

Course Wrap-Up

Itthi Chatnuntawech

Nanoinformatics and Artificial Intelligence (NAI)

National Nanotechnology Center (NANOTEC)

National Science and Technology Development Agency (NSTDA)

September 28, 2022

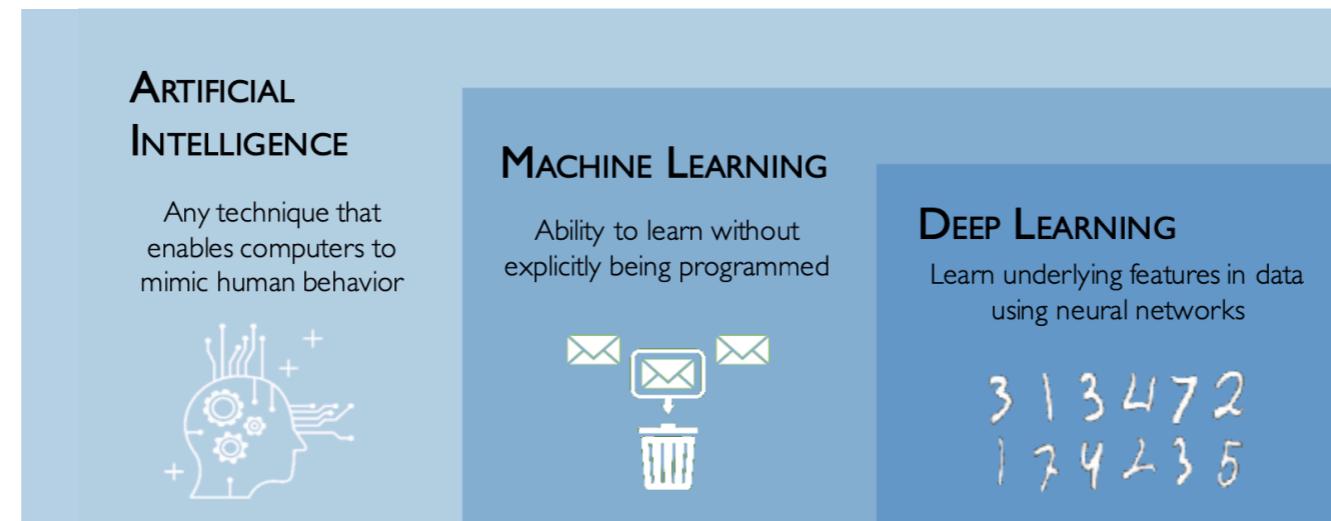
Outline

- ❖ Machine Learning Pitfalls
 - ❖ ML algorithms are not one-size-fits-all
 - ❖ Having more data doesn't always help
 - ❖ Exploiting spurious correlations
 - ❖ Training data don't capture real use cases
 - ❖ Having data leakage
 - ❖ Reporting a “non-informative” evaluation metric
 - ❖ Solving “irrelevant” problems
- ❖ A Way to Communicate with ML Practitioners
 - ❖ Be clear about what the inputs and outputs are
 - ❖ State the requirements and use-cases
 - ❖ Provide domain- / field-specific information
 - ❖ Summary of related papers
- ❖ Resources

Outline

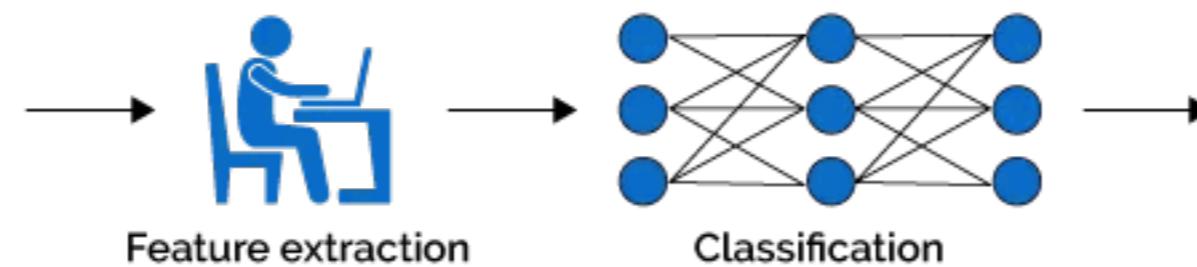
- ❖ Machine Learning Pitfalls
 - ❖ ML algorithms are not one-size-fits-all
 - ❖ Having more data doesn't always help
 - ❖ Exploiting spurious correlations
 - ❖ Training data don't capture real use cases
 - ❖ Having data leakage
 - ❖ Reporting a “non-informative” evaluation metric
 - ❖ Solving “irrelevant” problems
- ❖ A Way to Communicate with ML Practitioners
 - ❖ Be clear about what the inputs and outputs are
 - ❖ State the requirements and use-cases
 - ❖ Provide domain- / field-specific information
 - ❖ Summary of related papers
- ❖ Resources

Pitfalls: ML Algorithms Are Not One-Size-Fits-All



Traditional Machine Learning

Input image



Output

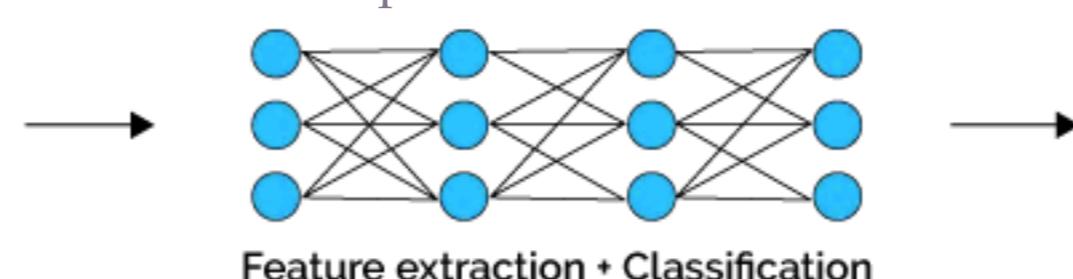
watermelon

Input image



Deep Learning

deep neural network

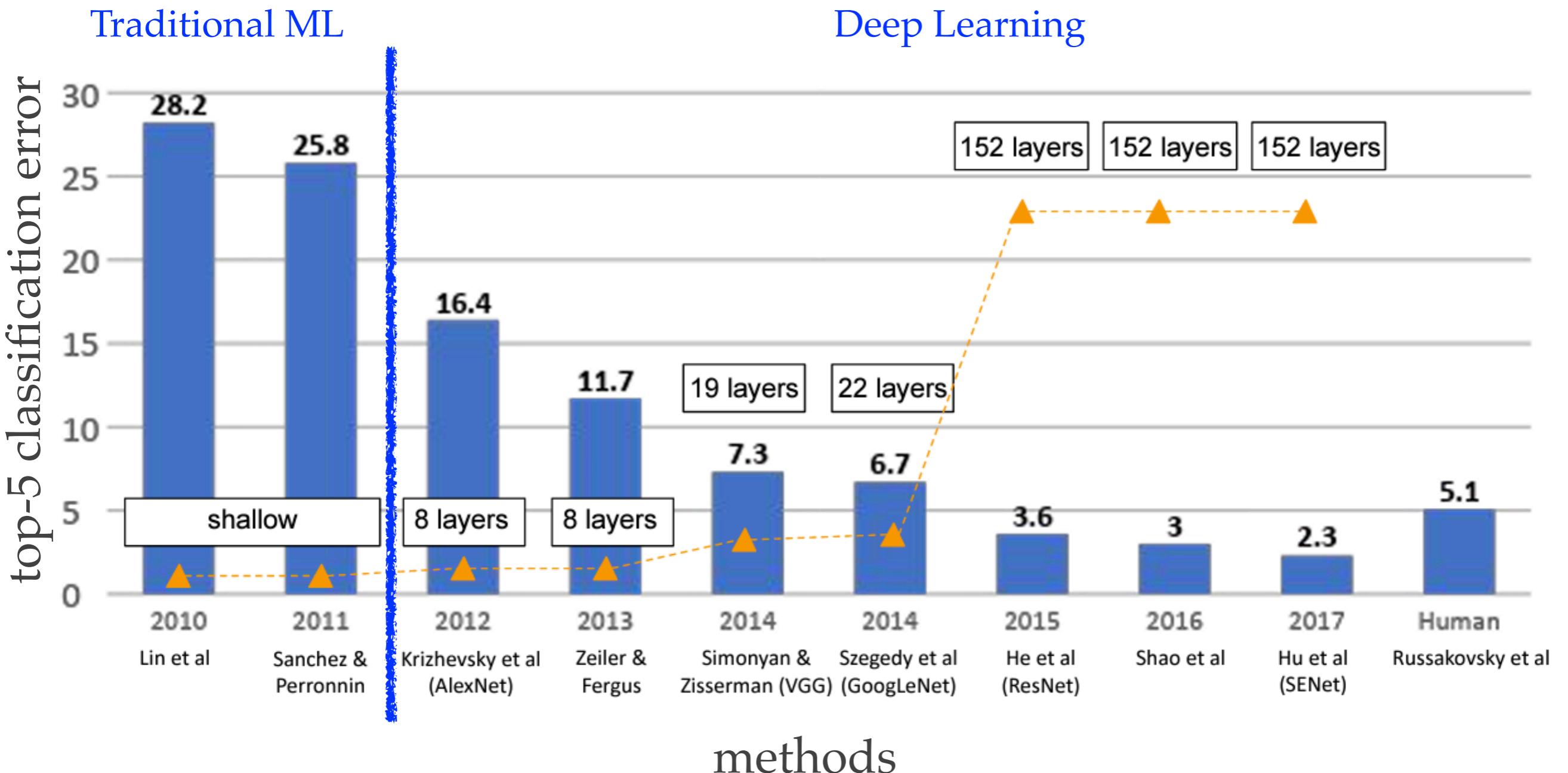


Output

watermelon

Pitfalls: ML Algorithms Are Not One-Size-Fits-All

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



Pitfalls: ML Algorithms Are Not One-Size-Fits-All



Pitfalls: ML Algorithms Are Not One-Size-Fits-All

Table 1: Dataset properties. Notation: “RMSE” ~ root-mean-square error, “Acc.” ~ accuracy.

Tabular Data

	CA	AD	HE	JA	HI	AL	EP	YE	CO	YA	MI
#objects	20640	48842	65196	83733	98050	108000	500000	515345	581012	709877	1200192
#num. features	8	6	27	54	28	128	2000	90	54	699	136
#cat. features	0	8	0	0	0	0	0	0	0	0	0
metric	RMSE	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	RMSE	Acc.	RMSE	RMSE
#classes	–	2	100	4	2	1000	2	–	7	–	–

larger datasets



Table 4: Results for ensembles of GBDT and the main DL models. For each model-dataset pair, the metric value averaged over three ensembles is reported. See supplementary for standard deviations. Notation follows Table 3.

	CA ↓	AD ↑	HE ↑	JA ↑	HI ↑	AL ↑	EP ↑	YE ↓	CO ↑	YA ↓	MI ↓
Default hyperparameters											
XGBoost	0.462	0.874	0.348	0.711	0.717	0.924	0.8799	9.192	0.964	0.761	0.751
CatBoost	0.428	0.873	0.386	0.724	0.728	0.948	0.8893	8.885	0.910	0.749	0.744
FT-Transformer	0.454	0.860	0.395	0.734	0.731	0.966	0.8969	8.727	0.973	0.747	0.742

FT-Transformer allows building powerful ensembles out of the box.

Deep learning



Pitfalls: ML Algorithms Are Not One-Size-Fits-All

Table 1: Dataset properties. Notation: “RMSE” ~ root-mean-square error, “Acc.” ~ accuracy.

Tabular Data

	CA	AD	HE	JA	HI	AL	EP	YE	CO	YA	MI
#objects	20640	48842	65196	83733	98050	108000	500000	515345	581012	709877	1200192
#num. features	8	6	27	54	28	128	2000	90	54	699	136
#cat. features	0	8	0	0	0	0	0	0	0	0	0
metric	RMSE	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	RMSE	Acc.	RMSE	RMSE
#classes	–	2	100	4	2	1000	2	–	7	–	–

