Why Machine Learning Models Fail: A Benchmarking Perspective

Dissertation
der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von

Claudio Michaelis
aus München

Tübingen
2022

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	19.12.2023
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter:	Prof. Dr. Matthias Bethge
2. Berichterstatter:	Prof. Dr. Fabian Sinz

Ich erkläre, dass ich die zur Promotion eingereichte Arbeit mit dem Titel:
Why Machine Learning Models Fail: A Benchmarking Perspective
selbstständig verfasst, nur die angegebenen Quellen und Hilfsmittel benutzt und wörtlich oder inhaltlich übernommene Stellen als solche gekennzeichnet habe. Ich versichere an Eides statt, dass diese Angaben wahr sind und dass ich nichts verschwiegen habe. Mir ist bekannt, dass die falsche Abgabe einer Versicherung an Eides statt mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft wird.

Tübingen, den ____________________ ____

Datum Unterschrift

Summary

Over the last years, machine performance at object recognition, language understanding and other capabilities that we associate with human intelligence has rapidly improved. One central element of this progress are machine learning models that learn the solution for a task directly from data. The other are benchmarks that use data to quantitatively measure model performance. In combination, they form a virtuous cycle where models can be optimized directly on benchmark performance. But while the resulting models perform very well on their benchmarks, they often fail unexpectedly outside the controlled setting. Innocuous changes such as image noise, rain or the wrong background can lead to wrong predictions. In this dissertation, I argue that to understand these failures, it is necessary to understand the relationship between benchmark performance and the desired capability. To support this argument, I study benchmarks in two ways.

In the first part, I investigate how to learn and evaluate a new capability. Therefore, I introduce one-shot object detection and define different benchmarks to analyze what makes this task hard for machine learning models and what is needed to solve it. I find that CNNs struggle to separate individual objects in cluttered environments, and that one-shot recognition of objects from novel categories can be challenging with real-world objects. I then continue to investigate what makes one-shot generalization difficult in real-world scenes, and identify the number of categories in the training dataset as the central factor. Using this insight, I show that excellent one-shot generalization can be achieved by training on broader datasets. These results highlight how much benchmark design influences what is measured, and that limitations in benchmarks can be confused for limitations of the models developed with them.

In the second part, I broaden the view and analyze the connection between model failures in different areas of machine learning. I find that many of these failures can be explained by shortcut learning, models exploiting a mismatch between a benchmark and its associated capability. Shortcut solutions use superficial cues that work very well within the training domain, but are unrelated to the capability. This demonstrates that good benchmarks performance is not sufficient to prove that a model acquired the associated capability, and that results have to be interpreted carefully.

Taken together, these findings put in question the common practice of evaluating models on a single, or at maximum a few, benchmarks. Rather, my results indicate that to anticipate model failures, it is essential to measure broadly. And to avoid them, it is necessary to verify that models acquire the desired capability. This will require investment into better data, new benchmarks and other complementary forms of evaluation, but provides the basis for further progress towards powerful, reliable and safe models.

Zusammenfassung

In den letzten Jahren hat sich die Leistung von Maschinen bei der Erkennung von Objekten, dem Verstehen von Sprache und anderen Fähigkeiten, die wir mit menschlicher Intelligenz in Verbindung bringen, rapide verbessert. Ein zentrales Element dieses Prozesses sind Modelle des maschinellen Lernens, die die Lösung einer Aufgabe direkt von Daten lernen. Das andere sind Benchmarks, die die Leistung von Modellen anhand von Daten quantitativ messen. In Kombination bilden sie einen sich gegenseitig verstärkenden Kreislauf, bei dem die Modelle direkt anhand der Benchmark-Leistung optimiert werden können. Doch während die resultierenden Modelle in ihren Benchmarks sehr gut funktionieren, versagen sie oft unerwartet außerhalb der kontrollierten Umgebung. Scheinbar unbedeutende Veränderungen wie Rauschen in Bildern, Regen oder ein ungewöhnlicher Hintergrund können zu falschen Vorhersagen führen. In dieser Dissertation argumentiere ich, dass zum Verständnis dieser Fehler die Beziehung zwischen der Leistung auf Benchmarks und der gewünschten Fähigkeit notwendig ist. Zur Unterstützung dieses Arguments untersuche ich Benchmarks auf zwei Arten.

Im ersten Teil betrachte ich, wie eine neue Fähigkeit gelernt und gemessen werden kann. Dazu führe ich Objekterkennung auf Basis eines einzelnen Beispiels (One-Shot) ein und analysiere anhand verschiedener Benchmarks, was diese Aufgabe für maschinelle Lernmodelle schwierig macht und was zu ihrer Lösung erforderlich ist. Ich stelle fest, dass CNNs in unübersichtlichen Umgebungen Mühe haben Objekte zu trennen, und dass die One-Shot Erkennung von Objekten aus neuen Kategorien in realen Szenen eine große Herausforderung darstellt. Anschließend untersuche ich, was die One-Shot-Generalisierung in realen Szenen so schwierig macht, und identifiziere die Anzahl der Kategorien im Trainingsdatensatz als zentralen Faktor. Auf der Grundlage dieser Erkenntnis zeige ich, dass durch Training auf breiten Datensätzen eine hervorragende One-Shot-Generalisierung erreicht werden kann. Diese Ergebnisse verdeutlichen, wie sehr das Design von Benchmarks die Ergebnisse beeinflusst und dass Limitationen von Benchmarks mit Limitationen der auf ihnen entwickelten Modelle verwechselt werden können.

Im zweiten Teil weite ich den Blick und analysiere den Zusammenhang zwischen Fehlern von Modellen in unterschiedlichen Bereichen des maschinellen Lernens. Ich stelle fest, dass viele dieser Fehlschläge durch Abkürzungen beim Lernen erklärt werden können, d. h. dadurch, dass Modelle die Diskrepanz zwischen einem Benchmark und der damit verbundenen Fähigkeit ausnutzen. Solche Lösungen nutzen scheinbare Anhaltspunkte, die im Training sehr gut funktionieren, aber nicht der Fähigkeit entsprechen. Dies zeigt, dass eine gute Leistung auf einem Benchmark nicht als Beweis dafür ausreicht, dass ein Modell die zugehörige Fähigkeit erworben hat, und Ergebnisse mit Vorsicht zu bewerten sind.

Diese Ergebnisse stellen die gängige Praxis in Frage, Modelle anhand eines einzigen oder höchstens einiger weniger Benchmarks zu bewerten. Vielmehr deuten sie darauf hin, dass es zur Vorhersage von Fehlern unerlässlich ist, umfassend zu testen. Um Fehler zu vermeiden, muss sichergestellt werden, dass das Modell die gewünschte Fähigkeit erlangt hat. Dies erfordert bessere Daten, neue Benchmarks und ergänzende Formen der Evaluation, welche wiederum die Grundlage für die Entwicklung leistungsfähiger, zuverlässiger und sicherer Modelle bilden.

Publications

Publications in this dissertation

- **Claudio Michaelis**, Matthias Bethge, Alexander S. Ecker. One-shot Segmentation in Clutter. *ICML 2018*

- **Claudio Michaelis**, Ivan Ustyuzhaninov, Matthias Bethge, Alexander S. Ecker. One-shot Instance Segmentation. *ArXiv 2018*

- **Claudio Michaelis**, Matthias Bethge, Alexander S. Ecker. A Broad Dataset is All You Need for One-Shot Object Detection. *ArXiv 2020*

- Robert Geirhos*, Jörn-Henrik Jacobsen*, **Claudio Michaelis***, Richard Zemel, Wieland Brendel, Matthias Bethge & Felix A. Wichmann. Shortcut Learning in Deep Neural Networks. *Nature Machine Intelligence 2: 665–673 (2020)*

Detailed contribution statements for the papers in this dissertation can be found in the corresponding sections 3.4, 3.5, 3.6 & 4.2.

Publications I co-authored during my PhD, which are not part of this dissertation

- Ivan Ustyuzhaninov, **Claudio Michaelis**, Wieland Brendel, and Matthias Bethge. One-shot Texture Segmentation. *ArXiv 2018*

- Robert Geirhos, Patricia Rubisch, **Claudio Michaelis**, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *ICLR 2019 (Oral)*

- **Claudio Michaelis**, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S. Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *NeurIPS Workshop on Machine Learning for Autonomous Driving 2019*

- Michal Rolínek, Vít Musil, Anselm Paulus, Marin Vlastelica, **Claudio Michaelis**, Georg Martius. Optimizing Rank-based Metrics with Blackbox Differentiation. *CVPR 2020 (Oral, Best Paper Runner Up)*

- Kai J Sandbrink, Pranav Mamidanna, **Claudio Michaelis**, Mackenzie Weygandt Mathis, Matthias Bethge, Alexander Mathis. Contrasting action and posture coding with hierarchical deep neural network models of proprioception. *eLife 2023*

Contents

Seht ihr das auch?
Könnt ihr das auch sehen?
Deichkind 2019

1 Introduction

In the last years, the field of Artificial Intelligence (AI) has made a lot of progress due to the virtuous combination of machine learning and standardized benchmarks. Machine learning allows learning tasks from examples. Standardized benchmarks allow measuring the performance of the resulting models in a quantitative and comparable fashion. Their combination enabled developing models that can accomplish tasks we associate with intelligence, like identifying images with cats or translating text. Computers can now process visual information, language, and speech to a degree that had been beyond imagination for long. Today, machines learning models are used for tasks like translating websites or driving cars, that previously only humans could do.

But despite all successes, machines are still farm from achieving human-level capabilities. Machine learning systems need millions of examples, where humans may only need one [Lake et al., 2015]. Object recognition models fail when small amounts of noise are added to the images [Geirhos et al., 2018; Hendrycks and Dietterich, 2019]. And the perception algorithms of self-driving cars struggle in bad weather conditions [Zhang et al., 2021b; Yurtsever et al., 2020].

