Generator
Share maybe	Optimization Method
Share	Parameters Compute Graph
Share not necessarily	Ground Truth Data

OpenMined: at the frontier of federated learning with confidentiality guarantees meetup.com/OpenMined-Vienna

Takeaways

- **One-shot learning = ultimate goal**
  - … where machines ask the right questions
  - … where models are learnt from from private data
- **Datascience is 20% work, but payback is highly non-linear**
- **Make iteration of your analysis pipeline:**
  - Collaborative
  - Effortless
- **Join OpenMined-Vienna**
- **Work with us!**