larger datasets



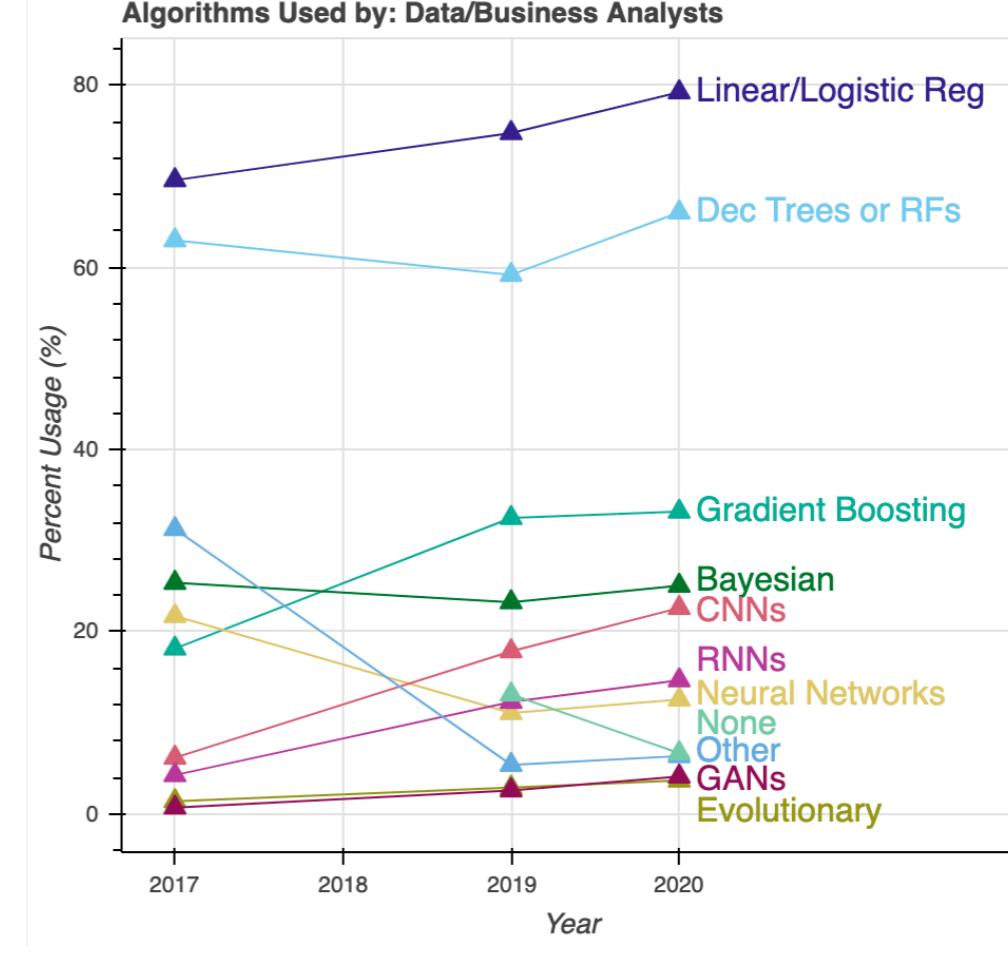
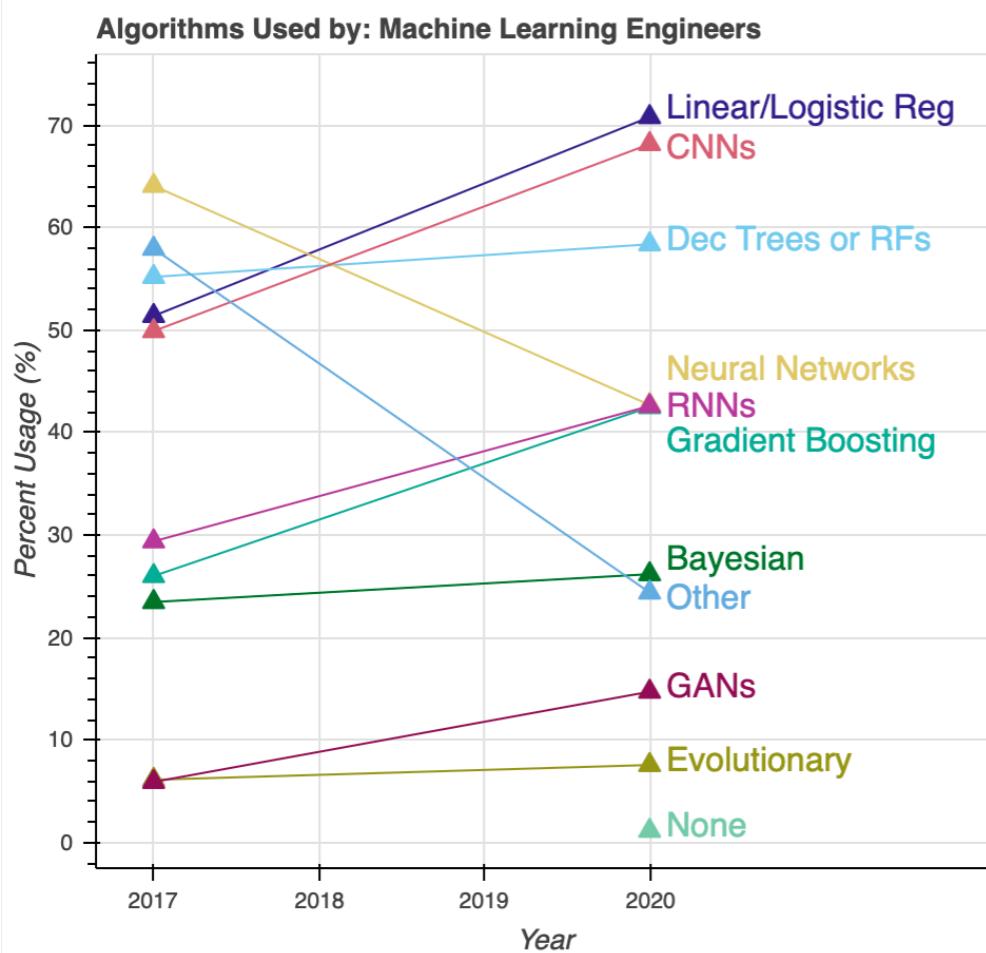
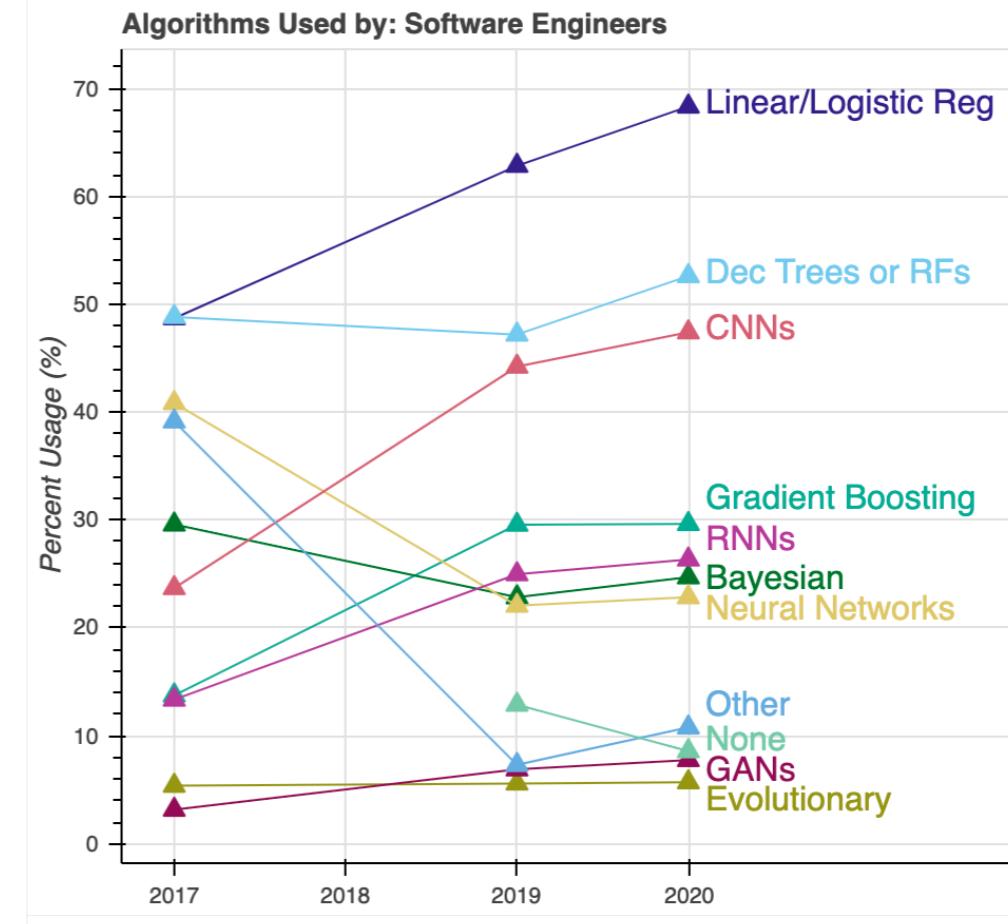
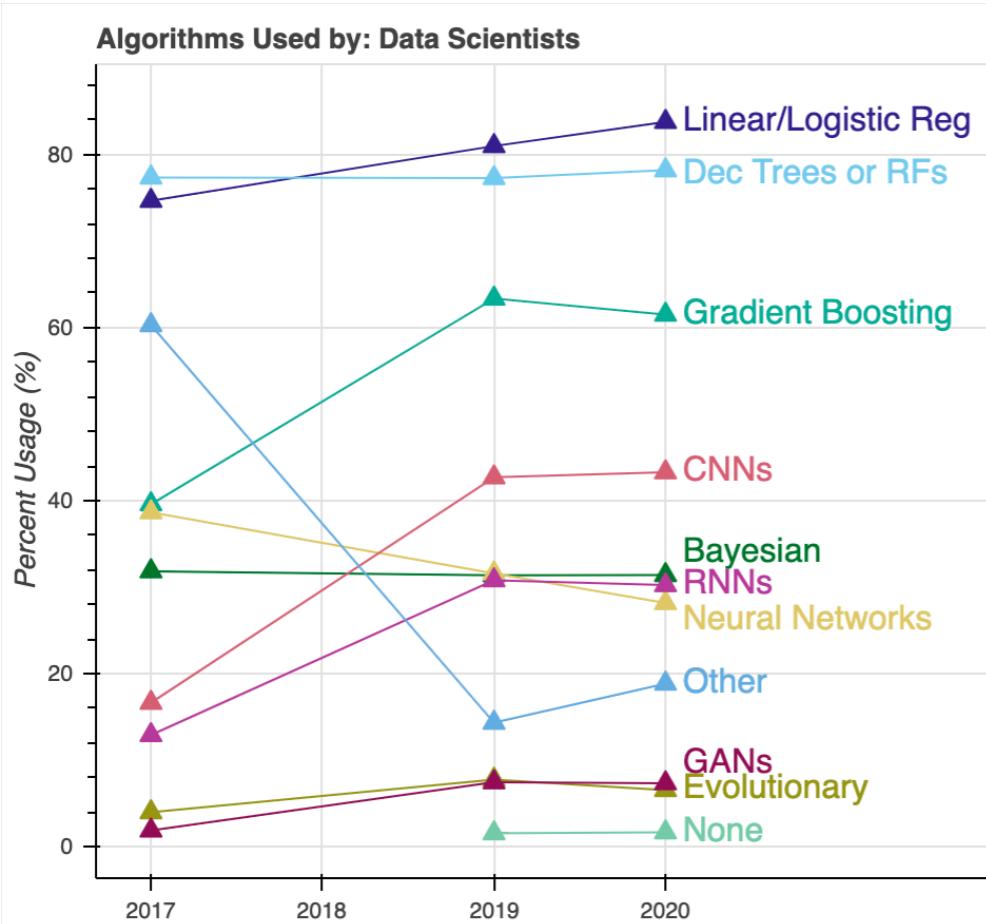
Table 4: Results for ensembles of GBDT and the main DL models. For each model-dataset pair, the metric value averaged over three ensembles is reported. See supplementary for standard deviations. Notation follows Table 3.

	CA ↓	AD ↑	HE ↑	JA ↑	HI ↑	AL ↑	EP ↑	YE ↓	CO ↑	YA ↓	MI ↓
Default hyperparameters											
XGBoost	0.462	0.874	0.348	0.711	0.717	0.924	0.8799	9.192	0.964	0.761	0.751
CatBoost	0.428	0.873	0.386	0.724	0.728	0.948	0.8893	8.885	0.910	0.749	0.744
FT-Transformer	0.454	0.860	0.395	0.734	0.731	0.966	0.8969	8.727	0.973	0.747	0.742
Tuned hyperparameters											
XGBoost	0.431	0.872	0.377	0.724	0.728	–	0.8861	8.819	0.969	0.732	0.742
CatBoost	0.423	0.874	0.388	0.727	0.729	–	0.8898	8.837	0.968	0.740	0.741
ResNet	0.478	0.857	0.398	0.734	0.731	0.966	0.8976	8.770	0.967	0.751	0.745
FT-Transformer	0.448	0.860	0.398	0.739	0.731	0.967	0.8984	8.751	0.973	0.747	0.743