In this dissertation, I argue, that the difference between narrow benchmark objectives and the broad human capabilities they are associated with plays a significant role in this human-machine gap. Take the following example: We may think object recognition tasks such as spotting a zebra are straightforward. But if one morning a zebra was standing in your kitchen, you would likely freak out. In contrast, an object recognition model would probably miss the zebra, because of the unfamiliar background [Beery et al., 2018; Kolesnikov et al., 2019]. This may be a contrived example, but it illustrates that there is much more to object recognition than meets the eye. Context, emotions, experiences, abstract knowledge all influence our perception, mostly without us even noticing.

It is clear that current benchmarks and models do not cover all human capabilities, and that there are many things that could be added. But, the assumption behind my thesis is that the current benchmarks miss much more than we may expect. I approach this from two sides: In the first part of the dissertation, I introduce benchmarks for a new capability, one-shot object detection. Using this example, I demonstrate how benchmark design influences which aspects of a capability are measured and that insights from studying different benchmarks can help to identify and overcome central challenges of a capability in ways that would not have been clear from a single benchmark. In the second part of the dissertation, I zoom out and analyze surprising model failures across different areas of machine learning. This systematic analysis exposes shortcut learning, models exploiting a mismatch between a benchmark and the desired capability, as a widely appearing phenomenon that can explain many model failures.

2 Benchmarks

Generally, a benchmark is a way to measure something, from business metrics to computer speed, in a way that is quantifiable and comparable. It is basically an exam for machines or other systems: Everyone gets the same questions, answers are scored according to the same rules and the final result is a score that can be used to compare and rank.

In AI, benchmarks are used to evaluate broad **capabilities** that are not directly measurable, such as recognizing objects in images (object recognition). To make these broad capabilities measurable, **tasks** are defined that have fixed rules and a success criterion. For example, the most common object recognition task is to classify images according to the foreground object they show (image classification). A task is more specified than a capability, but it does not yet fix all the details, such as for example whether dogs, cats, zebras, or coffee makers should be classified. A **benchmarks** then provides the precise problem and metric used for evaluation. This is often done by providing a dataset, and the metric then measures how well a model can make predictions on that dataset. For example, in image classification the dataset usually is a collection of images with labels (e.g., cat, dog, or zebra) and the typical metric is the accuracy with which a model predicts the right label for each image.

So a definition could be: *A benchmark is the attempt to measure performance at a capability in a quantifiable and comparable fashion.* A broad capability that intuitively makes sense but can be hard to define precisely, such as object recognition, is turned into a measurable quantity, such as the classification accuracy of images with cats and dogs.

To ensure results are comparable, the evaluation protocols have to be well-defined and in the best case unambiguous (standardized benchmark). Standardized does not imply that a benchmark has to be deterministic. Evaluation protocols can include stochastic or dynamic elements, as long as the results are comparable. Similarly, the dataset does not have to be fixed. Instead, it can also use a protocol or environment that are sufficiently specific to allow a fair comparison. For example, computer games can be used as benchmarks for interactive agents [Bellemare et al., 2013].

2.1 Benchmarking One-Shot Object Detection

To make the connection between benchmarks, tasks, and capabilities more concrete, let us explore an example: One-shot object detection describes the capability of recognizing and localizing objects that have been seen only once. Human perception has this ability. Say your roommate bought a new coffee maker, which you briefly saw when he unpacked it. You will have no problem identifying the coffee machine in the next morning. And you could even do it if you found a zebra standing in your kitchen (once you have overcome the shock). This may not seem special, but the ability to recognize objects from a few or even a single example is remarkable, and is used it a lot in people's everyday lives. Machines with this capability could be easily instructed and hence used for many different applications.

There are many ways to turn this capability int a measurable task. So let us go through the coffee maker in kitchen scenario to identify its components? First, the reference is visual. You have seen the coffee maker the day before, not only heard that there is a new one. Second, the coffee maker is located within a larger scene full of known objects, like your toaster and dishwasher, and potential unknown objects, like a zebra. Finally, the goal is to localize the coffee maker within the kitchen. Therefore, the problem is localization of an object in a scene based on one visual example.

When I started the research for this dissertation, no benchmarks existed with such a task. Existing benchmarks that focused on detecting objects in scenes provided thousands of training examples per category (e.g., [Lin et al., 2014]). And benchmarks which targeted recognition from one example, so-called one-shot learning, used a task that required on classifying tiny images showing a single object into one of five novel categories (e.g., [Lake et al., 2015; Vinyals et al., 2016]). Therefore, in this dissertation, I introduce a new task for this problem: Given an image of an object, find all objects of the same type in a scene. So given an image of your kitchen, the model is expected to find the new coffee maker if it is shown an image of a coffee maker and the dishwasher if it is shown an image of a dishwasher.

Benchmarks can now be defined upon this task by specifying a dataset, for example images of kitchens annotated with coffee makers, dishwashers and the occasional zebra, as well as an evaluation procedure and metric, for example if all coffee makers are found.

2.2 Why Research Benchmarks?

Benchmarking is a central element of AI research. As in other areas of science, "Measurement has been a key factor for progress" [Welty et al., 2019]. Quantitative problems have clear metrics with which different methods can easily be compared. This allows incremental hill-climbing, an unspectacular but highly efficient way to make progress on a problem. Through leaderboards and challenges, this form of development has been made fun by stylizing it as a game-like competition (gamification [Deterding et al., 2011]).

In machine learning (ML). benchmarks often also serve other purposes beyond evaluation. Many benchmarks provide training data or environments that are widely used. This makes results more comparable and enables rapid model development because the tedious task of data collection can be skipped. Often these benchmark ecosystems do not only benefit progress towards the specific benchmark, but are used for other problems as well. New benchmarks can use the same toolkits, data formats, metrics, and environments. And large datasets can be used to pre-train models, which then can be applied to other problems via transfer learning [Donahue et al., 2014; Devlin et al., 2019].

But while benchmarks are widely used in model development, they themselves are investigated much more rarely [Sculley et al., 2018; Dehghani et al., 2021; Lipton and Steinhardt, 2019; Sambasivan et al., 2021; Marie et al., 2021; Church and Hestness, 2019]. A number of studies on benchmarks and other evaluation methods exist (e.g., [Torralba and Efros, 2011; Recht et al., 2019; Shankar et al., 2020; Beyer et al., 2020; Bowman and Dahl, 2021; Balduzzi et al., 2018; Dodge et al., 2019; Welty et al., 2019; Bouthillier et al., 2021]), but they are far less numerous than modelling studies. A major reason for this neglect of benchmarking research are bad incentives that encourage researchers to focus on achieving state-of-the-art performance on existing benchmarks [Dehghani et al., 2021; Lipton and Steinhardt, 2019; Sculley et al., 2018; Lin, 2019; Dacrema et al., 2019; Wagstaff, 2012; Church, 2017]. This would be ok, if studies mostly found benchmarks to work well. But a large fraction find major issues and red flags in current evaluation practices (see for example [Dacrema et al., 2019; Marie et al., 2021; Musgrave et al., 2020; Ponce et al., 2006; Torralba and Efros, 2011; Northcutt et al., 2021; Lewis et al., 2021]).

2.3 Research Question and Methodology

In this dissertation, I investigate how benchmarks are related to shortcomings and failures of current machine learning models. Specifically, I ask three questions. How well do benchmarks measure capabilities, and can a mismatch between benchmarks and capabilities explain model failures? How do benchmarks need to be designed, to identify model failures? And how can benchmarks help fix these failures?

To answer these questions, I use two methods. In the first part of the dissertation, I introduce a challenging new capability and inspect it up close. To do so, I introduce a series of benchmarks and compare how different models perform across them to gain insights into the benchmarks as well as the models. In contrast, the common approach is to compare different models on the same benchmark. As I demonstrate, the comparison of models across benchmarks can help get a deeper understanding of the problem, which in turn can be used to improve models in ways that would not have been obvious otherwise. In the second part of the thesis, I perform a systematic analysis of model failures across different areas of machine learning. This reveals a pattern of failure that arises in any benchmark, and suggests that the challenges and methods discussed in the first part are relevant in all areas of machine learning.

3 Part I: Benchmarking One-Shot Object Detection

This part of the dissertation uses a new capability, one-shot object detection, to explore how benchmark design influences which aspects of the underlying capability are measured. The next sections provide an overview over previous research into object recognition (Section 3.1) and learning from few examples (Section 3.2) that form the basis upon which the new one-shot object detection benchmarks are built. Then I describe the specific task setup and discuss how it compares to existing, similar tasks (Section 3.3). Finally, this part includes summaries of three publications (Section 3.4-3.6), that use different benchmarks to explore different aspects of one-shot object detection and use the insights to overcome a key challenge of the task, generalization to novel categories. The papers can be found in full length at the end of the dissertation.

3.1 Recognizing Objects: Object Detection & Segmentation

Object recognition is one of the central problems in computer vision and has been a dominant area of research in the last years [Lee and Qiufan, 2021].

3.1.1 Object Image Classification

The standard object recognition task is (object) image classification (Figure 1A). Classifying images based on the object or objects in them. The most common task uses images with a single foreground object per image that has to be assigned to one of a number of categories (forced choice), but tasks with multiple objects per scene exist as well (for example Figure 1A).