Deep learning

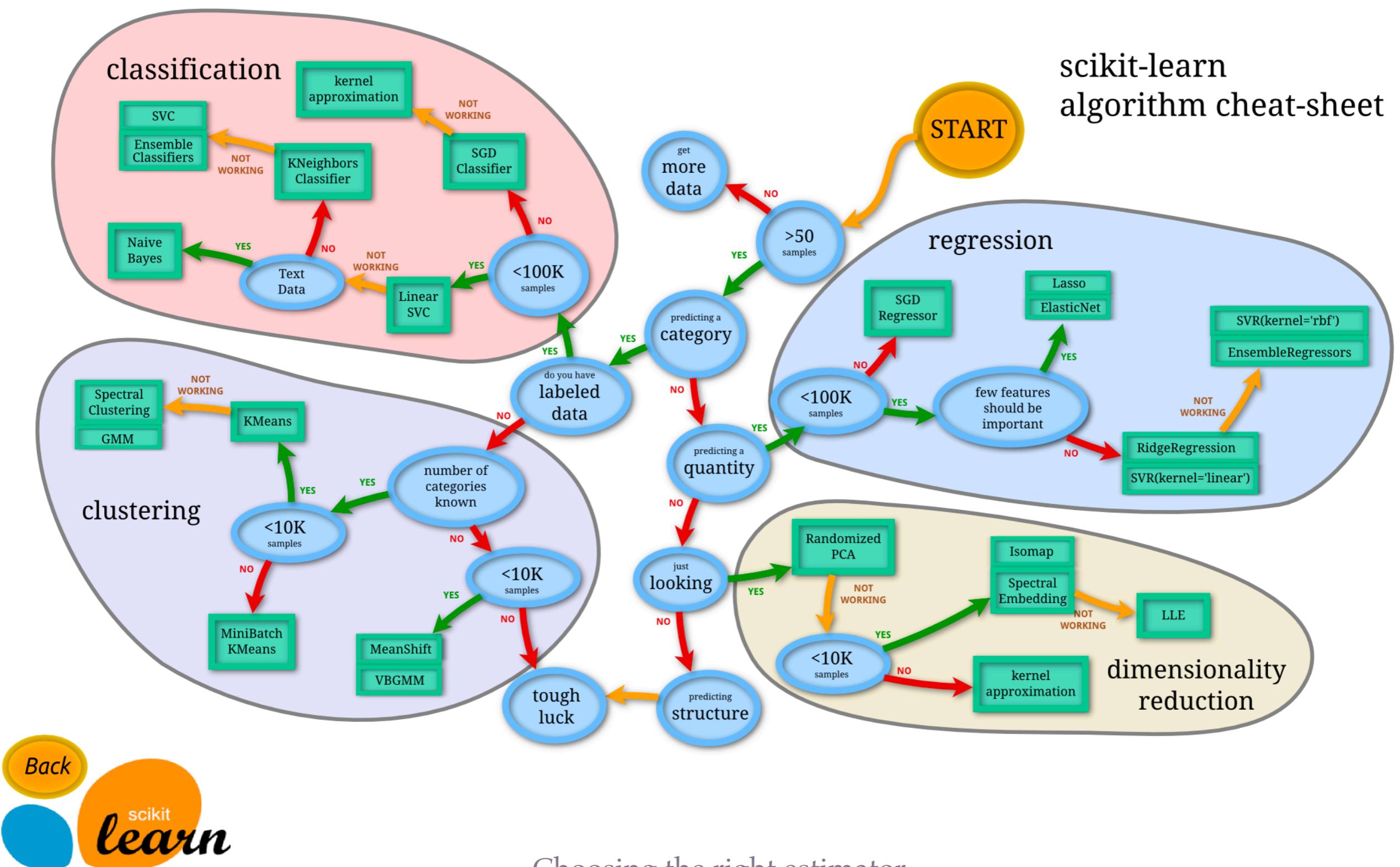
Deep learning

No single best method



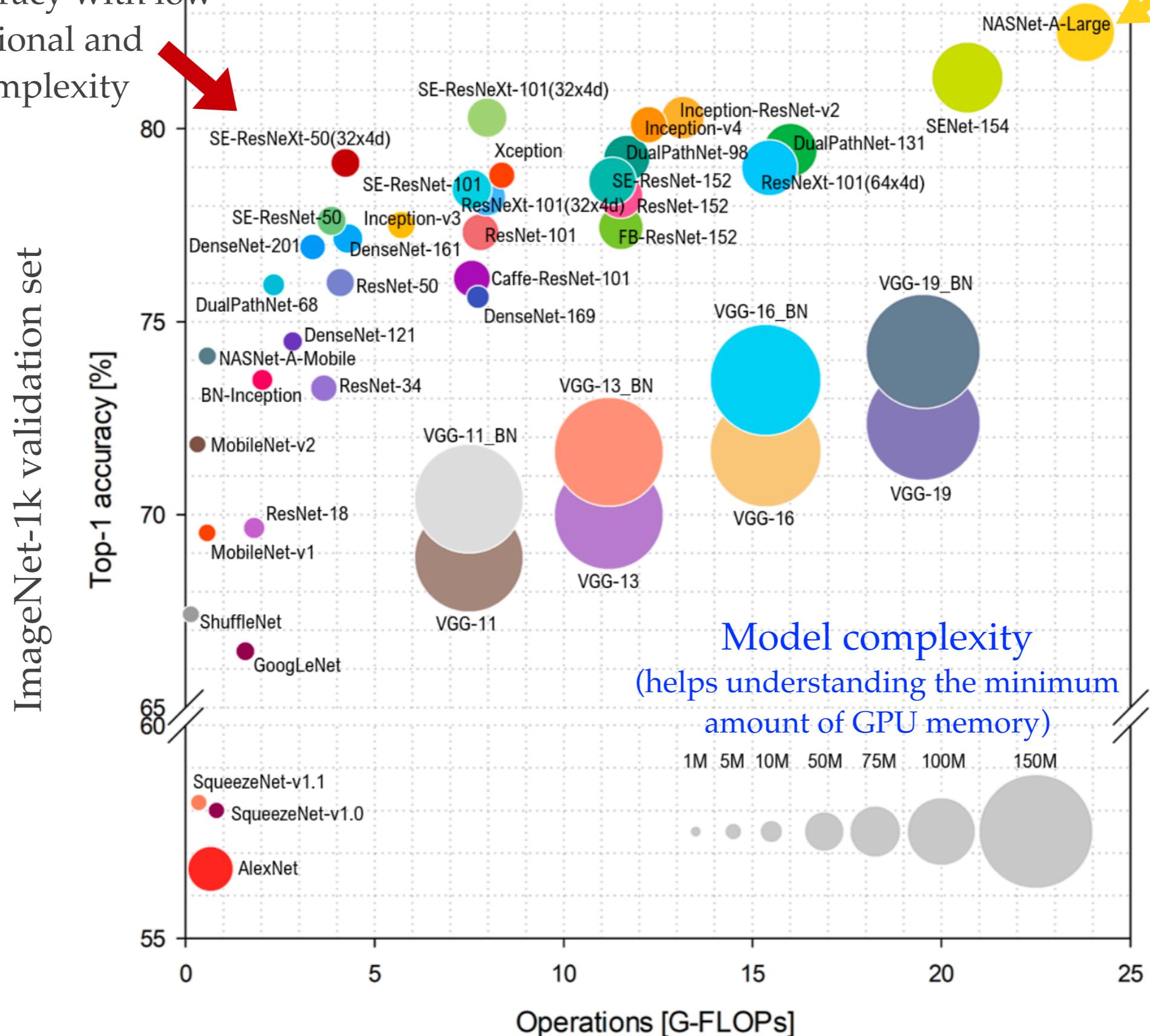
Data Science Trends Based on 4 Years of Kaggle Surveys

Pitfalls: ML Algorithms Are Not One-Size-Fits-All



high accuracy with low computational and model complexity

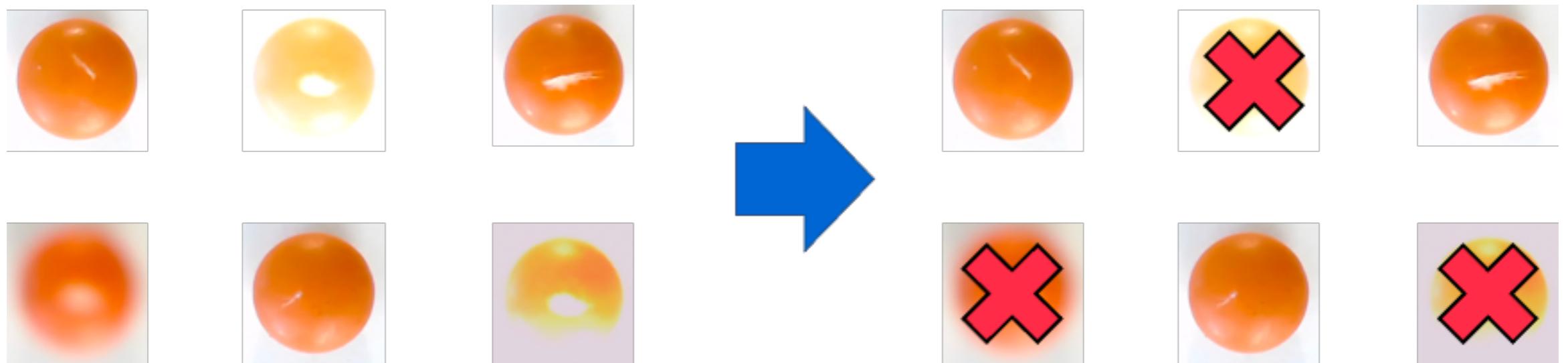
highest accuracy & computational complexity



Floating-point operations (FLOPs) required for a single forward pass (computational cost)

Pitfalls: Having More Data Doesn't Always Help

Low quality/irrelevant samples



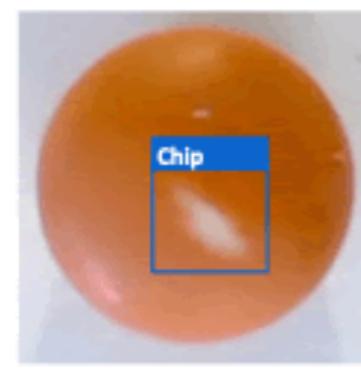
Inconsistent labels

Examples of inconsistencies

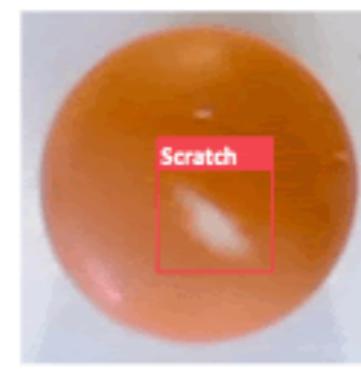
Label name

Bounding box size

Number of bounding boxes

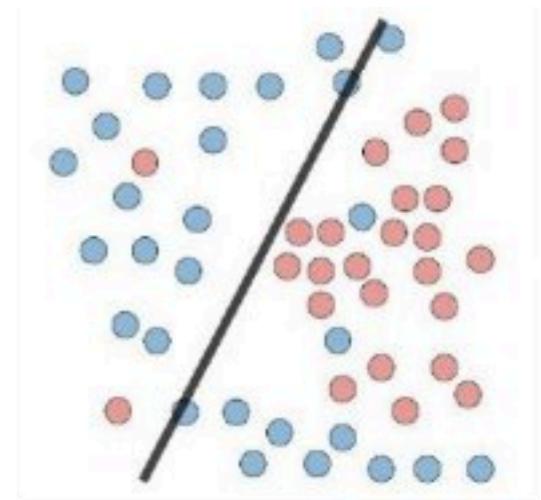
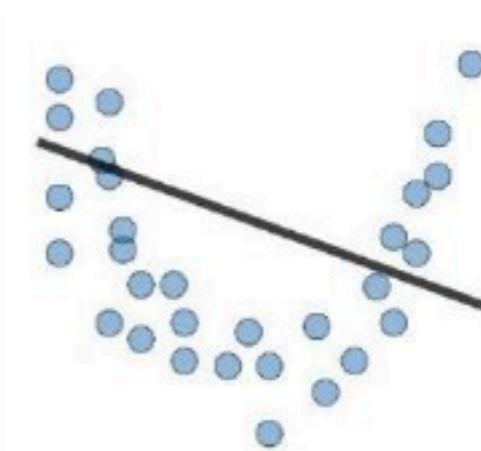


Labeler 1



Labeler 2

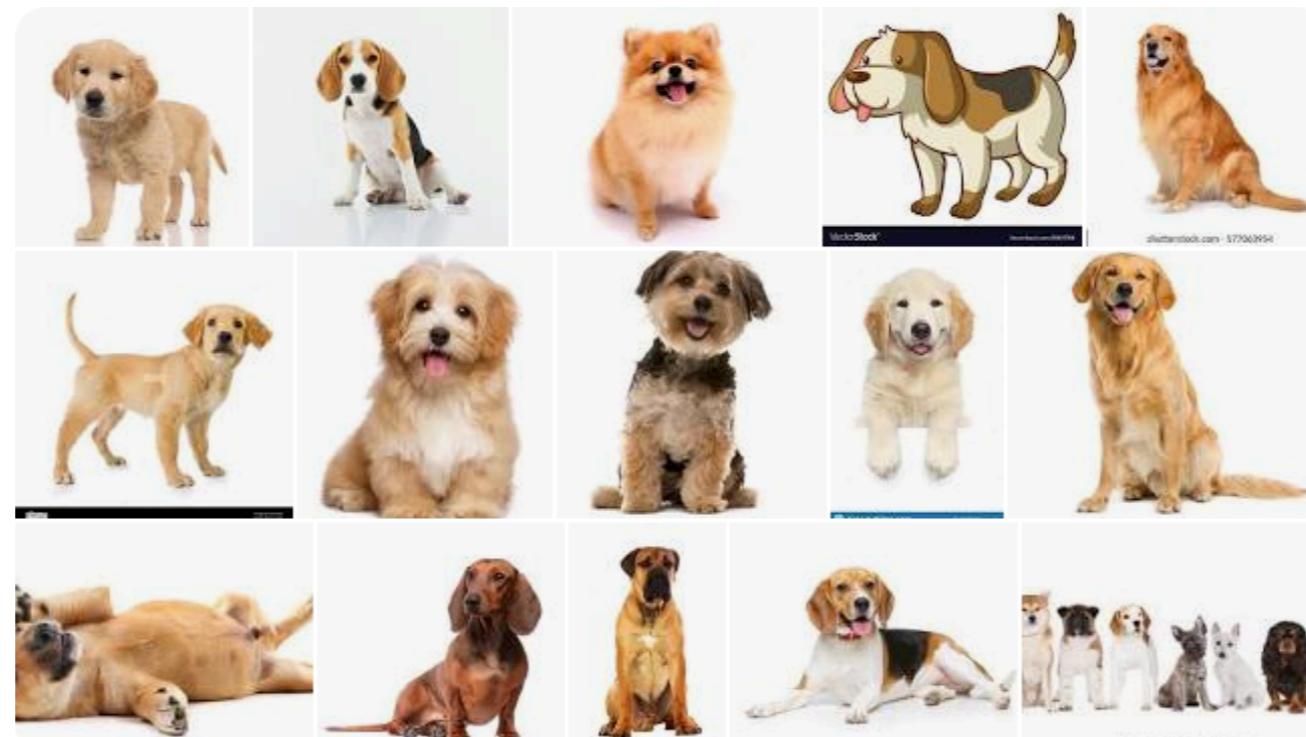
Underfitting



<https://landing.ai/tips-for-a-data-centric-ai-approach/>

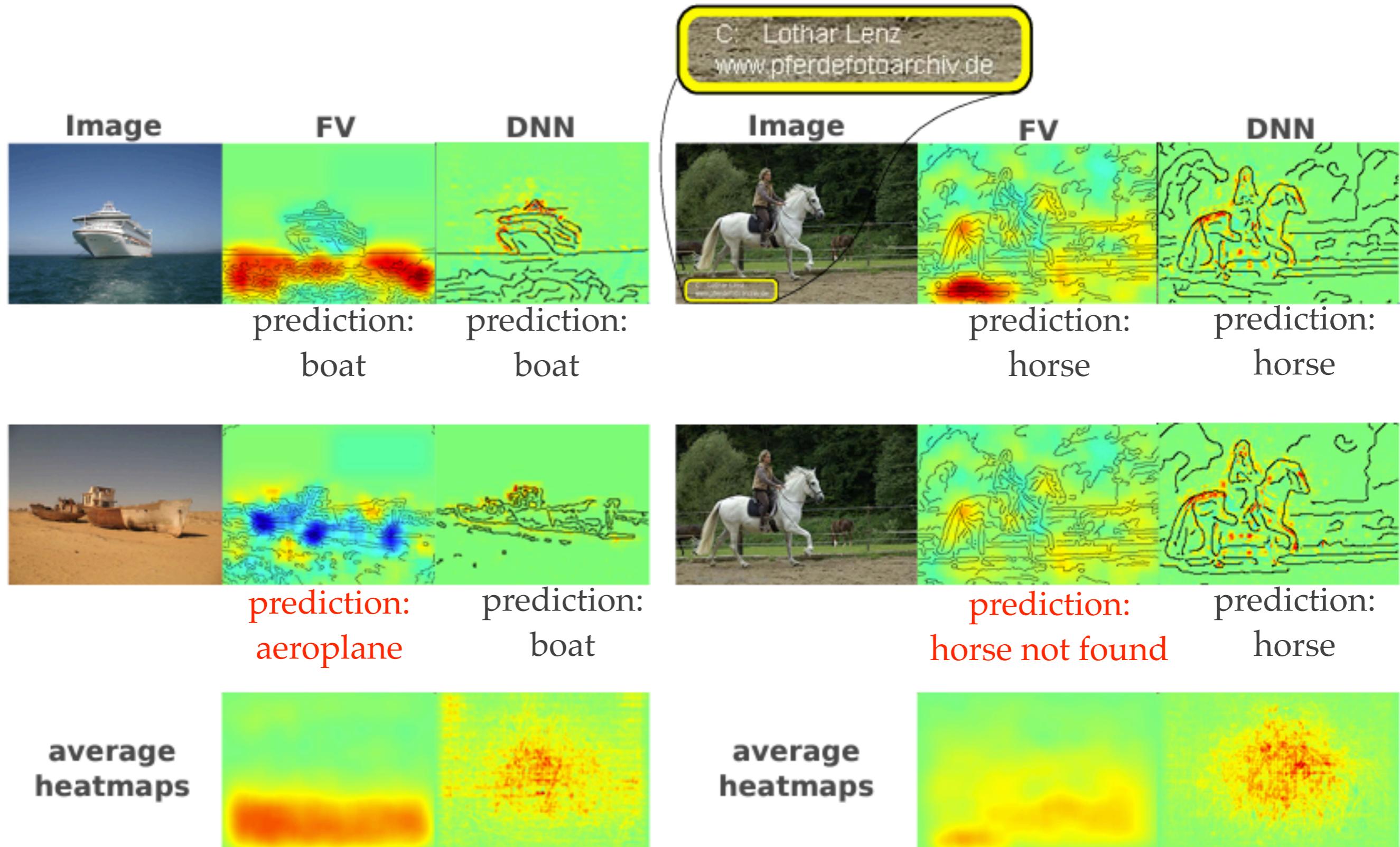
<https://i.pinimg.com/originals/72/e2/22/72e222c1542539754df1d914cb671bd7.png>

Pitfalls: Exploiting Spurious Correlations



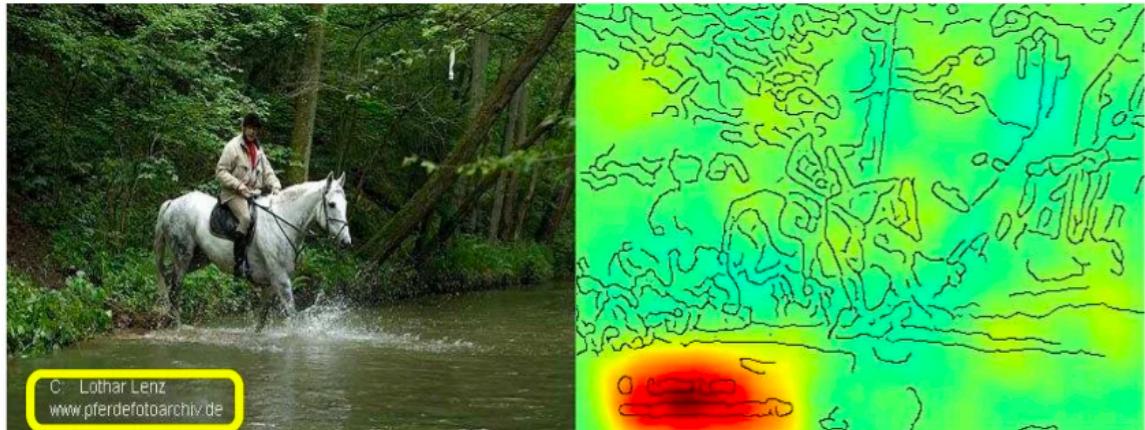
Images obtained using
Google Images

Pitfalls: Exploiting Spurious Correlations



Pitfalls: Exploiting Spurious Correlations

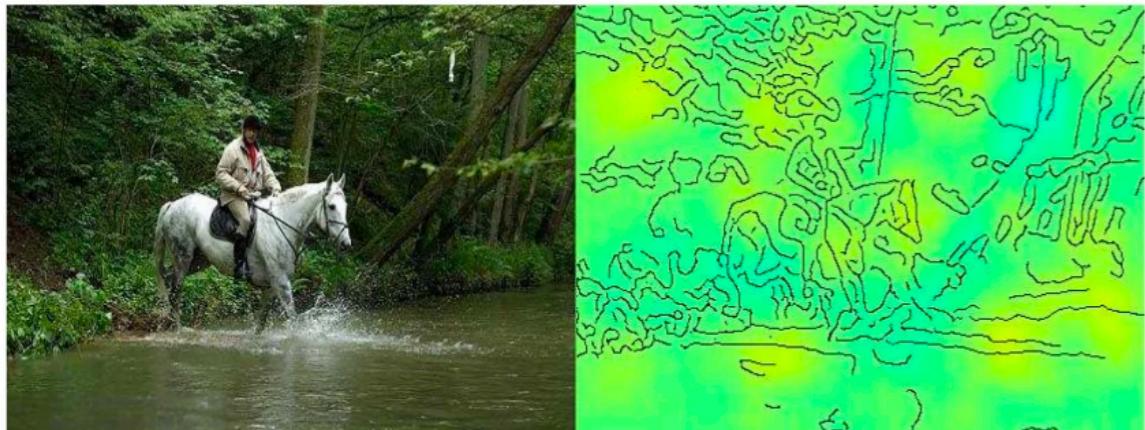
Horse-picture from Pascal VOC data set



Source tag present
↓

Classified as horse

Artificial picture of a car



No source tag present
↓

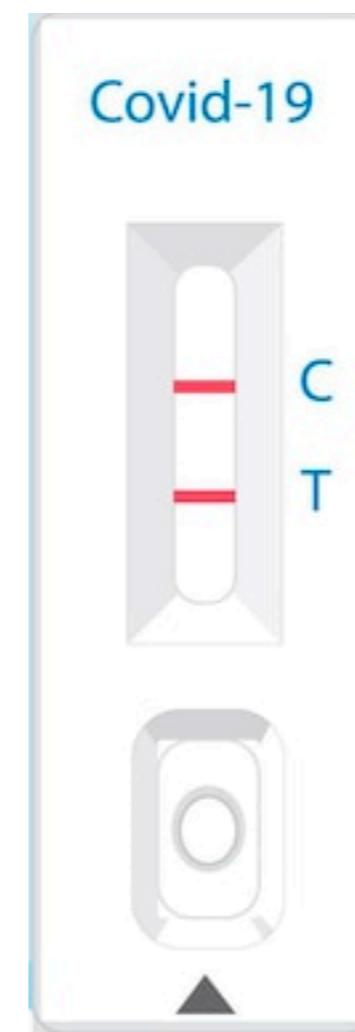
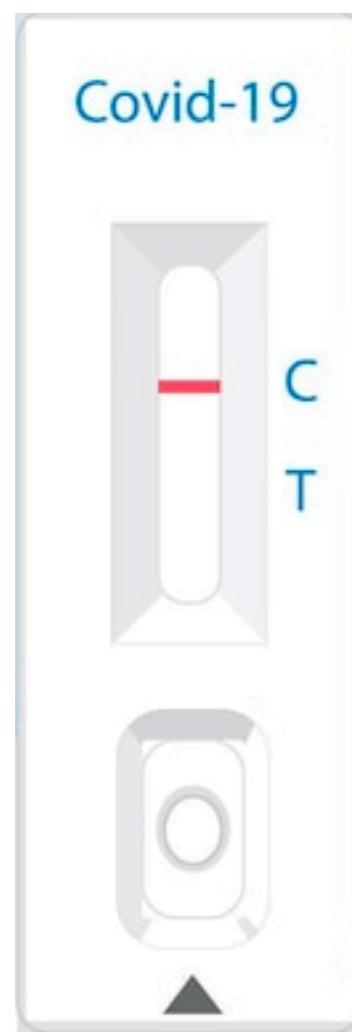
Not classified as horse



"The Fisher vector classifier trained on the PASCAL VOC 2007 data set focuses on a source tag present in about one-fifth of the horse figures. Removing the tag also removes the ability to classify the picture as a horse. Furthermore, inserting the tag on a car image changes the classification from car to horse."