The two main challenges in classification are identifying the foreground object and predicting its category. Therefore, early benchmarks consisted of digits on black background [LeCun et al., 1998] or simple images that show only one object without a lot of background [Fei-Fei et al., 2004; Krizhevsky et al., 2009; Ponce et al., 2006]. A major milestone was ImageNet [Russakovsky et al., 2015] which consists of widely varying and sometimes chaotic web images that are annotated with one of over 21,000 categories that span most of the nouns in WordNet [Miller, 1995], a lexical database of all English words. A subset of the dataset with roughly 1.4 million images from a thousand categories that was selected for the 2012 ImageNet large scale visual recognition challenge (ILSVRC) is used today as the major dataset to train and evaluate image classification models. Due to the enormous scale, large set of categories and variety of images, ImageNet was seen as a major challenge at the time and led to the breakthrough of deep learning. Deep neural networks (DNN) that are trained end-to-end for the image classification task were better at handling this complexity than previous feature-based methods [Krizhevsky et al., 2012].

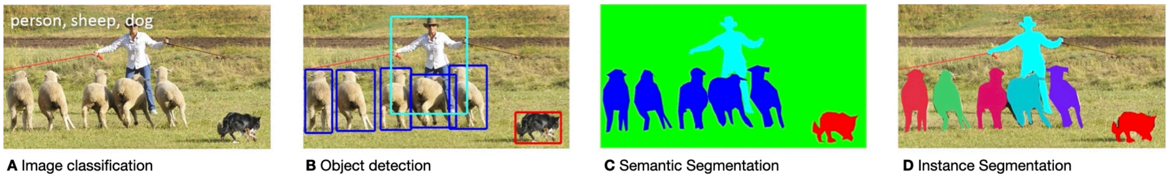


Figure 1: Illustration of the four main object recognition tasks (adapted from Lin et al. [2014]).

Most current computer vision approaches are built upon this work and follow roughly the same approach. Images are encoded with a DNN, often a pre-trained convolutional neural network (CNN) [LeCun et al., 1990], and predictions are made from this encoding. For image classification, the prediction can be made with a simple linear layer, the classifier. For other tasks, the prediction mechanism can be vastly more complex.

3.1.2 Object Detection

For many applications it is not only important to know what objects are in a scene but also where they are. Object detection is the task of localizing objects within scenes (Figure 1B). The most common task requires detecting all objects for a set of categories, and return a bounding box which tightly outlines the object and class label for each of them.

The main challenges are object discovery, variations in object size as well as clutter and occlusions. Older benchmarks like ImageNet Detection [Russakovsky et al., 2015] and Pascal VOC [Everingham et al., 2010] therefore consisted of scenes with only few objects, most of which were the focus of an image. The currently most important benchmark COCO [Lin et al., 2014] brought significantly more challenging scenes with many more objects and larger variations in object scale (See Table 1). A number of more recent benchmarks increase the difficulty by raising the number of objects and or categories per image (e.g., LVIS [Gupta et al., 2019] & ADE20k [Zhou et al., 2017]) or provide significantly more annotated images (e.g., OpenImages [Kuznetsova et al., 2020] & Objects365 [Shao et al., 2019]), but a clear successor has yet to emerge.

As for image classification, most modern detectors use a CNN (called the backbone) to encode the image. This encoding is then used to discover and classify objects. Object discovery can be done in many ways such as a sliding window approach [Felzenszwalb et al., 2009; Sermanet et al., 2014], by using pre-defined anchor boxes [Girshick et al., 2014; He et al., 2015b; Ren et al., 2015] or by directly regressing a set of bounding boxes [Szegedy et al., 2013; Redmon et al., 2016]. Object classification is performed similar to image classification, but on a per-object basis. Discovery and classification can happen in one (single-stage detector) or two steps (two-stage detector), the latter being more accurate but often slower [Huang et al., 2017]. The most popular model at the time of this dissertation is Faster R-CNN [Ren et al., 2015], a two-stage detector using anchor boxes to create object proposals.

3.1.3 Segmentation

If not only an object's location but also its shape is required, the task is segmentation, drawing the outlines of objects. Two main segmentation tasks are studied. Instance segmentation is closely related to object detection as individual instances of objects have to be identified but instead of a bounding box a segmentation mask has to be predicted (Figure 1D). Semantic segmentation in contrast does not require to separate individual instances of each object, but just to label each pixel in an image with a category (Figure 1C).

Instance segmentation shares most of the challenges with object detection, except that objects additionally have to be segmented. The benchmarks are mostly the same as for object detection because most datasets including COCO and Pascal VOC include segmentation masks. A number of solution approaches exist. The whole scene can be segmented first, and the segments then separated into individual objects [Bai and Urtasun, 2017; Kirillov et al., 2017; Arnab and Torr, 2017; Liu et al., 2017]. Alternatively, the scene can be processed in parts, either using mask proposals [Hariharan et al., 2014] or segmenting the image in a sliding window fashion [Pinheiro et al., 2015, 2016; Dai et al., 2016a; Chen et al., 2019c]. Finally, it is possible to segment bounding boxes predicted by an object detection model [Dai et al., 2016b; Li et al., 2017b; He et al., 2017]. In the last years this last object detection based approach has largely taken over the field, driven by the success of Mask R-CNN [He et al., 2017], a variant of Faster R-CNN which segments each predicted bounding box.

Semantic segmentation shares some challenges with instance segmentation, such as scale invariance, but has a stronger focus on segmentation quality. In addition to COCO and Pascal VOC, Cityscapes [Cordts et al., 2016] and ADE20k [Zhou et al., 2017] play a central role due to their high quality annotations that include a lot of small or delicate objects. For semantic segmentation, a per-pixel classification of the image is required. To achieve a detailed output, the encoded image is processed by a decoder that upsamples the downsampled CNN representation. Typical decoders consist of a cascade of upsampling steps which are interleaved with convolutional layers [Long et al., 2015]. A popular choice is to simply mirror the encoder [Ronneberger et al., 2015; Noh et al., 2015; Badrinarayanan et al., 2017]. Because convolutional and up- and down-sampling layers perform only local computations, these "fully convolutional" [Long et al., 2015] networks are independent of the image size and very easy to implement. To pass additional information about low level features and fine-grained structures to the decoder, skip connections between the encoder and decoder are commonly used [Ronneberger et al., 2015].

Dataset	Version	Categories	Images	Instances	Ins/Img	Cat/Img	Thr.
ImageNet	Det	200	332k	473k	1.4	1.1	✓
Pascal VOC	07+12	20	8k	23k	2.9	1.6	✓
COCO	2017	80	118k	860k	7.3	2.9	✓
Cityscapes		8	5k	50k	16.9	3.0	✓
OpenImages	v6	600	1.9M	15.8M	≥8.3	≥2.3	✗
Objects365	v2	365	1.9M	28M	14.6	6.1	✓
LVIS	v1	1,203	100k	1.27M	≥12.8	≥3.6	✗
ADE20k		3,688	27k	708k	25.7	9.9	✓

Table 1: Popular object detection datasets roughly sorted by image complexity, which depends on the number of object instances and categories per image (Ins/Img and Cat/Img). Throughout annotated (Thr.) means that every instance of every class is annotated in every image. Non exhaustively labelled datasets (Thr. = ✗) have potentially more objects and categories per image than are annotated.

3.1.4 State-of-the-Art

Performance on all object recognition tasks has improved significantly over the last years. A major factor for progress are improvements in CNN architecture and training [Xie et al., 2017; Tan and Le, 2019; Wang et al., 2020b; Gao et al., 2019; Zhang et al., 2020]. In object detection and segmentation, handling of objects at varying scales has significantly improved. Methods range from processing images at varying scales [He et al., 2015b] over reduced downsampling in the encoder [Chen et al., 2017] and passing information through skip connections [Ronneberger et al., 2015] to processing features at different scales [Lin et al., 2017a; Zhao et al., 2017] or making convolutions deformable to attend to the correct areas of the image [Dai et al., 2017; Zhu et al., 2019]. Other problems such as avoiding duplicate detections and segmenting fine details led to the development of sophisticated post-processing methods [Bodla et al., 2017; Chen et al., 2017].

Especially in object detection, improvements usually do not come in the form of completely new models, but as interchangeable parts of existing models [Bochkovskiy et al., 2020]. This has led to the development of powerful toolboxes that allow quick design and modification of existing models [Wu et al., 2019b; Chen et al., 2019a]. Due to the many small improvements, the Faster R-CNN implementations in these modern frameworks score almost twice as high as the original implementation [Ren et al., 2015].

3.2 Learning from Few Examples: Few-Shot Learning

CNN models for the object recognition tasks discussed in the previous section have become very good over the last years. However, they do not have the flexibility and fast learning ability humans have, but require large amounts of labelled data [Halevy et al., 2009; Sun et al., 2017]. And once trained, new concepts cannot easily be added to a model because of so-called "catastrophic forgetting" of the concepts a model already knows [McCloskey and Cohen, 1989; McClelland et al., 1995; Kirkpatrick et al., 2017]. A wide variety of approaches exists to overcome these limitations (Table 2), and the one-shot object detection benchmarks developed in this dissertation are one addition to this set of approaches. It is most closely related to few-shot learning, the task of learning new concepts from just a few examples.

3.2.1 Few-Shot Learning

Few-shot learning describes tasks that use just a few samples, with a few usually referring to anything between one (one-shot) and twenty (20-shot) samples. The most deeply studied area is few-shot object recognition [Fei-Fei et al., 2006; Lake et al., 2015; Wang et al., 2020c]. It is usually realized as the task of learning a classifier for a set of never before encountered categories from only a few examples per category [Lake et al., 2015; Vinyals et al., 2016]. This few-shot learning task was popularized in 2015 with the Omniglot benchmark [Lake et al., 2015]. The Omniglot dataset consists of 1623 characters from 50 alphabets. The characters from 30 of these alphabets are used for pre-training. At test time, a series of episodes is created from the characters of the other 20 alphabets. Each episode has a small training set, for example one image for each of five characters. Then the pre-trained model is trained again on this training set and evaluated on a number of test samples of the five characters. The number of categories is called the ways and the number of samples per category the shots, so the example from above would be a five-way, one-shot task. The ways and shots are kept the same in all episodes. This evaluation scheme is supposed to lead the focus away from learning to classify a specific set of novel categories towards learning a model that can learn a classifier for any set of novel categories.