Pitfalls: Training Data Don't Capture Real Use Cases

Training data

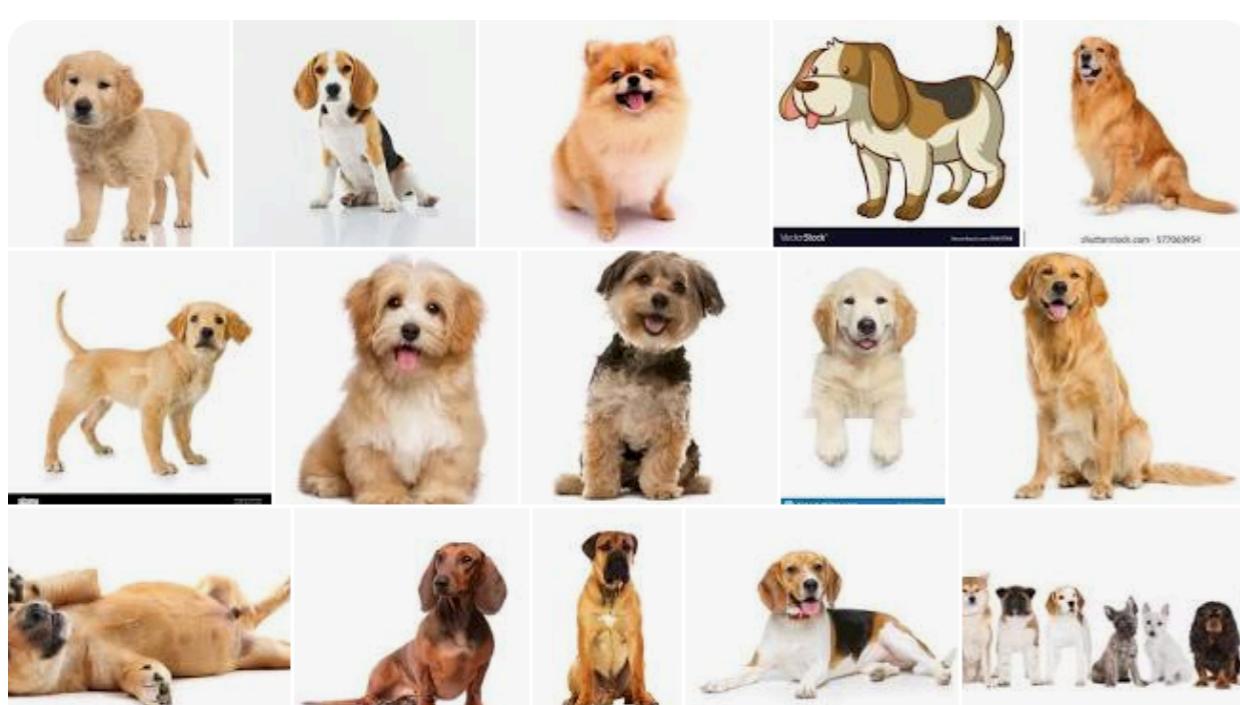


Test data



Pitfalls: Training Data Don't Capture Real Use Cases

Training data

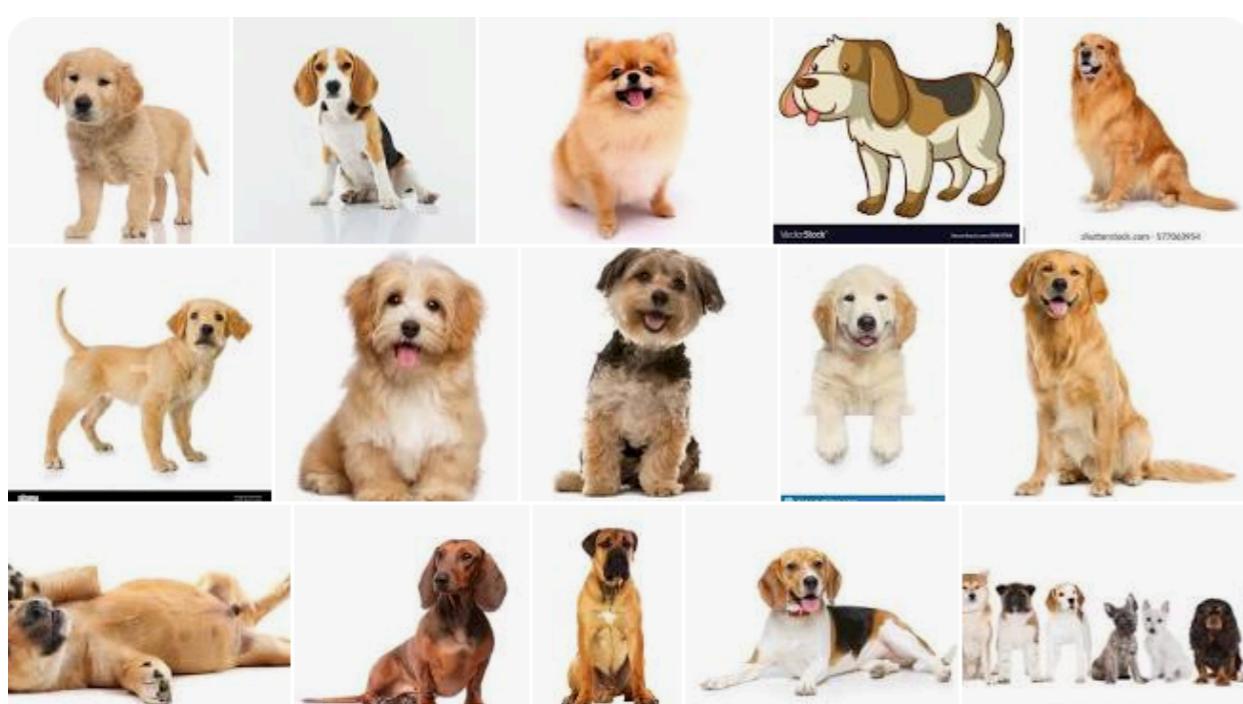


Test data



Pitfalls: Training Data Don't Capture Real Use Cases

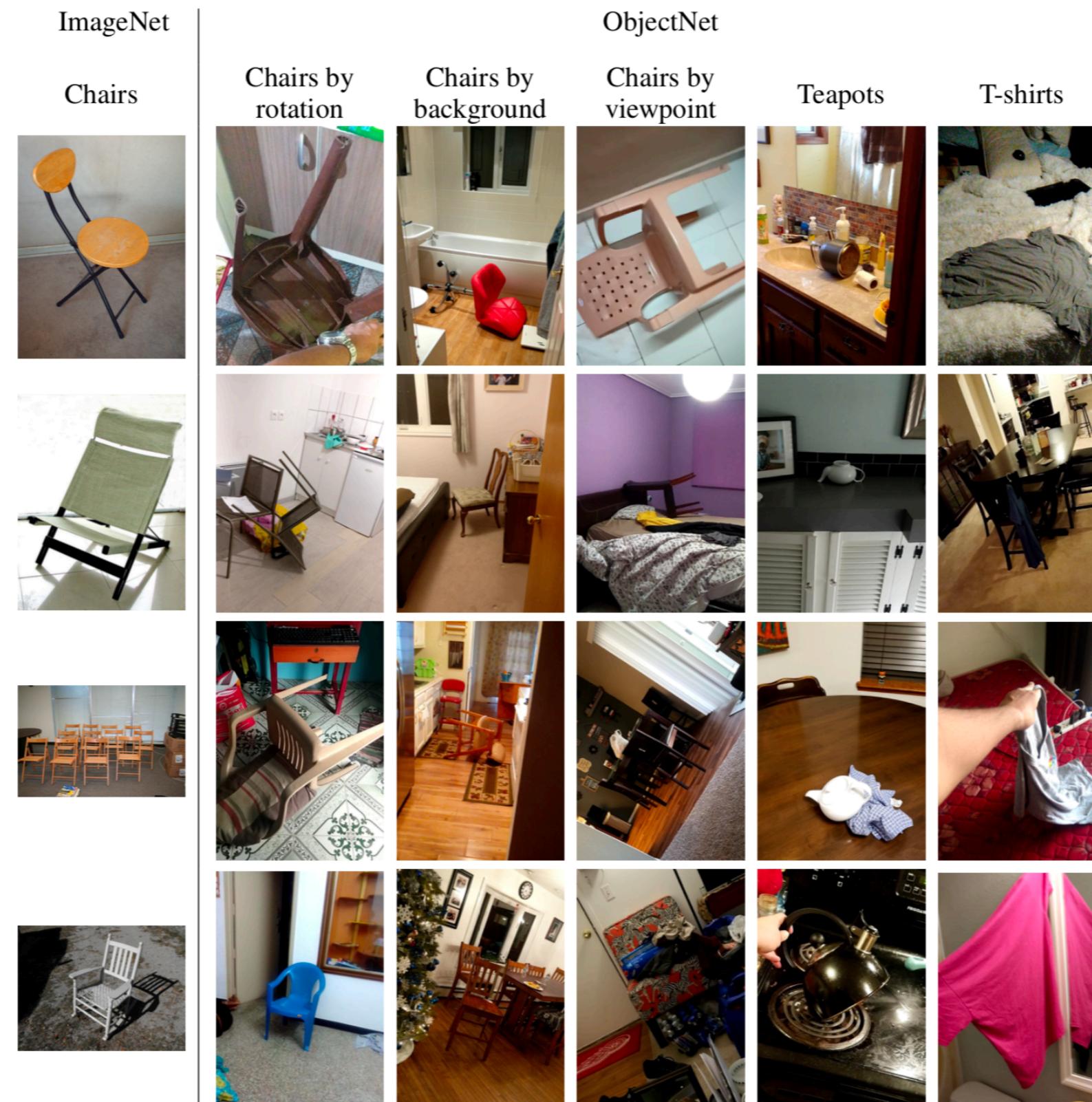
Training data



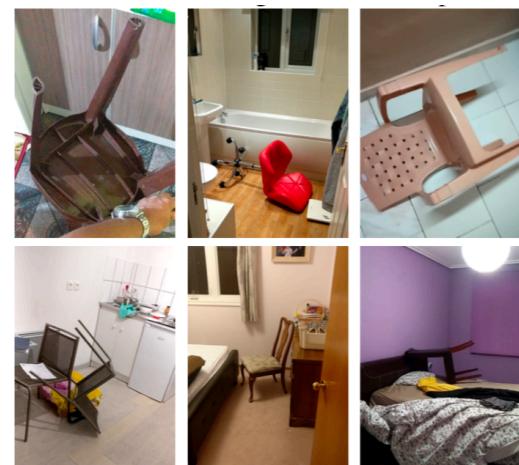
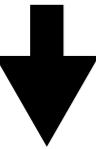
Test data



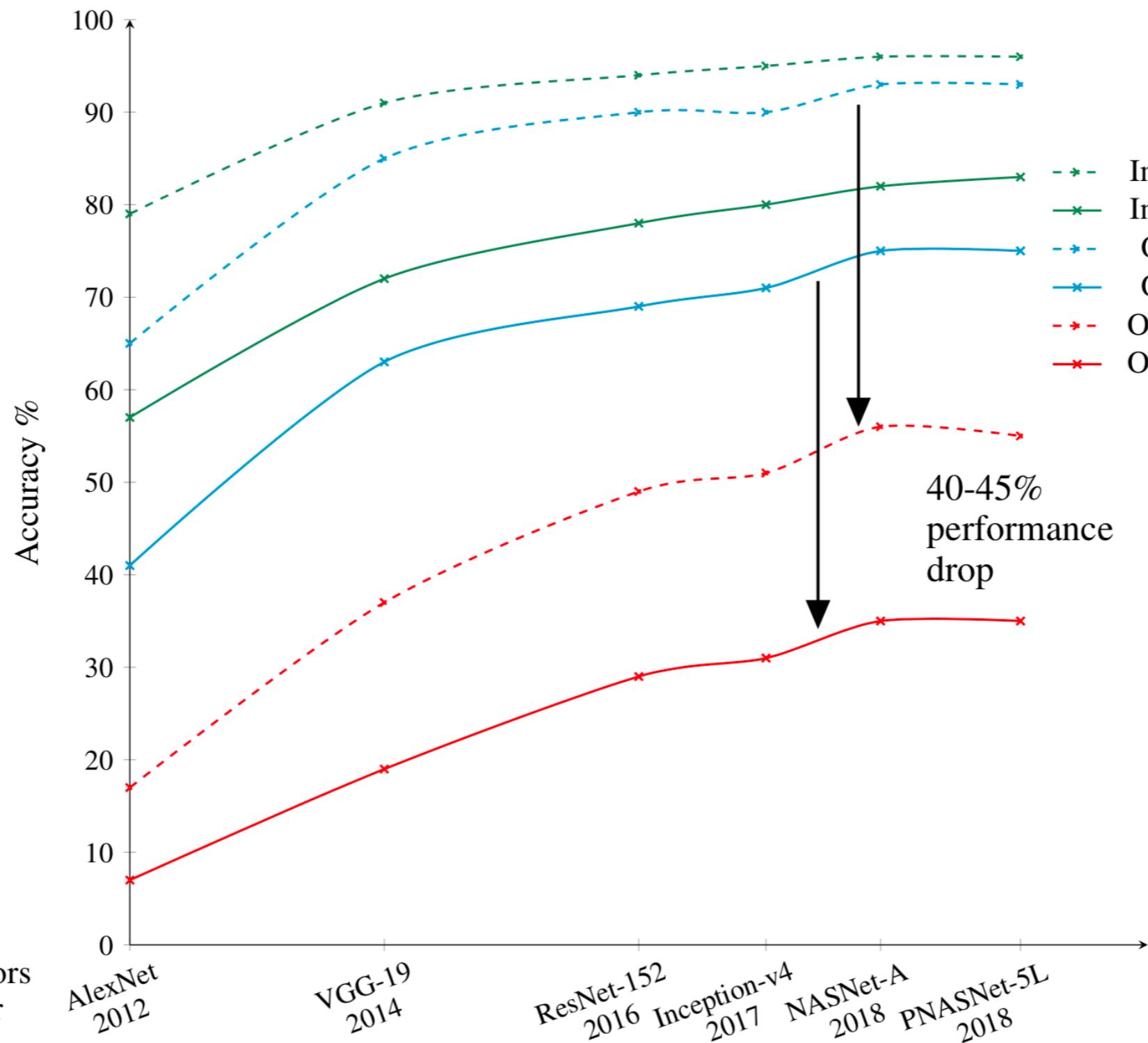
Pitfalls: Training Data Don't Capture Real Use Cases



Pitfalls: Training Data Don't Capture Real Use Cases

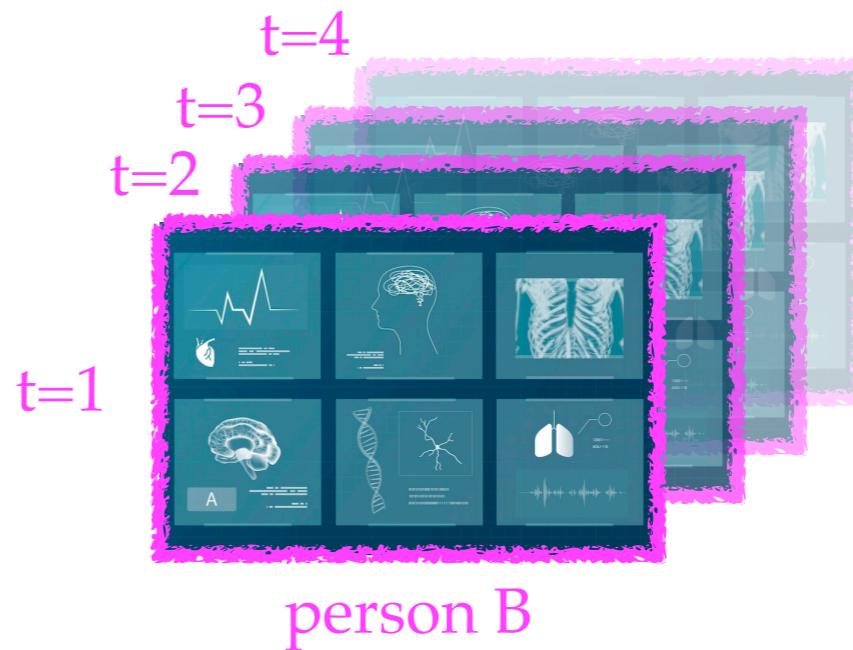
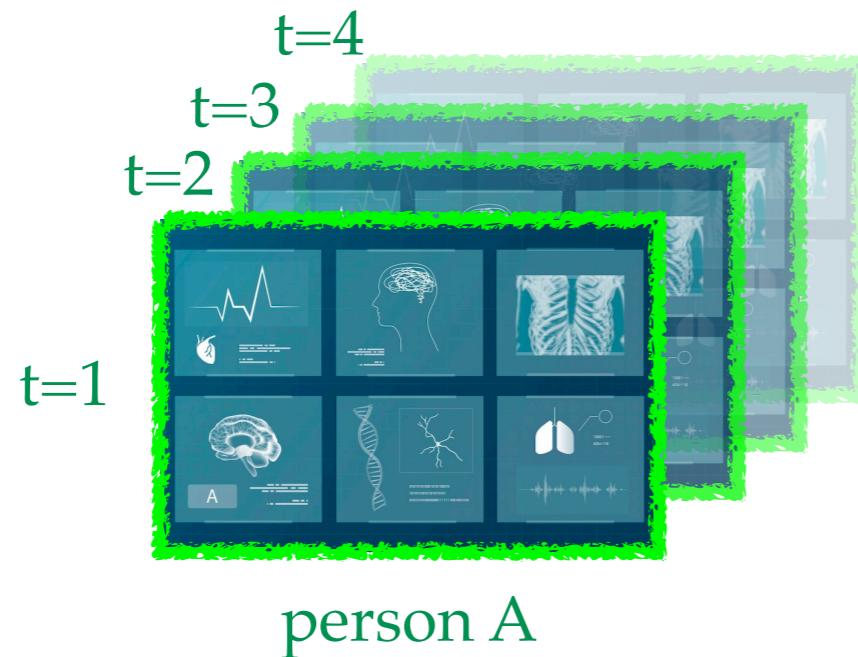