The main few-shot learning benchmarks are Omniglot [Lake et al., 2015] as described above and *mini*ImageNet [Vinyals et al., 2016] which uses a subset of ImageNet and a smaller image size for faster development than would have been possible on the whole ImageNet dataset. A number of similar benchmarks exist, all using the same episodic evaluation scheme and small images [Ren et al., 2018; Welinder et al., 2010; Bertinetto et al., 2019; Oreshkin et al., 2018]. Other tasks exist, but are far less popular. Hariharan and Girshick [2017] use full-scale ImageNet images and evaluate performance across 500 known and novel categories. Wertheimer and Hariharan [2019] use the iNaturalist dataset [Van Horn et al., 2018] to create a fine-grained recognition benchmark in which the number of shots varies between classes. Triantafillou et al. [2020] create a new Meta-Dataset composed of a variety of existing datasets covering different concepts (e.g., common objects, street signs or mushroom species) and image types (e.g., natural images or sketches).

Method	Objective	Data	Samples per category		Example	Exemplary papers
	Learn ...		labelled	un-labelled		
Unsupervised learning	a good representation	Unlabelled samples	–	Many	Relate two versions of the same image	[Oord et al., 2018] [Chen et al., 2020a] [Devlin et al., 2019]
Transfer learning	a new task	A small dataset	10-100	–	Classify flowers using a model trained on ImageNet	[Donahue et al., 2014] [Kolesnikov et al., 2019] [Devlin et al., 2019]
Semi-supervised learning	with less data	A small dataset & unlabelled data	10-100	Many	Train ImageNet with 1% of the data.	[Tarvainen and Valpola, 2017] [Chen et al., 2020b] [Xie et al., 2020]
Weakly-supervised learning	with less expensive data	A small dataset & data labelled for a different task	10-100 (labelled for the task)	Many (labelled diffe-rently)	Learn an object detector from mostly classification data	[Oquab et al., 2015] [Dai et al., 2015] [Hu et al., 2018b] [Mahajan et al., 2018]
Few-shot learning	from few examples	Very few samples per category	1-20	–	Learn a classifier from a few samples per category	[Lake et al., 2015] [Snell et al., 2017] [Finn et al., 2017] [Liu et al., 2019]
One-shot learning	from a single example	One sample per category	1	–	Learn a classifier from a single sample per category	[Lake et al., 2015] [Snell et al., 2017] [Finn et al., 2017] [Liu et al., 2019]
Zero-shot learning	from auxiliary information	Data & auxiliary information	Only auxiliary information	–	Learn to recognize novel categories from attributes	[Romera-Paredes and Torr, 2015] [Xian et al., 2018] [Radford et al., 2021]
Long-tail recognition	from unbalanced datasets	A very unbalaced dataset	1-1000s	–	A dataset with 1000s of images of cars but only few images of zebras	[Zhang et al., 2021a] [Cui et al., 2019] [Lin et al., 2017b] [Gupta et al., 2019]
Online learning	from incoming samples	A stream of data	A few at a time	–	Continually improve a spam filter	[Hazan, 2017] [Perozzi et al., 2014] [Mairal et al., 2010]
Continual learning	continually new concepts	A stream of data with new concepts	1-100s over time	–	Learn to classify new concepts as they are encountered	[Parisi et al., 2019] [Chen and Liu, 2018] [Kirkpatrick et al., 2017] [Rebuffi et al., 2017]
Active learning	to select the right samples	A small dataset & unlabelled data	1-100	Many	Select the right samples to label for a dataset	[Settles, 2009] [Ren et al., 2021]

Table 2: Overview of methods for learning with limited data.

Most methods use a DNN as image encoder and one of three major approaches to adapt it for each episode. In **metric learning**, no explicit adaptation happens, but the model applies a metric it learned during pre-training to classify the novel categories [Chopra et al., 2005; Koch et al., 2015; Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018; Chen et al., 2019b]. Therefore, the samples of the new categories and the test samples are usually encoded with the same network (often referred to as Siamese encoding [Chopra et al., 2005; Koch et al., 2015]) and the predicted category is chosen as that of the closest sample [Koch et al., 2015] or class prototype [Snell et al., 2017] as measured by the metric. In **meta-learning**, the objective is to learn how to learn from few examples [Finn et al., 2017; Santoro et al., 2016; Ravi and Larochelle, 2017; Nichol et al., 2018]. Instead of fine-tuning a trained model, the whole training and fine-tuning process is optimized jointly to get a model that can quickly adapt, and this model is then trained on the new samples in each episode. In **transfer learning**, a model trained on the training categories is fine-tuned on the new training data of each test episode [Dhillon et al., 2019; Kolesnikov et al., 2019]. The difference to meta learning is that the optimization used for fine-tuning is usually much simpler. While transfer learning was usually done with hundreds of examples, powerful models and pre-training strategies allow the same approach to succeed if only a handful of samples are available.

3.2.2 State-of-the-Art

At the time I started with the research for this dissertation, only a handful of methods for few-shot learning existed [Lake et al., 2015; Koch et al., 2015; Santoro et al., 2016; Vinyals et al., 2016]. The repertoire quickly expanded, first establishing the main principled methods, such as learning prototypes [Snell et al., 2017] and meta-learning [Finn et al., 2017]. Later methods became increasingly complex, but most of the progress was related to better training methods [Chen et al., 2019b; Dhillon et al., 2019], rather than deeper insights or actual learning to learn [Raghu et al., 2020]. With the same training strategies, simple baselines perform as good or better than previous state-of-the-art models [Chen et al., 2019b; Dhillon et al., 2019; Chen et al., 2021b; Tian et al., 2020; Wang et al., 2019c]. While the field has not moved too far in the initially studied inductive setting, where images are classified one by one, a lot of development happened in a new transductive setting [Liu et al., 2019], where a batch of images is evaluated at once. In this setting, the internal data structure of the batch can be used to form clusters and better approximate the data manifold [Kim et al., 2019; Hu et al., 2021; Requeima et al., 2019; Bateni et al., 2020].

3.2.3 What is Missing?

In 2017, when I started the research for this dissertation, object recognition models had become quite good and were able to identify and localize objects even in challenging scenes, but they still require thousands of samples per category for training. At the same time, research in few-shot learning had demonstrated some potential, but challenging real-world benchmarks were missing. The one-shot object detection benchmarks in this dissertation were developed to bridge this gap. They require models to solve difficult recognition problems, while at the same time being flexible enough to handle new categories.

3.3 A New Task: Visual Search

There are many possible tasks conceivable for one-shot object detection. The benchmarks in this part of the dissertation all follow a common visual search task: Given an image of an object (the reference), localize all objects of the same type in a separate image showing

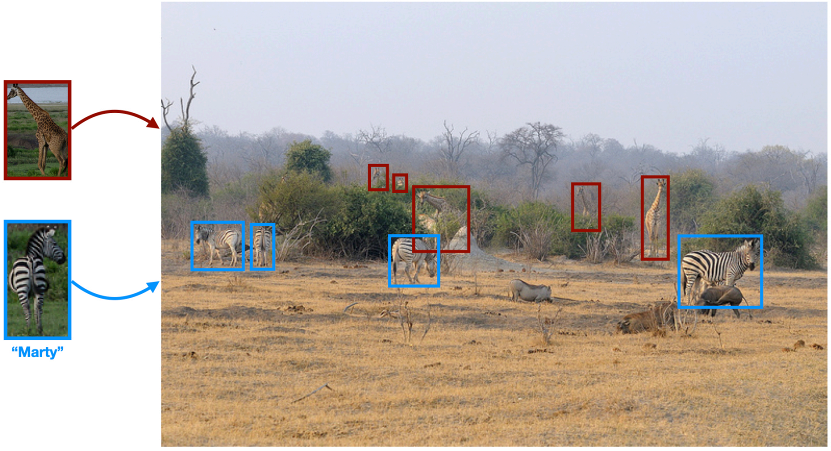


Figure 2: One-shot object detection as visual search: Given the image of a zebra, find all zebras in a scene. Given an image of a giraffe, find all giraffes.

a larger scene (the scene). There are many ways to realize such a search task and the benchmarks in this dissertation focus on three characteristics: First, the objective is not to find the identical object but rather objects of the same category. So for example, given an image of a zebra called Marty, the goal is not to detect just Marty but all zebras. Second, the scenes should be complex to model the human visual experience of seeing large scenes in which it is necessary to focus on the relevant details (something which humans do very successfully on that note). Third, the answer should be the precise location or a segmentation of each object.

This specific task design has a few goals. First of all, it should be suitable to measure detection of novel objects. Because the task is defined through a reference image, this can easily be achieved by showing reference images of novel objects at test time. Then, the task should be close to our human experience. The setup with a scene image which can contain objects known from training as well as other unknown objects and background fulfills this requirement much better than previous few-shot learning tasks. In fact, the task was directly inspired by the popular children book Where is Waldo, a hidden object game where the goal is to identify the fictitious character Waldo. Finally, the task should be challenging, so it can be used to explore the boundaries of what current models can achieve and where they fail. While few-shot learning and object detection are difficult on their own, their combination makes the task challenging. Inferring which category to detect from the reference requires strong inductive biases, especially on what constitutes an object and a category. This makes the task a good opportunity to investigate from a different angle what current object recognition models do and do not learn.

A number of different benchmarks building upon this task are introduced in the publications in this part of the dissertation. These vary in the kind of objects (characters and natural objects) and their specific goals (focusing more on detection or segmentation) but all share the same basic structure and objective.