Detectors
by year



Pitfalls: Having Data Leakage

The whole dataset



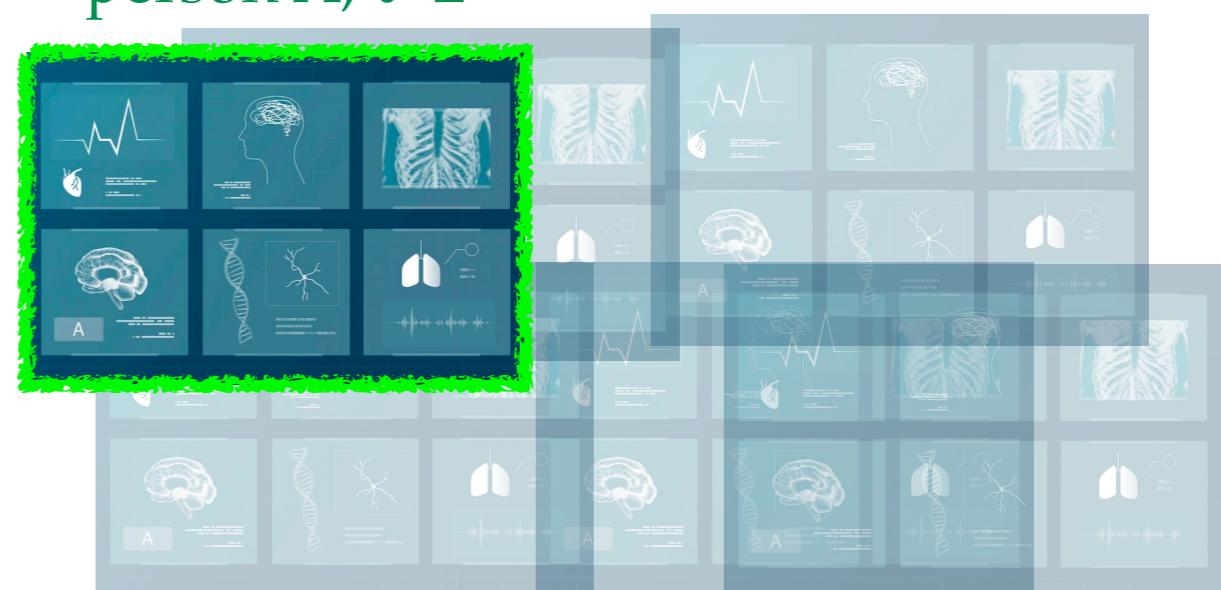
Training data

person A, t=2



Test data

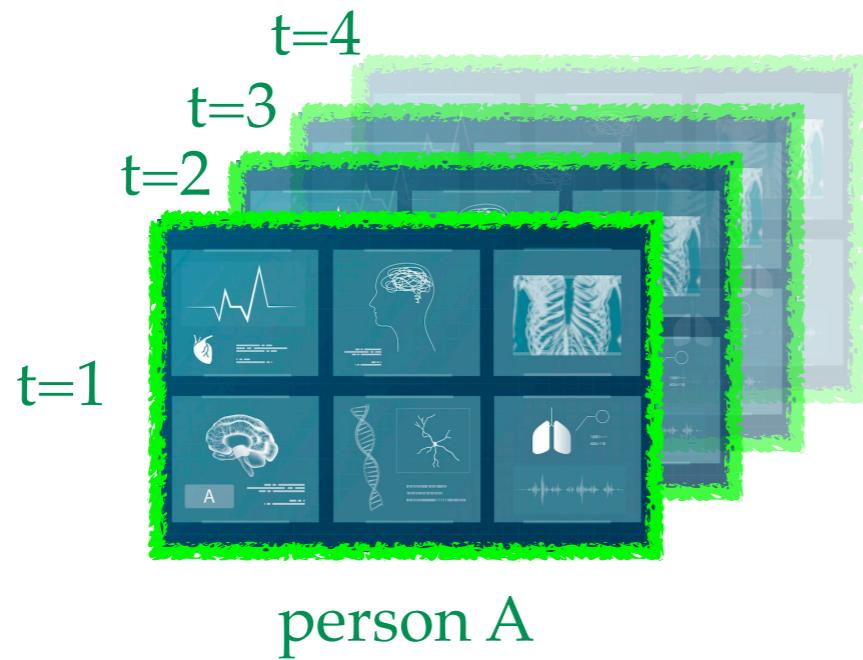
person A, t=2



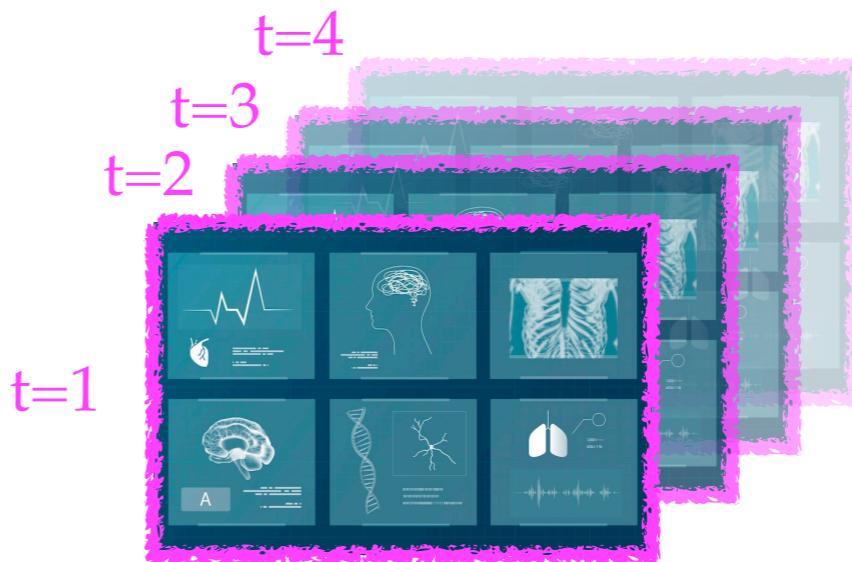
Some of the training and test samples are identical

Pitfalls: Having Data Leakage

The whole dataset



person A



person B

Training data

person A, t=1

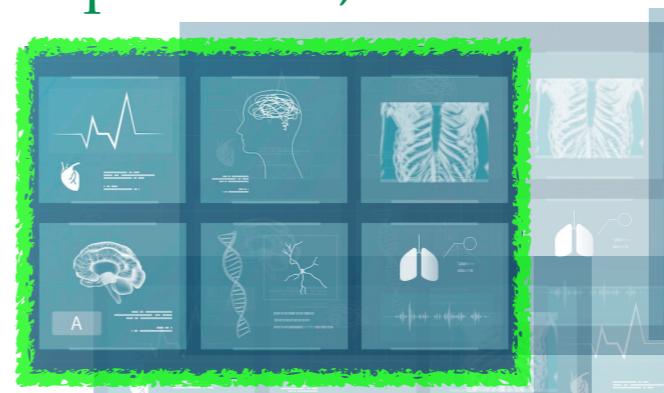


person B, t=4



Test data

person A, t=2

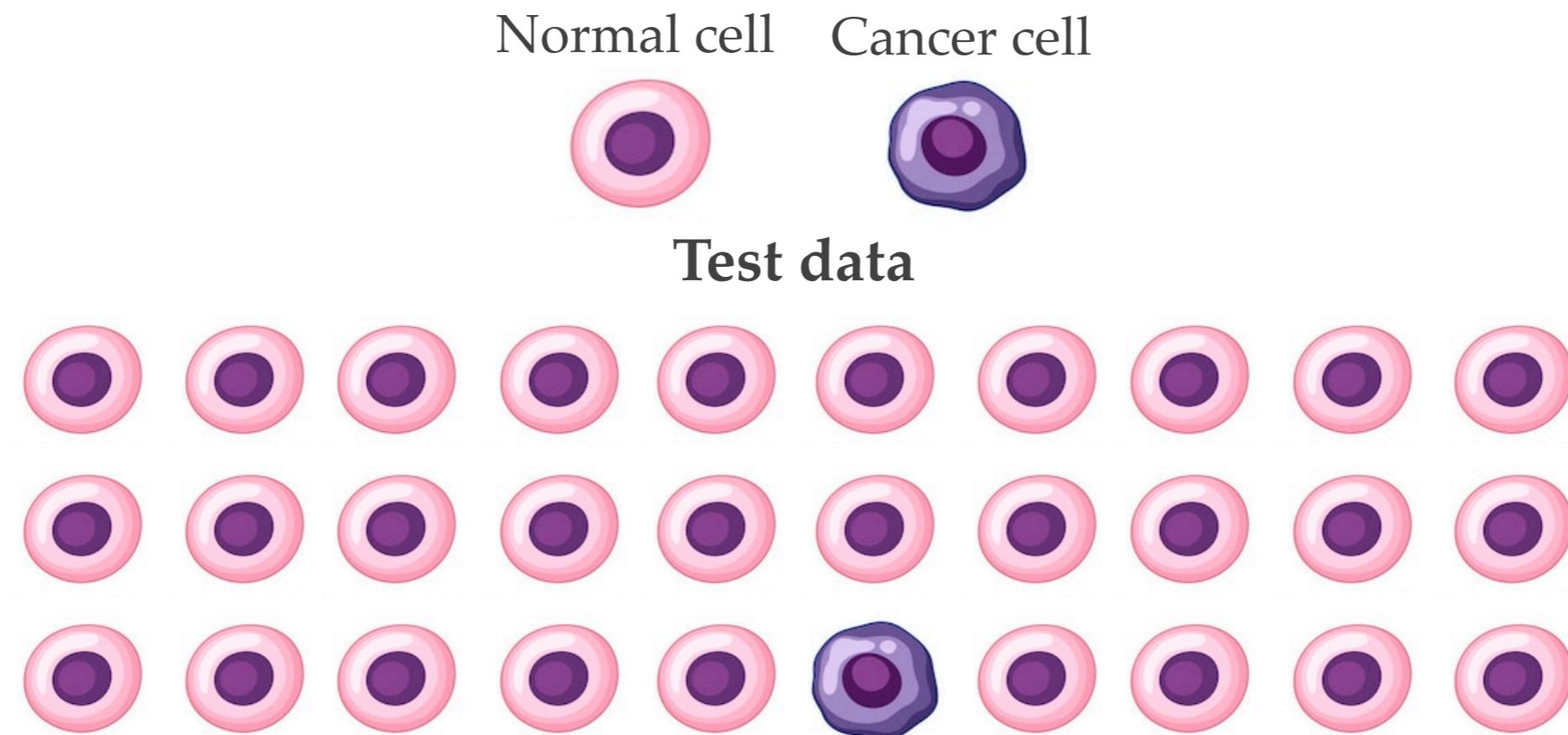


person B, t=1



Images from the same person appear in both the training and test data (with different t) 22

Pitfalls: Reporting a “Non-informative” Evaluation Metric

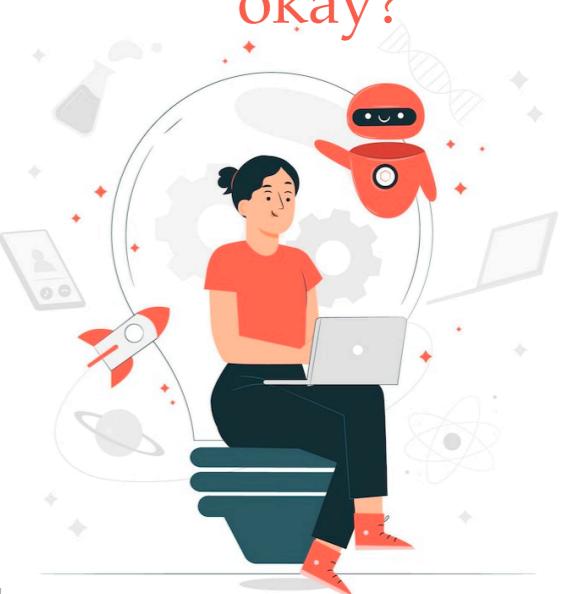


We got only one test sample wrong, and achieved the classification accuracy of **96.67%**!



This is much higher than the accuracy of a random guess!

Just say “normal cell”, okay?



Pitfalls: Solving “Irrelevant” Problems

MIT Technology Review

OPINION

Too many AI researchers think real-world problems are not relevant

The community's hyperfocus on novel methods ignores what's really important.

By Hannah Kerner

August 18, 2020

“Many studies applying machine learning to viticulture aim to optimize grape yields, but winemakers “want the right levels of sugar and acid, not just lots of big watery berries,” says Drake Whitcraft of Whitcraft Winery in California.”

“For example, most studies applying deep learning to echocardiogram analysis try to surpass a physician’s ability to predict disease. But predicting normal heart function would actually save cardiologists more time by identifying patients who do not need their expertise.”