3.3.1 Relationship to Object Recognition and Few-Shot Learning Tasks

The visual search task is closely related to the existing object detection and segmentation tasks. The main difference is that instead of having a fixed set of pre-defined categories, a reference image is shown to determine the category of interest. Therefore, the datasets and metrics can mostly be kept, and only the evaluation procedure has to be changed. The benefit of the reference based evaluation is that where typical object detection systems have to be trained with thousands of examples per category, it allows the trained models to be applied to novel categories simply by showing a corresponding reference image.

Compared to the typical task design in few-shot learning, the visual search task has a number of differences. While the evaluation is still episodic, each episode now is the search of one type of object within a scene. This does not only require methods to identify objects within a scene, but also means that at test time, objects of known and novel categories co-exist in a scene. This makes the problem significantly harder because a model may be biased towards the known categories from training. Additionally, the goal is to use full-size scene images, rather than downsized samples. This does not only make the task harder, but results are also much more directly applicable to real-world computer vision problems.

3.3.2 Relationship to Other Tasks

The visual search task is not only related to object detection and few-shot learning, which it directly adds onto, but also to a number of other tasks.

The first related task is retrieval, the task of retrieving matching images from an image collection [Niblack et al., 1993; Smeulders et al., 2000; Sivic and Zisserman, 2003; Sharif Razavian et al., 2014]. It has well known applications in reverse image search [Google Inc., 2020; Jing and Baluja, 2008; Hu et al., 2018a] or fashion and shopping applications [Jing et al., 2015; Zhang et al., 2018; Yang et al., 2017]. Like the task here, the objective is searching images using visual examples. However, there are two major differences. The first difference is that the majority of existing benchmarks addresses particular object retrieval, identifying the same object in different images. For example, finding images of a specific building or product [Philbin et al., 2007, 2008; Oh Song et al., 2016]. In contrast, the search task defined here addresses categorical retrieval, identifying objects of the same category. For example, finding all chairs given an image of a chair. The second difference is that the focus of the visual search task is on localization of multiple objects within complex scenes, while most retrieval benchmarks evaluate the ranking of the top retrieved examples. This additionally requires models to localize objects within scenes, and forces them to make a decision for each object if it is from the reference category instead of only predicting a relative ranking between objects. These two differences make the task quite different from most current retrieval benchmarks, despite having a very similar core structure.

The next two closely related tasks are co-localization [Sivic et al., 2005; Grauman and Darrell, 2006] and co-segmentation [Rother et al., 2006], localizing or segmenting common objects in two or more images. These tasks were designed with the goal to learn object detection and segmentation with little to no supervision. While this is very similar in spirit to the goal of the task discussed here, the approach is quite different. In the search task, the object of interest is explicitly specified in the reference image, while the co-localization tasks require models to identify the common object. Having to discover the common object requires the data to be selected such that it has well-defined correspondences. This does not scale to large images and many categories, and as a result co-localization and segmentation models can only be applied to object centric images [Fergus et al., 2003; Shotton et al., 2006; Batra et al., 2010; Rubinstein et al., 2013] or pre-select image pairs [Li et al., 2018; Hsu et al., 2019]. In contrast, the visual search task is designed to learn generalization to new concepts after training. It requires annotations during training, but the trained model can later be applied to detect objects of novel categories in any image collection.

The last related task is tracking of objects in videos [Yilmaz et al., 2006; Kalal et al., 2011; Wu et al., 2013; Bernardin and Stiefelhagen, 2008; Milan et al., 2016]. Like in retrieval, this is a correspondence problem between two views of the identical object, not a categorical relationship task. But the major difference is that the videos contain temporal information with strong inductive biases. As a result, even the quite related task of one-shot video object segmentation [Perazzi et al., 2016; Caelles et al., 2017, 2019] is in fact quite different from the problems discussed here.

3.3.3 Parallel Work

In parallel to our work, a number of other one-shot and few-shot object detection tasks emerged. They fall roughly into three categories: 1. Episodic tasks which are evaluated on n-way, k-shot episodes as done in traditional few-shot learning [Dong et al., 2018; Chen et al., 2018; Schwartz et al., 2019; Wu et al., 2019a]. 2. Transfer tasks that require adapting a detector to a new dataset, with very few examples [Dong et al., 2018; Chen et al., 2018]. 3. Continual learning tasks where an existing detector is extended to cover additional categories [Kang et al., 2018]. Each of these tasks has a different focus and different strengths and weaknesses. The visual search task is the most flexible because the model can directly be applied to novel scenes and objects by providing respective scene and reference inputs. But it has a number of drawbacks, discussed in the next section.

Researchers are used to having a single task to focus on [Dehghani et al., 2021], but the variety of problems studied in few-shot object detection is an opportunity, because as the following sections will show, a single task and benchmark can never sufficiently approximate all aspects of a capability. However, while it would have been very interesting to study the differences between these tasks, I here focus only on the visual search task. This focus on a single task design allows for deeper discovery into the effect data has on learning and identifying core aspects of the capabilities of object recognition and few-shot learning.

3.3.4 Advantages and Limitations of the Task

Realizing one-shot object detection as visual search has a few advantages. Models can be evaluated on any new category at test time simply by providing an appropriate reference image. This overcomes a key limitation of almost all standard object detection models, that are trained to recognize a fixed set of categories. Visual search is also a typical problem that humans experience, for example when searching for lost keys or a new gadget. This makes it intuitively understandable, which in turn can help to analyze errors models make. Finally, it is more challenging than existing object detection and few-shot learning tasks. Models trained and tested on it must have a good general representation of objects and scenes, two aspects of human vision that current models struggle with.

But the task design also has limitations. The first is that the task is not well-defined. While the difference between a coffee maker and a zebra is clear, the difference between a zebra and a horse is less clear. If the task is to separate kitchen appliances from animals, the horse, and the zebra belong to the same category. However, if the task was more fine-grained, the horse and zebra would likely belong to different categories. The issue with the task setup is that a model can learn this only implicitly from the categories used during training. But for many objects, especially if categories become more fine-grained, the difference is not obvious. To fix this issue, the category would have to be specified explicitly, for example by providing the word horse or animal with the image of a horse. However, while perfect performance is likely not possible with this task setup, humans have a strong intuition even in such situations and the task is designed to measure this capability rather than evaluating the ability to reproduce some objective truth.

The second limitation is that models trained for the visual search task alone lack a few components to make them useful for most applications. They would likely require a way to classify known objects without relying on an explicit reference and a continual learning component that enables adding novel objects to the known objects over time. Both of these can be added relatively easily, but are left out here to keep the task as simple as possible and allow focussing on understanding how models recognize objects and few-shot learn.

3.4 One-shot Segmentation in Clutter

Claudio Michaelis, Matthias Bethge, Alexander S. Ecker; *ICML 2018*

The project was initiated by A.E. and M.B. and led by C.M. and A.E.. All experiments were designed and conducted by C.M. with input from A.E.. C.M. and A.E. wrote the first draft and designed the figures with input from M.B.. All authors contributed to the final version and provided critical revisions from different perspectives.

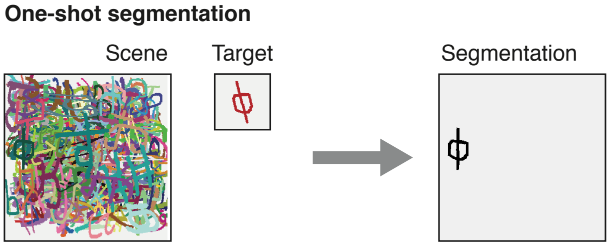


Figure 3: One-Shot segmentation task [Michaelis et al., 2018a]

3.4.1 Motivation

The goal of this study was to create a few-shot learning benchmark that requires to learn an understanding of objects. The existing benchmarks Omniglot [Lake et al., 2015] and *mini*ImageNet [Vinyals et al., 2016] present objects in separate images and require a model to sort them into one of five or twenty categories. This is very different from human experience where objects are usually embedded into the surrounding and to identify some specific object, it may be necessary to check for a number of objects if they are what is sought. This is a significantly harder challenge because it requires separating the objects in a scene from each other and making a separate choice for each of them. Therefore, the goal was to create a benchmark where scene complexity can be precisely controlled by the amount of background clutter. And to develop models, with which we could investigate how scene complexity impacts one-shot segmentation performance.

3.4.2 Benchmark

In this study we defined the first instance of the visual search task (Section 3.3), one-shot segmentation on *cluttered Omniglot*: Segment a target character, specified by a reference image, from a larger scene (Figure 3). We created *cluttered Omniglot* by placing between 4 and 256 strongly augmented and randomly colored characters from Omniglot into a scene (Figure 4). The target character is specified through a separate reference image, and the segmentation of the scene is provided as the label. The metric is the segmentation quality of the character in the scene. We used 30 of the 50 alphabets in Omniglot to create the training dataset and the other 20 to create the test set.

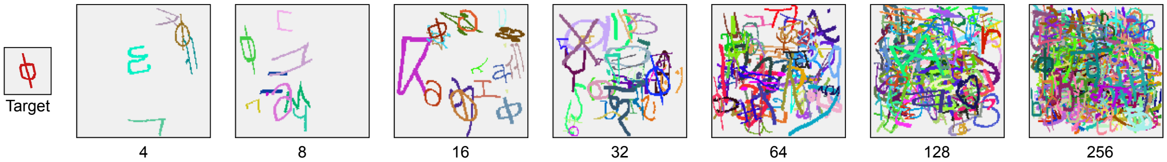


Figure 4: *clutteredOmniglot* [Michaelis et al., 2018a]

3.4.3 Results

To tackle the problem, we built a baseline model by combining a popular segmentation model (U-Net [Ronneberger et al., 2015]) with a popular few-shot learning method (Siamese Networks [Koch et al., 2015]). This *Siamese-U-Net* generalizes very well to novel categories and performs almost the same for novel characters as it is for characters known from training. However, while it performs almost perfectly in simple scenes, performance quickly drops off when clutter increases.