Outline

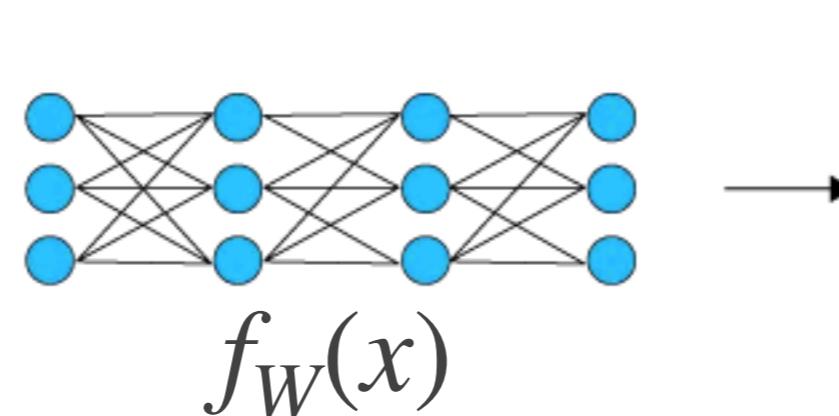
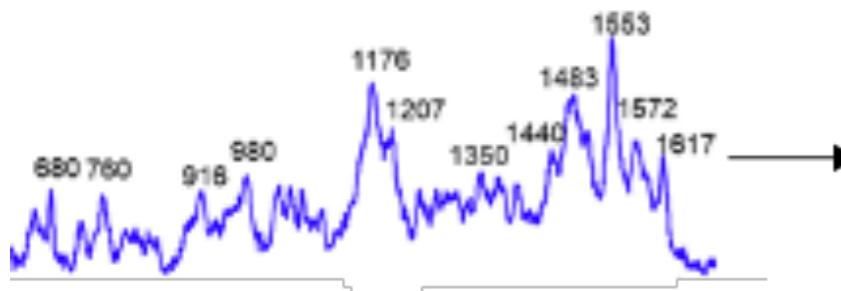
- ❖ Machine Learning Pitfalls
 - ❖ ML algorithms are not one-size-fits-all
 - ❖ Having more data doesn't always help
 - ❖ Exploiting spurious correlations
 - ❖ Training data don't capture real use cases
 - ❖ Having data leakage
 - ❖ Reporting a “non-informative” evaluation metric
 - ❖ Solving “irrelevant” problems
- ❖ A Way to Communicate with ML Practitioners
 - ❖ Be clear about what the inputs and outputs are
 - ❖ State the requirements and use-cases
 - ❖ Provide domain- / field-specific information
 - ❖ Summary of related papers
- ❖ Resources

A Way to Communicate with ML Practitioners

- Be clear about what the inputs and outputs are

Spectrum classification

Input: a list of numbers

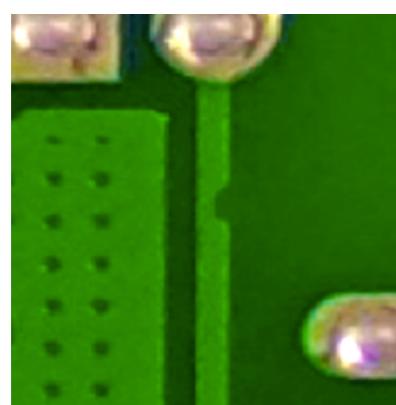


Output

Category

Object detection

Input: an image



Huang, Weibo, and Peng Wei. "A PCB dataset for defects detection and classification." arXiv preprint arXiv:1901.08204 (2019).

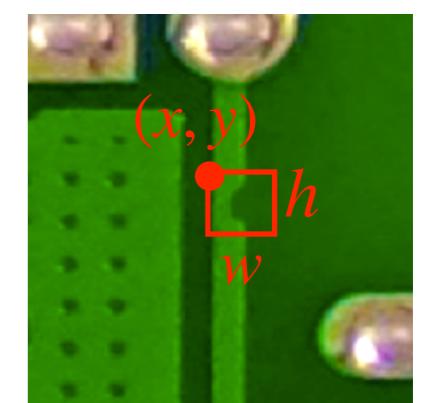


$$f_W(x)$$

Output: rectangular
box(es) and the
corresponding class(es)

$$\begin{bmatrix} x \\ y \\ w \\ h \end{bmatrix}$$

corresponding class



A Way to Communicate with ML Practitioners

- Be clear about what the inputs and outputs are

Molecule classification

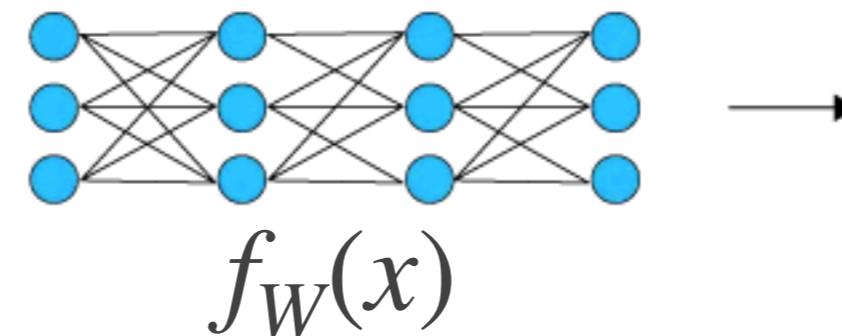
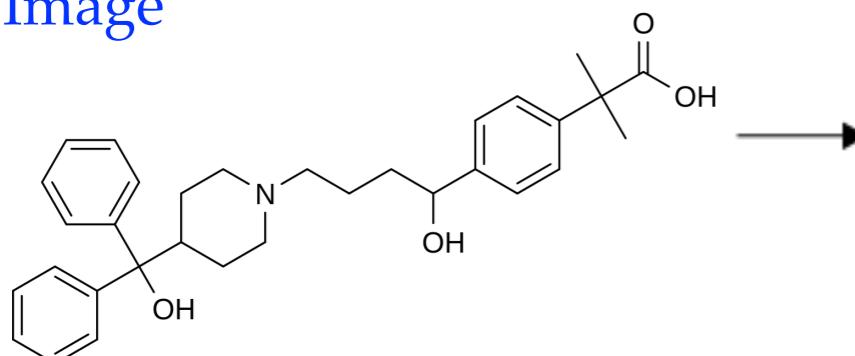
Input

String of characters

Output

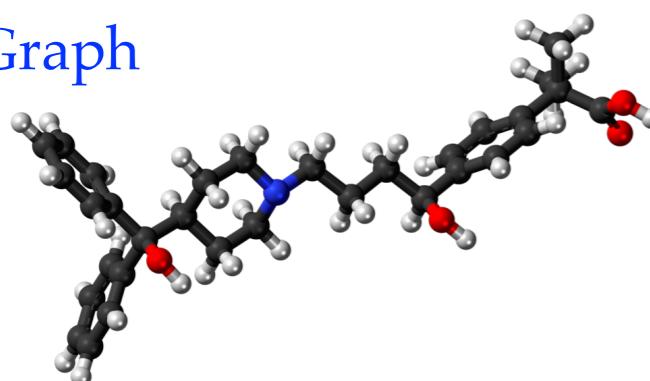
C₃₂H₃₉NO₄

Image



2nd-generation antihistamine drug

Graph



A Way to Communicate with ML Practitioners

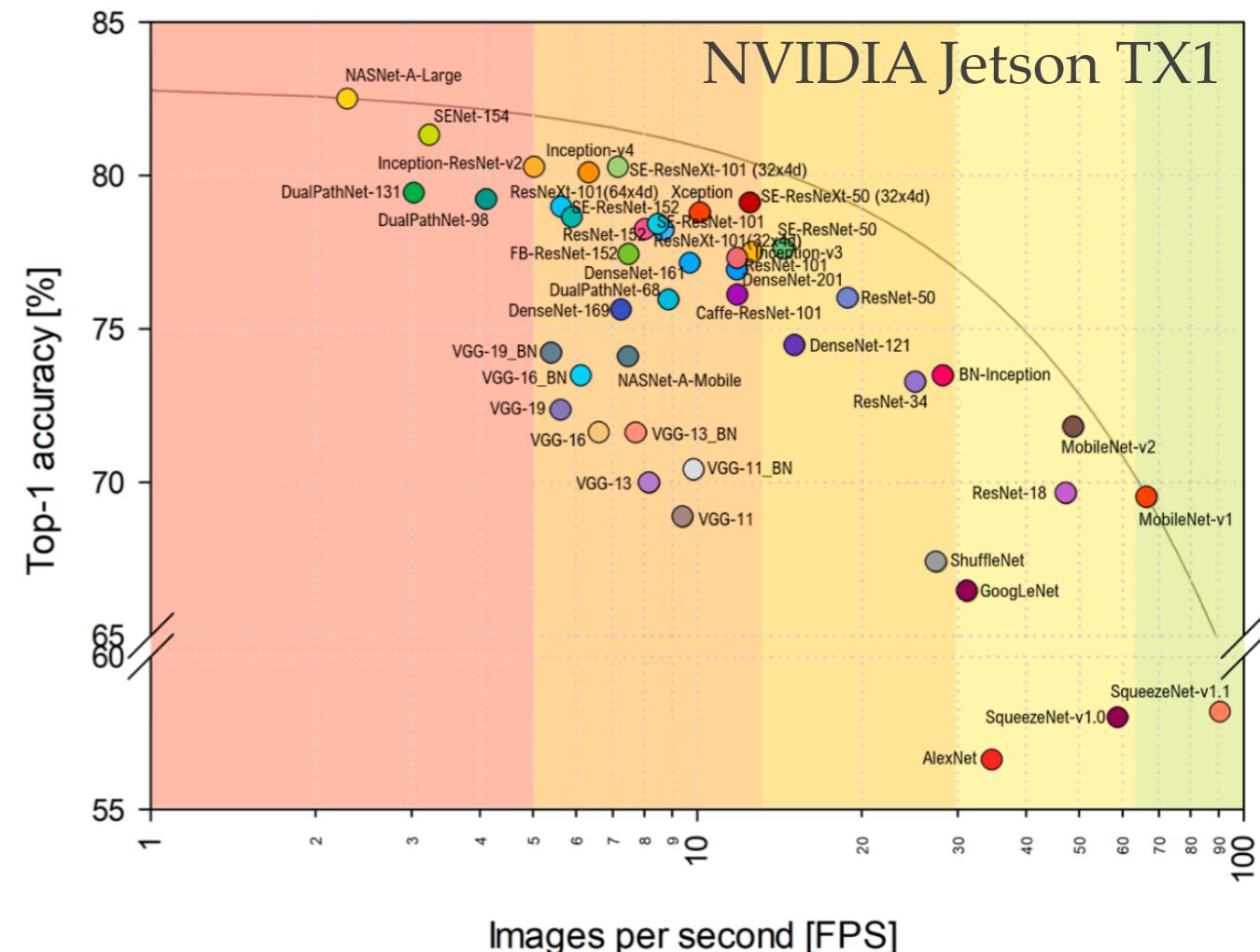
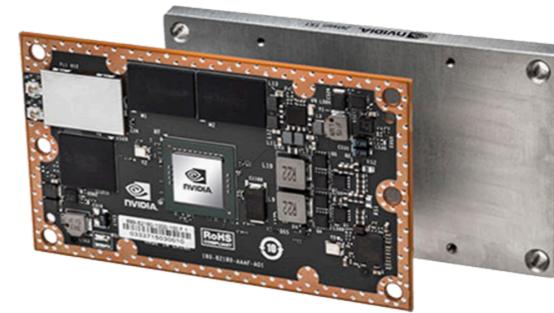
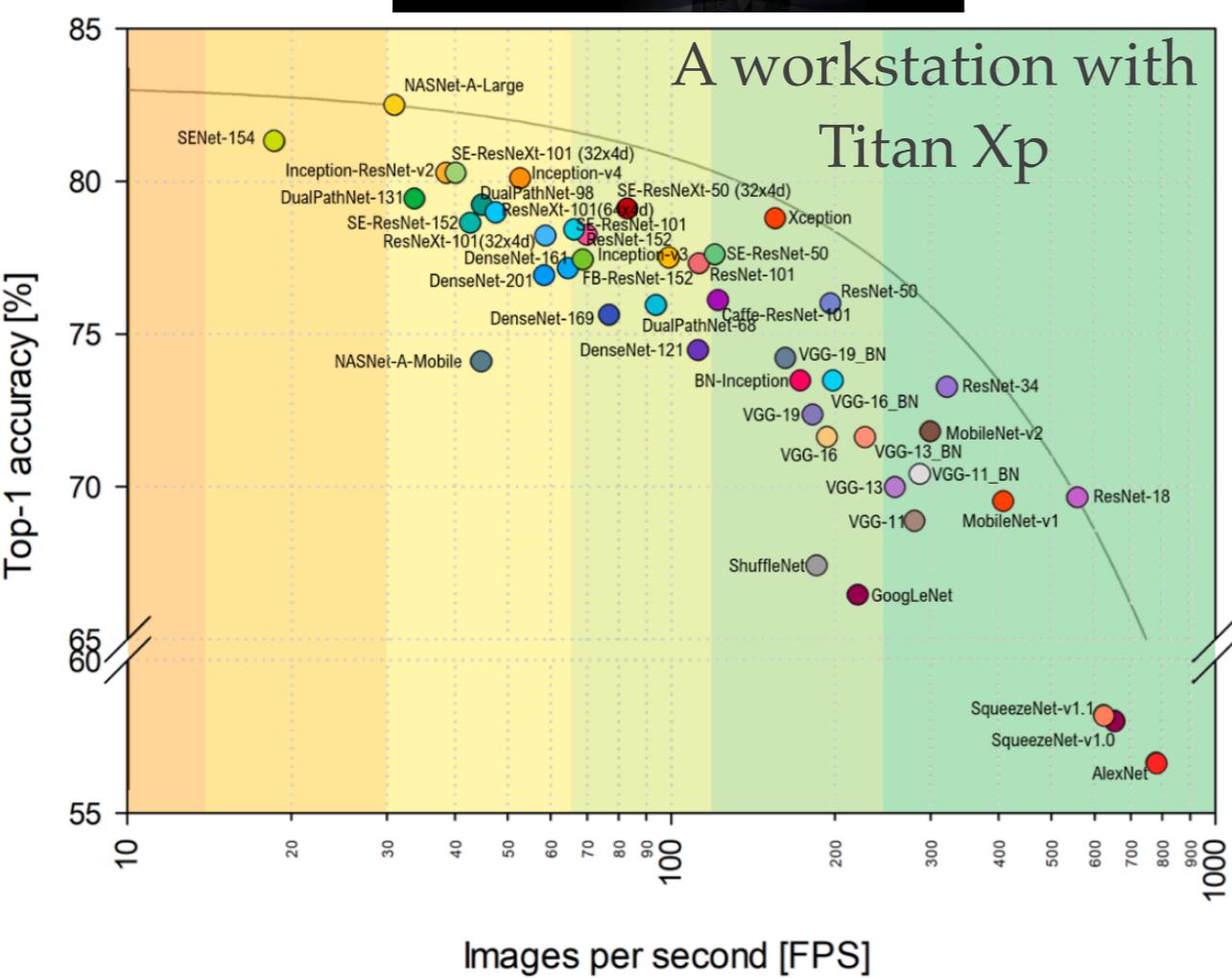
- State the requirements and use-cases
 - Computational resources: computer servers, workstations, IoT devices, smartphones
 - Model deployment
 - Web/mobile applications



- Required processing speed
 - How fast do you need it to run? How many samples per second? Real-time?