What causes this performance drop? There are two possible explanations: The model can fail to properly segment the characters, or it can fail to recognize the right character among the large number of other characters. To distinguish between these hypotheses, we simplified the task by extracting the individual characters from the scene, turning it into a set of between 4 and 256 individual images each showing a single character. Without background clutter, a standard Siamese network trained to identify the target character among them performed almost perfectly, even in the most difficult case with 256 characters. However, when background clutter is kept, the model's performance dropped as clutter increased. This shows that identifying the correct character among many characters is much easier than separating the characters from the background.

Building upon this insight, we created a background masking model *MaskNet* which first extracted a set of candidate characters and then decided which is the correct one. This "Segment first, Decide later" approach indeed boosts performance in cluttered scenes, but cannot close the gap to the pre-segmented model.

3.4.4 Discussion

There are two things that are surprising in our results: The first is that all models performed almost as well on novel characters as on those they knew from training. This is good news, because it indicates that they learned a solution which works for all objects, not just those in the training set. However, this may be not surprising since Omniglot characters are relatively simple and the standard Omniglot few-shot learning benchmark is mostly solved.

The second surprising finding is that the presence of background clutter poses such a big challenge. This is especially surprising because characters can easily be told apart by color (each character has a unique and uniform color) and therefore indicates that there are significant issues with the general assumption that CNNs decompose scenes into objects [LeCun et al., 2015; Kriegeskorte, 2015]. The cause of this is hard to determine, but one reason could be the purely feed-forward structure of CNNs. It is known that the human brain uses a lot of attention and recurrence in its processing, and using similar mechanisms may improve model performance. Evidence for this hypothesis is provided by our *MaskNet* model, which mimics a form of object based attention [Treisman and Gelade, 1980] and performs better when facing clutter. Similar results have been found by Mnih et al. [2014] and Spoerer et al. [2017], who show that recurrent models perform better than traditional feed-forward models on another clutter task. However, a large gap remains between the ability of *MaskNet* and the pre-segmented model at localizing the target character.

3.5 One-shot Instance Segmentation

Claudio Michaelis, Ivan Ustyuzhaninov, Matthias Bethge, Alexander S. Ecker; *ArXiv 2018*

The project was initiated by C.M. and A.E. and led by C.M.. A.E., C.M. and M.B. defined the task and shaped the direction of the study. The Faster R-CNN model was conceptualized by C.M. and A.E. and implemented by C.M.. The experiments were designed and conducted by C.M. and I.U. with input from A.E. and M.B.. C.M., A.E. and I.U. wrote the first draft and designed the figures with input from M.B.. All authors contributed to the final version and provided critical revisions from different perspectives.

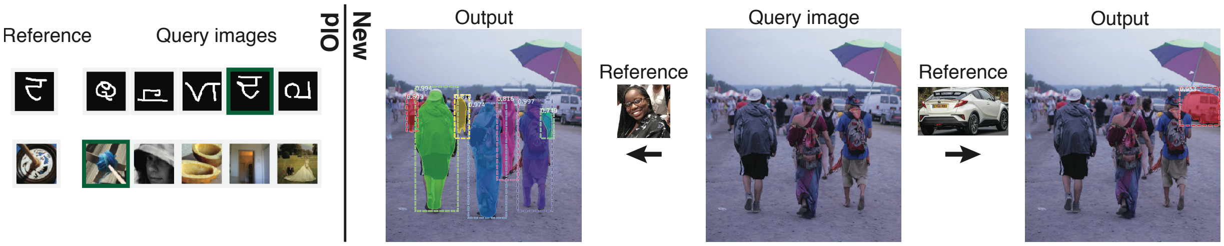


Figure 5: Comparison between classical one-shot learning tasks and the new one-shot instance segmentation benchmark on COCO. Predictions from our model.[Michaelis et al., 2018b]

3.5.1 Motivation

Key to the success of CNNs is their ability to handle complex natural images. Therefore, after studying a highly controlled setup in *cluttered Omniglot* this study focuses on real-world scenes. At the time, a similar type of task had been studied on Pascal VOC [Shaban et al., 2017] and on ImageNet Detection [Schwartz et al., 2019]. But both of these datasets contain on average only objects of a single category (see Table 1 in Section 3.1), and therefore models can perform reasonably simply by selecting the foreground object(s) [Michaelis et al., 2020]. After seeing CNNs struggle to identify objects in *cluttered Omniglot*, the goal was to create a real-world benchmark with multiple objects per scene that is challenging enough that models can not rely on a simple trick.

3.5.2 Benchmarks

We defined two new benchmarks, one-shot object detection and one-shot instance segmentation (Figure 5). As dataset, we selected COCO [Lin et al., 2014] because the complexity of the scenes means there is no single foreground object. The objective was the same as for the standard object detection and instance segmentation tasks, but instead of classifying each object into one of 80 categories, they had to be classified whether they were from the same category as the object in the reference image. That reference image was a random instance from another image, which was cropped tightly out of the respective scene. To test one-shot generalization, the dataset was split into four subsets of 20 categories and for each of these subsets a model was trained without using annotations for the categories in the respective subset. As the final metric, we used the standard AP^{50} metric common in object detection by assigning the detections for each reference to the corresponding category.

3.5.3 Results

To evaluate the difficulty of the task, we combined the state-of-the-art instance segmentation model (Mask R-CNN[He et al., 2017]) with the same few-shot learning model used before (Siamese Networks [Koch et al., 2015]). We find that performance on the object detection and instance segmentation benchmarks is very similar, with instance segmentation being slightly harder. For both tasks, performance on novel categories was significantly lower than for the training categories. A qualitative analysis revealed that the Siamese Mask R-CNN model generates good segmentation masks and bounding box estimates, but predicts a lot of false positives. Finally, an analysis of the performance on images with different numbers of objects revealed that performance dropped in scenes with more objects.

3.5.4 Discussion

This study had two clear insights: 1. One-shot generalization is challenging in natural scenes. 2. Classification rather than segmentation is the main issue. These results are almost exactly opposite to what we found on *cluttered Omniglot*. How can this be?

To understand why generalization is so much harder, two factors appear logical. First the objects are much more complex with two instances of a chair or a person being visually much further apart than two characters despite strong data augmentation in *cluttered Omniglot*. Second, the number of categories in COCO (80) is significantly smaller than in Omniglot (1623). In combination, there is less diversity in the categories that are used to learn a more challenging problem, making memorizing the categories a much more attractive solution. This view is consistent when comparing our results with those of Schwartz et al. [2019] who observe a much smaller performance gap between novel and known categories in their ImageNet task. Like *cluttered Omniglot* they use more categories (100) from a much smaller set of concepts (animals only) while additionally using much simpler images which usually contain only a single foreground object.

The second question, why segmentations looks surprisingly good, is harder to answer. For one, while natural images are highly complex, they have a less cluttered appearance than *cluttered Omniglot* because objects have relatively convex shapes and do not overlap as much. Additionally, by using the state-of-the-art instance segmentation model, we were able to harness all the domain knowledge which is integrated into these models. This could also explain the surprising quality not only of the bounding boxes but also of the segmentation masks: Mask R-CNN segments the predicted bounding boxes, making the problem significantly easier than segmenting the whole scene.

3.6 A Broad Dataset is All You Need for One-Shot Object Detection

Claudio Michaelis, Matthias Bethge, Alexander S. Ecker; *ArXiv 2020*

The project was initiated by C.M. and A.E. and led by C.M.. A.E., C.M. and M.B. shaped the direction of the study. The experiments were designed by C.M. and A.E. and conducted by C.M.. C.M. wrote the first draft and created the figures with input from A.E.. All authors contributed to the final version and provided critical revisions from different perspectives.

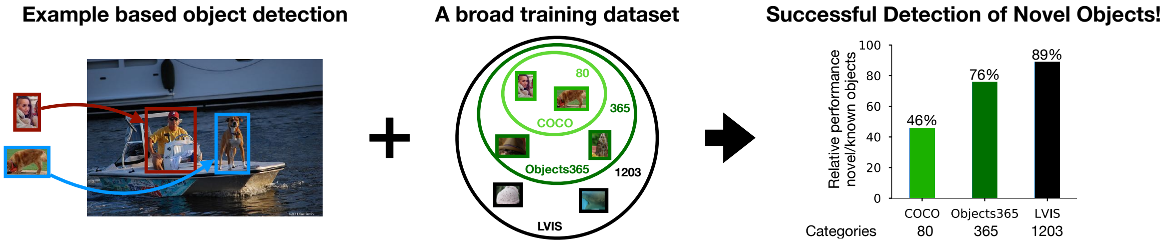


Figure 6: Relative performance in one-shot object detection increases with the number of training categories [Michaelis et al., 2020]

3.6.1 Motivation

In the two previous studies, we found seemingly contradicting results. On *cluttered Omniglot* [Michaelis et al., 2018a] models perform almost as well on novel characters as they do on known characters, while on COCO [Michaelis et al., 2018b] there is a large generalization gap. The goal of this study was to better understand this difference. As hypothesized in the previous section, the ratio of object complexity to the breadth of categories could be the reason for this effect. Therefore, in this study, we increased the number of categories in the one-shot object detection task by introducing new benchmarks that use broader datasets.

3.6.2 Benchmarks

We defined two new benchmarks using the same task and metric as in one-shot object detection [Michaelis et al., 2018b] but on datasets with a broader set of categories: Objects365 [Shao et al., 2019] and LVIS [Gupta et al., 2019] with 365 and 1203 categories respectively. As for COCO, we split the categories for these datasets into four equal subsets and follow the same training concept to be able to evaluate the detection of novel categories. For LVIS we defined four additional subsets, leaving out all categories that correspond to each of the COCO subsets, to be able to use LVIS as a training dataset for the COCO task.