A Way to Communicate with ML Practitioners

- State the requirements and use-cases



The desired throughput can be used to pick a model on a specific platform

A Way to Communicate with ML Practitioners

- State the requirements and use-cases
 - Computational resources: computer servers, workstations, embedded systems, IoT devices, smartphones
 - Model deployment
 - Web/mobile applications



- Required processing speed
 - How fast do you need it to run? How many samples per second? Real-time?
- Number of samples (ideally with labels) that you can prepare to train a model

A Way to Communicate with ML Practitioners

- Only after the ML practitioners get the big picture (task, inputs, outputs) along with the requirements correctly, you can start providing domain-/ field-specific information
 - What could be a good feature for this task?
 - While there are methods that can extract feature automatically, having good / relevant handcrafted features is extremely helpful, especially when it is difficult to prepare large amount of data to train a model

A Way to Communicate with ML Practitioners

- Only after the ML practitioners get the big picture (task, inputs, outputs) along with the requirements correctly, you can start providing domain-/ field-specific information
 - What are meaningful evaluation metrics for your application?
 - Classification accuracy, false positive, false negative, ROC, AUC, sensitivity, specificity, f1-score, MSE, MAE, MAPE

Highly imbalanced data

Ex. Two-class classification (5 samples from class 1 and 95 samples from class 2)

A “model” that always classifies any sample as class 2 already achieves 95% accuracy

Is classification accuracy a good metric?

ATK



Which one do you consider worse?

1. Diagnose a healthy person as “positive”
2. Diagnose a person with COVID-19 as “negative”

A Way to Communicate with ML Practitioners

- If there are papers that tackle related / similar tasks, you can provide the ML practitioners with the references
 - It would be even better if you also provide a brief summary of the papers, possibly by answering some of these questions
 - What is the task?
 - What are the inputs and outputs? What are the features?
 - What machine learning methods do they use?
 - How many samples do they use for training and testing the method(s)?
 - How do they split the data? If they need to pick hyperparameters, do they use validation data?
 - What are the evaluation metrics that they use? Do they make sense?
 - How many times do they run their experiments?
 - What are the limitations of their methods? Frequently, many papers are able to report high performance because they restrict the use-cases

Outline

- ❖ Machine Learning Pitfalls
 - ❖ ML algorithms are not one-size-fits-all
 - ❖ Having more data doesn't always help
 - ❖ Exploiting spurious correlations
 - ❖ Training data don't capture real use cases
 - ❖ Having data leakage
 - ❖ Reporting a “non-informative” evaluation metric
 - ❖ Solving “irrelevant” problems
- ❖ A Way to Communicate with ML Practitioners
 - ❖ Be clear about what the inputs and outputs are
 - ❖ State the requirements and use-cases
 - ❖ Provide domain- / field-specific information
 - ❖ Summary of related papers
- ❖ Resources

Resources

Conferences



This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Ashish Vaswani^{*}
Google Brain
avaswani@google.com

Noam Shazeer^{*}
Google Brain
noam@google.com

Niki Parmar^{*}
Google Research
nikip@google.com

Jakob Uszkoreit^{*}
Google Research
usz@google.com

Llion Jones^{*}
Google Research
llion@google.com

Aidan N. Gomez^{* †}
University of Toronto
aidan@cs.toronto.edu

Lukasz Kaiser^{*}
Google Brain
lukasz.kaiser@google.com

Illia Polosukhin^{* ‡}
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrent and convolutional entirely. Experiments on two machine translation tasks show that our models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

1 Introduction

Recurrent neural networks, long short-term memory [12] and gated recurrent [7] neural networks in particular, have firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [29, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [31, 21, 13].

^{*}Figure 1 shows training order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed implementation, took care of all the mechanics of the original codebase, and tested many variants. Aidan also experimented with many model variants, was responsible for our earlier codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2Tensor, greatly improving results and massively accelerating our research.

[†]Work performed while at Google Brain.

[‡]Work performed while at Google Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
(kahe, v-xiangyu, v-shren, jianjun)@microsoft.com

Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. As an example, we introduce the “ResNet” architecture, based solely on identity mappings, dispensing with recurrent and convolutional entirely. Experiments on two machine translation tasks show that our models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

1. Introduction

Deep convolutional neural networks [22, 21] have led to a series of breakthroughs for image classification [21, 49, 39]. Deep networks naturally integrate low/mid/high-level features [49] and classify in an end-to-end multi-layer fashion. One of the main challenges is how to be able to learn deeper networks. Recently, evidence [40, 43] reveals that network depth is of crucial importance, and the leading results [40, 43, 12, 16] on the challenging ImageNet dataset [35] all exploit “very deep” [40] models, with a depth of sixteen [40] to thirty [16]. Many other non-trivial visual recognition tasks [7, 11, 6, 32, 27] have also

¹<http://image-net.org/challenges/LSVRC/2015/> and <http://mscoco.org/dataset/#detections-challenge2015>.

770



We gratefully acknowledge support from the Simons Foundation and member institutions.

Login

Search... All fields Help | Advanced Search

COVID-19 Quick Links

See COVID-19 SARS-CoV-2 preprints from

- [arXiv](#)
- [medRxiv](#) and [bioRxiv](#)

Important: e-prints posted on arXiv are not peer-reviewed by arXiv; they should not be relied upon without context to guide clinical practice or health-related behavior and should not be reported in news media as established information without consulting multiple experts in the field.

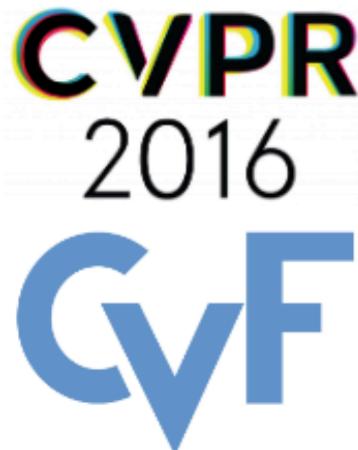
Subject search and browse:

Physics	<input type="button" value="Search"/>
<input type="button" value="Form Interface"/>	<input type="button" value="Catchup"/>

News

Read about recent news and updates on [arXiv's blog](#). (View the former "what's new" pages here). Read [robots beware](#) before attempting any automated download.

Resources



CVPR 2016 open access

These CVPR 2016 papers are the Open Access versions, provided by the [Computer Vision Foundation](#). Except for the watermark, they are identical to the accepted versions; the final published version of the proceedings is available on [IEEE Xplore](#).

This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright.

Deep Residual Learning for Image Recognition

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778

Cited 134,083 times (as of September 2022)

Attention is All you Need

Part of [Advances in Neural Information Processing Systems 30 \(NIPS 2017\)](#)

[Bibtex](#) [Metadata](#) [Paper](#) [Reviews](#)

Authors

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin

Cited 52,002 times (as of September 2022)

Resources

Conferences

Attention Is All You Need

Ashish Vaswani^{*}
Google Brain
avasani@google.com

Noam Shazeer^{*}
Google Brain
noam@google.com

Niki Parmar^{*}
Google Research
nikip@google.com

Jakob Uszkoreit^{*}
Google Research
usz@google.com

Llion Jones^{*}
Google Research
llion@google.com

Aidan N. Gomez^{* †}
University of Toronto
aidan@cs.toronto.edu

Lukasz Kaiser^{*}
Google Brain
lukasz.kaiser@google.com

Illia Polosukhin^{* †}
illia.pолосухин@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a simple model network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrent and convolutional entirely. Experiments on two machine translation tasks show that our models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

1 Introduction

Recent neural networks, long short-term memory [12] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [29, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [31, 21, 13].

Figure 1 shows training order is random. Jaksic proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed implementation, tested and evaluated the first prototypes of the original Transformer and tens of thousands of variants. Lukasz and Aidan experimented with many model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2Tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

^{*}Work performed while at Google Brain.

[†]Work performed while at Google Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.



This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
(kahe, v-xiangyu, v-shren, jianun)@microsoft.com

Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. As an example, we introduce residual blocks into deep neural networks without significantly increasing the computational cost. This allows us to build networks that are 10 times deeper than before. On the ImageNet dataset, we show that a residual network with 152 layers—8x deeper than VGG [16]—can have a top-5 test error rate of 3.57% on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task. We also present analysis on ImageNet with 100 and 1000 layers.

The depth of representations is of central importance for many visual recognition tasks. Solely due to its extremely deep nature, we believe that residual networks will also be competitive on the COCO object detection task. Deep residual nets are foundations of our submissions to ILSVRC & COCO 2015 competitions¹, where we also won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.

1. Introduction

Deep convolutional neural networks [22, 21] have led to a series of breakthroughs for image classification [21, 49, 39]. Deep networks naturally integrate low/mid/high-level features [49] and classify in an end-to-end multi-layer fashion. The depth of a network can be measured by the number of stacked layers (depth). Recent evidence [40, 43] reveals that network depth is of crucial importance, and the leading results [40, 43, 12, 16] on the challenging ImageNet dataset [35] all exploit “very deep” [40] models, with a depth of sixteen [40] to thirty [16]. Many other non-trivial visual recognition tasks [7, 11, 6, 32, 27] have also

¹<http://image-net.org/challenges/LSVRC/2015/> and <http://mscoco.org/dataset/#detections-challenge2015>.

770



Cornell University



arXiv

We gratefully acknowledge support from the Simons Foundation and member institutions.

Login

Search... All fields Search Help | Advanced Search

COVID-19 Quick Links

See COVID-19 SARS-CoV-2 preprints from

- arXiv
- medRxiv and bioRxiv

Important: e-prints posted on arXiv are not peer-reviewed by arXiv; they should not be relied upon without context to guide clinical practice or health-related behavior and should not be reported in news media as established information without consulting multiple experts in the field.

arXiv is a free distribution service and an open-access archive for 2,131,058 scholarly articles in the fields of physics, mathematics, computer science, quantitative biology, quantitative finance, statistics, electrical engineering and systems science, and economics. Materials on this site are not peer-reviewed by arXiv.

Subject search and browse:

Physics Form Interface Catchup

News

Read about recent news and updates on arXiv's blog. (View the former "what's new" pages here). Read robots beware before attempting any automated download.

Courses

CS229

CS229: Machine Learning Instructor



Course Description This course provides a broad introduction to machine learning and statistical pattern recognition. Topics include: supervised learning (generative/discriminative learning, parametric/non-parametric learning, neural networks, support vector machines); unsupervised learning (clustering, dimensionality reduction, kernel methods); learning theory (bias/variance tradeoffs, practical advice); reinforcement learning and adaptive control. The course will also discuss recent applications of machine learning, such as to robotic control, data mining, autonomous navigation, bioinformatics, speech recognition, and text and web data processing.

CS230 Deep Learning

Deep Learning is one of the most highly sought after skills in AI. In this course, you will learn the foundations of Deep Learning, understand how to build neural networks, and learn how to lead successful machine learning projects. You will learn about Convolutional networks, RNNs, LSTM, Adam, Dropout, BatchNorm, Xavier/He initialization, and more.

Syllabus Ed Lecture videos (Canvas)

Lecture videos (Fall 2018)

CS231n Home Course Notes Coursework Schedule Office Hours Final Projects Ed

CS231n: Deep Learning for Computer Vision

Course Description

Computer Vision has become ubiquitous in our society, with applications in search, image understanding, apps, mapping, medicine, drones, and self-driving cars. Core to many of these applications are visual recognition tasks such as image classification, localization and detection. Recent developments in neural network (aka “deep learning”) approaches have greatly advanced the performance of these state-of-the-art visual recognition systems. This course is a deep dive into the details of deep learning architectures with a focus on learning end-to-end models for these tasks, particularly image classification. During the 10-week course, students will learn to implement and train their own neural networks and gain a detailed understanding of cutting-edge research in computer vision. Additionally, the final assignment will give them the opportunity to train and apply multi-million parameter networks on real-world vision problems of their choice. Through multiple hands-on assignments and the final course project, students will acquire the toolset for setting up deep learning tasks and practical engineering tricks for training and fine-tuning deep neural networks.

INTRODUCTION

2110573 Pattern Recognition



Pattern Recognition 2022: L01-Introduction
🕒 Unlisted
1:163 views Streamed live on Jan 13, 2022 0:00 pre-stream
18:00 Introduction & course logistics
18:07 Technology trends and machine learning
18:27 A&ML&PR Terminologies
1:51:11 Introduction to supervised learning
2:06:18 Typical workflow in machine learning
2:12:49 Feature extraction
2:33:29 Metrics in prediction problems (Precision, recall, accuracy, ...)
3:04:35 In class exercise
3:06:39 In class solution

Textbooks

Blog posts

Journals

YouTube videos

Tutorials

Outline

- ❖ Machine Learning Pitfalls
 - ❖ ML algorithms are not one-size-fits-all
 - ❖ Having more data doesn't always help
 - ❖ Exploiting spurious correlations
 - ❖ Training data don't capture real use cases
 - ❖ Having data leakage
 - ❖ Reporting a “non-informative” evaluation metric
 - ❖ Solving “irrelevant” problems
- ❖ A Way to Communicate with ML Practitioners
 - ❖ Be clear about what the inputs and outputs are
 - ❖ State the requirements and use-cases
 - ❖ Provide domain- / field-specific information
 - ❖ Summary of related papers
- ❖ Resources

Q&A