3.6.3 Results

We found that the number of categories plays a crucial role in generalization. On COCO, performance on novel categories was on 45% of that on known categories. This relative performance improved to 76% on Objects365 and to 89% on LVIS. In other words, the model was able to detect novel categories almost as well as known categories on LVIS. While the exact numbers vary slightly, this effect was consistent across a range of models and model variations. A systematic study training with only a fraction of categories or instances verified that the number of categories was the key factor and more important than the raw number of samples.

This closing of the generalization gap also led to qualitative changes. On COCO, more powerful models or longer training times primarily improved performance for known categories and almost not at all for novel categories. On LVIS and Objects 365 the same changes increased performance on novel categories as well. The smaller the generalization gap was, the more the improvements also transferred. Training with the additional categories in LVIS also improved performance on COCO and allowed us to achieve a new state-of-the-art performance for one-shot object detection on COCO.

3.6.4 Discussion

The key insight of our study is that the breadth of the training dataset is the main factor that determines how well a model generalizes to novel categories. Therefore, in settings in which broad datasets with many categories are available, few-shot learning may have a surprisingly simple solution. In contrast, algorithmic methods may play less of a role, as a comparison between our simple baseline trained on a larger dataset and more complex models [Hsieh et al., 2019; Chen et al., 2021a] shows. And our results indicate that once the generalization gap is closed, most improvements in detection performance directly transfer to novel categories, simplifying the problem from detection and generalization to mostly detection.

This finding that one-shot generalization depends to a large extent on the breadth of the dataset used for training is supported by other studies that found similar effects in few-shot learning [Sbai et al., 2020; Jiang et al., 2020] and one-shot semantic segmentation [Luddecke and Ecker, 2021]. The same principle is behind breakthrough results on one-shot ImageNet [Kolesnikov et al., 2019], zero-shot ImageNet [Radford et al., 2021; Jia et al., 2021] and few-shot generalization in natural language processing tasks [Brown et al., 2020], all of which results were achieved by a combination of training on larger datasets and scaling up model size. Kolesnikov et al. [2019] even confirms our finding that only once the dataset reached a sufficient size, additional model capacity benefits generalization.

Thus, if possible, pre-training on broad datasets seems to be sufficient for successful few-shot learning. But a few questions remain. How far does the generalization gap close with even more categories? Does it eventually vanish completely? And is simply the number of categories relevant, or are specific categories more helpful? Studies by Sbai et al. [2020] and Jiang et al. [2020] indicate that semantically related categories are more helpful than unrelated ones. And Meta-Dataset [Triantafillou et al., 2020] showed that domain shifts, such as going from natural images to sketches, are relevant even for models trained on very broad datasets [Dumoulin et al., 2021]. Finally, what influence does the granularity of categories have? Are fine-grained tasks easier or harder [Wertheimer and Hariharan, 2019; Schwartz et al., 2019], and what happens when sub- or super-categories are included?

Going forward, the field of few-shot learning will likely require a variety of harder, more realistic problems. Some of them may require new methods, but for most of them the solution may be much simpler than expected a few years ago. The thought is quite exciting to have pre-trained models that are excellent few-shot learners to build upon. In natural language processing, this paradigm is already well established with GPT-3 [Brown et al., 2020]. Of course, one may question if using a model pre-trained on a large dataset can still be considered few-shot learning. But compared to humans, who can rely on broad world knowledge when few-shot learning, this is probably fair. Morgan [1989] estimate that children hear 4.2 million sentences until the age of five. Maybe, but here I wander into the realm of speculation, generalization is not that difficult to achieve after all, but comes naturally when trying to make sense of a complex world.

4 Part II: Connecting Surprising Model Failures

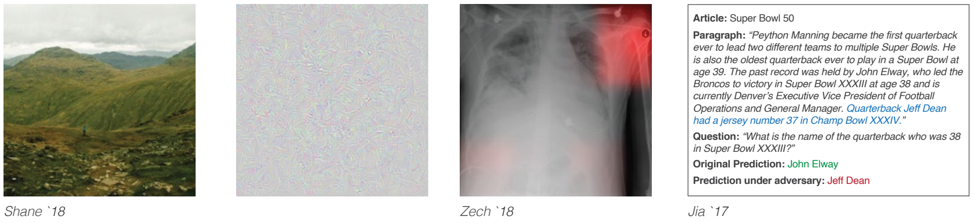
This part of the dissertation broadens the view to investigate whether surprising model failures in different areas of machine learning can be explained by shortcomings in the respective benchmarks. The next section provides an overview of a number of model failures that have been puzzling researchers (Section 4.1). Then I summarize a publication (Section 4.2) that identifies shortcut learning, models exploiting a mismatch between the precise benchmark objective and the desired broad capability, as the underlying cause behind many of these failures. The full length paper can be found at the end of the dissertation.

4.1 Generalization Failures

In most benchmarks, train and test data are created from the same distribution by randomly splitting the samples. This setting is usually called i.i.d., because in the samples in the train and test set are independent and identically distributed. Most benchmarks only evaluate i.i.d. performance, and most of the recent progress in model performance is assessed with this kind of measure.

But when tested outside their training distribution, these models often fail. Computer vision models are derailed by small amounts of noise or distortions [Geirhos et al., 2018; Hendrycks and Dietterich, 2019; Engstrom et al., 2019; Michaelis et al., 2019][1] or rely only on image background [Beery et al., 2018; Shane, 2018; Xiao et al., 2021] (1st column in Figure 7). Question answering methods only use the last sentence of a question for the answer, even if the information in that sentence is irrelevant [Lapuschkin et al., 2019] (4th column in Figure 7). Reinforcement learning algorithms learn to cheat [Lehman et al.,

[1] I co-authored [Michaelis et al., 2019], but did not formally include it in this dissertation.



Article: Super Bowl 50

Paragraph: *"Peython Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had a jersey number 37 in Champ Bowl XXXIV."*

Question: *"What is the name of the quarterback who was 38 in Super Bowl XXXIII?"*

Original Prediction: John Elway

Prediction under adversary: Jeff Dean

Task for DNN	Caption image	Recognise object	Recognise pneumonia	Answer question
Problem	Describes green hillside as grazing sheep	Hallucinates teapot if certain patterns are present	Fails on scans from new hospitals	Changes answer if irrelevant information is added
Shortcut	Uses background to recognise primary object	Uses features irrecognisable to humans	Looks at hospital token, not lung	Only looks at last sentence and ignores context

Figure 7: Examples of DNN generalization failures [Geirhos et al., 2020]

2018], for example, by pausing a game to avoid loosing [Murphy Vii, 2013]. These failures are not just a research question, but can cause harm in applications. Autonomous driving algorithms struggle in bad weather conditions [Zhang et al., 2021b; Yurtsever et al., 2020; Sakaridis et al., 2018; Michaelis et al., 2019; Pitropov et al., 2021] or unexpected scenarios [Tesla Inc., 2016]. Translation and text generation algorithms reproduce harmful biases [Prates et al., 2020; Brown et al., 2020; Mehrabi et al., 2021; Barocas et al., 2019]. Tools to screen job applicants reproduce historic inequalities, such as being biased against women [Dastin, 2018]. And models for medical applications are prone to overfitting because datasets are small and various strong confounders exist [Zech et al., 2018; Roberts et al., 2021; Winkler et al., 2019; Narla et al., 2018] (3rd column in Figure 7).

To identify these kinds of failures, models have to be tested outside their training distribution. But this kind of out-of-distribution (short o.o.d) performance is not commonly evaluated. For some time, the effect was rather the opposite. The diversity of ImageNet [Russakovsky et al., 2015] and the astonishing results CNNs achieved on it created the illusion that these models actually figured out object recognition. CNNs quickly mastered a number of tasks [Sharif Razavian et al., 2014; Donahue et al., 2014; Girshick et al., 2014; Long et al., 2015] and were even seen as a model of the brain [Yamins et al., 2014; Yamins and DiCarlo, 2016; Cadena et al., 2019]. But the examples above make clear, that CNNs are far from achieving the robustness of human vision.

The hypothesis behind this part of the dissertation is that failures in o.o.d. generalization are not random, but that there is often a common pattern behind different failures. To better understand this idea, let us look at two examples, texture bias and adversarial attacks.

4.1.1 Texture Bias

In Part I of the dissertation (Section 3), a contradictory picture about object recognition emerged. On the one hand, clutter dominates performance on *cluttered Omniglot* [Michaelis et al., 2018a]. On the other hand, the segmentation predictions for the natural images look reasonably good [Michaelis et al., 2018b]. This is not limited to one-shot object detection. Algorithms for image classification and object detection perform increasingly well, even surpassing human performance in some tasks [He et al., 2015a; Shankar et al., 2020], but fail when a bit of noise or distortions are added to the images [Geirhos et al., 2018; Hendrycks and Dietterich, 2019; Michaelis et al., 2019]. This points at a knowledge gap. Which interpretation is correct? Are CNNs good at recognizing objects and simply struggle with the delicate shapes of the Omniglot characters and extreme levels of clutter? Or is something going fundamentally wrong in the detection of objects that the strong inductive biases of Faster R-CNN can cover up in the natural image benchmarks?

The traditional view is that CNNs integrate edges and corners into parts and then combine these parts into representations of objects [LeCun et al., 2015; Kriegeskorte, 2015]. However, this does not match our findings. If a CNN can not even learn to separate a simple uniformly character from the background, how should it recognize natural objects that are substantially more complex and have complex textures? The explanation may lie in CNNs tendency to classify objects by texture instead of shape, as demonstrated by Geirhos et al. [2019][2].

Texture bias is very interesting because it provides a potential explanation for a range of issues. Searching a character in *cluttered Omniglot*, while interpreting the scene as a texture and not a set of objects, makes the problem extremely challenging. In comparison, the scenes in COCO are much less cluttered and objects have texture, thus making detection by texture a potential solution. This view is supported by results from Ustyuzhaninov et al. [2018] who investigate a one-shot segmentation task that is very similar to the one-shot object detection task discussed here but instead of an object a texture patch is given as reference. Even with purely synthetic training data, models can readily segment natural images with surprising precision. Texture bias could also explain, why CNNs can classify as well from texture features as they do from all features [Gatys et al., 2015], and why they can perfectly well classify texturized images [Brendel and Bethge, 2019]. And, why noise and distortions that change texture much more than shape are such a big issue [Geirhos et al., 2018; Hendrycks and Dietterich, 2019; Michaelis et al., 2019]. Of course, it is not the only explanation and recognition does not have to happen via texture. But it provides a blueprint, how a range of seemingly unrelated problems can be interpreted in the context of a single issue, providing a potentially simple explanation and opening up new ways to look at existing issues.

4.1.2 Adversarial Examples

One of the sharpest points of criticisms towards DNNs is the existence of adversarial examples [Szegedy et al., 2013; Biggio et al., 2013; Carlini and Wagner, 2017; Madry et al., 2018; Athalye et al., 2018; Brendel et al., 2018]. Adversarial examples are tiny image modifications which are so small that humans cannot perceive them, but change the prediction of a DNN (2nd column in Figure 7). So far, no method was found that can make models robust against this kind of modification, and some argue that the existence of adversarial examples is proof that something is fundamentally wrong with CNNs [Marcus, 2018]. While adversarials are still a problem, Ilyas et al. [2019] have shown that the tiny image modifications are not entirely random. By training a model on adversarially perturbed images and testing it on normal images they found, that the adversarial perturbations were actually predictive of the image category, despite being meaningless and mostly imperceptible to us.

[2] I co-authored this work, but did not formally include it in this dissertation.

This mechanism cannot explain all adversarial examples [Nakkiran, 2019], but it exposes how DNNs can latch onto any feature to make a prediction, even if it is unnoticeable and inexplicable.

4.1.3 What is Missing?

These two example cases demonstrate that the process by which DNNs make predictions is not nearly as well understood as one may think. And while some issues have been studied quite extensively, these efforts have usually focused only on one problem in isolation, such as noise robustness or domain transfer. What is missing is a mental model that connects all of these failures, a way to think about these failures that goes beyond a specific setting. The goal of this part of the dissertation is to provide such a mental model by analyzing the way benchmarks measure capabilities not just for a specific task or field, but across machine learning.

4.2 Shortcut Learning in Deep Neural Networks

Robert Geirhos*, Jörn-Henrik Jacobsen*, Claudio Michaelis*, Richard Zemel, Wieland Brendel, Matthias Bethge & Felix A. Wichmann; *Nature Machine Intelligence 2: 665–673 (2020)*

The project was initiated by R.G. and C.M. and led by R.G. with support from C.M. and J.J.; F.A.W. added the cognitive science and neuroscience connection; M.B. and W.B. reshaped the initial thrust of the perspective and together with R.Z. supervised the machine learning components. The toy experiment was conducted by J.J. with input from R.G. and C.M. Most figures were designed by R.G. and W.B. with input from all other authors. Figure 2 (left) was conceived by M.B. The first draft was written by R.G., J.J. and C.M. with input from F.A.W. All authors contributed to the final version and provided critical revisions from different perspectives.

Version notice: Due to the publisher's copyright assignment, reprinting the final formatted and published version is not possible; therefore, the preprint version (arXiv version v4) is included in this dissertation.

4.2.1 Motivation

The motivation for this study was a simple insight: Many problems in deep learning appear to follow the same pattern. While ImageNet models are considered learning to recognize objects, they in fact learn to recognize textures [Geirhos et al., 2019] (Figure 8). While question answering models are considered learning to answer based on a paragraph of text, they in fact base their answer only on the last sentence [Jia and Liang, 2017]. While medical AI models are considered learning to classify diseases, they in fact often use spurious signals, such as a hospital token in an x-ray image [Zech et al., 2018] (Figure 7). All of these issues are not apparent when these models are tested on i.i.d. test sets, but manifest themselves in surprising generalization errors. Shape bias makes computer vision models susceptible to noise, question answering systems can be fooled by adding irrelevant information, and medical AI systems fail in practice. This makes models highly unpredictable and hinders applications.

The goal of this study was to analyze this problem on a systemic level, looking beyond a specific problem such as texture bias and connecting the common patterns. By establishing this as a cross-disciplinary issue, we hoped to raise awareness across the machine learning community and initiate research into the underlying problem rather than patching each issue individually.

4.2.2 Analysis

The first part of our analysis was a hierarchy of solutions a model can learn. The first are non-solutions that fail even on the training data. The second are training solutions such as memorization that work on the training data but fail on i.i.d. test data. The third are shortcut solutions that work on i.i.d. test data but not on o.o.d. data that for a human represents the same task. The last are intended solutions that correspond to learning the capability that the model is intended to learn.

Many of the issues discussed on the last pages can be interpreted as different forms of shortcut learning, learning solutions that only work in the specific context but fail to generalize to other scenarios. The phenomenon behind shortcut learning is not a new problem, neither is it a problem just of machine learning. From students that root learn to ace simple tests without real understanding [Scouller, 1998; Chin and Brown, 2000] to animals which can fool researchers by using unintended cues [Geirhos et al., 2020], problems can often be solved in different ways than those who set them expect and learning agents will figure these out. But to make sure the recent progress in machine learning research is transferable to the real world, shortcut learning has to be taken into consideration.

Therefore, the second part of our analysis focused on understanding the origins of shortcut learning. The first element is the data, which often contains spurious correlations. These range from simple human interpretable relationships such as image background [Beery et al., 2018] to imperceptible patterns which carry no information whatsoever for humans [Ilyas et al., 2019]. The second element is the task design, which can favor simple yet non-intended decision rules. While forced choice image classification is easy to annotate, train on, and evaluate, it reduces a complex problem such as object recognition to a simple choice. If texture features are sufficient to solve this problem and easier to learn than object shape, it only makes sense for a model to focus on them.

The consequence of this are surprising generalization failures. Models fail in scenarios which humans expect to be equivalent to what the model encountered during training and testing. However, these scenarios are out of distribution with respect to some feature the models learned to use. It is anthropomorphism, our interpretation of solutions in human terms, that leads us on the wrong track. Conversely, models which appear to be simply bad at generalization are capable of surprising generalization behaviors that do not make sense to us [Jacobsen et al., 2018; Brendel and Bethge, 2019; Nguyen et al., 2015] (Figure 8, right).

4.2.3 Discussion

The key insight of this study is that the notions of overfitting and generalization have to be reconsidered. Traditionally, the term overfitting was mostly used to describe memorization of training examples that limits generalization to i.i.d. test sets. However, despite their

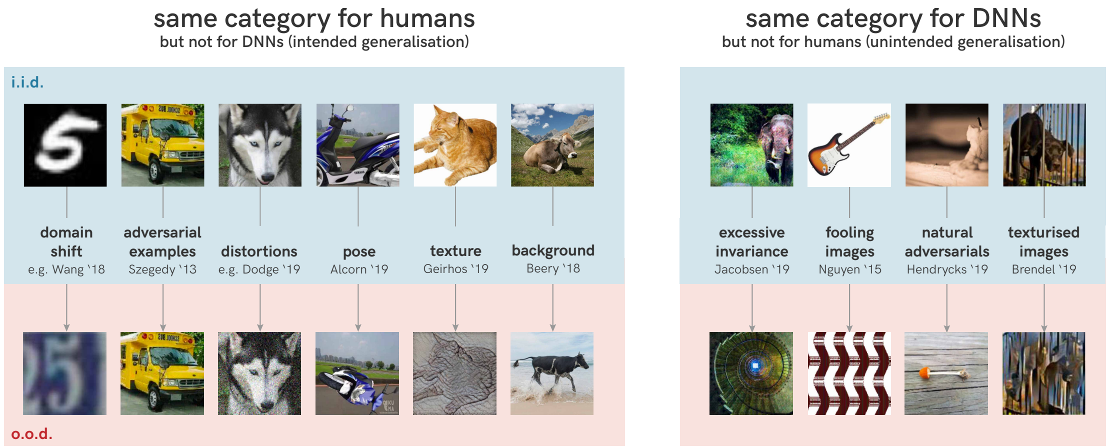


Figure 8: Examples of the different generalization directions of humans and machines. On the left are examples where humans generalize but machines fail. On the right are examples where machines generalize but humans fail. [Geirhos et al., 2020]

large parameter count, DNNs overfit little to the training samples and achieve very good i.i.d. generalization [Zhang et al., 2017]. Where DNNs overfit is on the specific tasks and benchmarks. Models which perform extremely well on ImageNet fail when noise is added to the images [Geirhos et al., 2018; Hendrycks and Dietterich, 2019]. This kind of o.o.d. tasks is where shortcut learning becomes visible.

Based on this analysis, there are three recommendations we make in the publication. 1. To avoid being surprised by generalization failures, it is important to interpret results carefully and distinguish between good performance on a specific benchmark and skill in the underlying capability. If in doubt, one should take a conservative stance and rather assume a model learned a shortcut than the intended solution. 2. To detect shortcuts early, it should become standard practice to test models on a range of o.o.d. tests. These tests will have to continuously develop alongside models to include newly identified issues and get rid of redundant tests which are highly correlated. 3. To better understand the origins of shortcut learning, it will be important to better understand the inductive biases of models. A key component to consider is the principle of least effort, which describes the tendency of learning systems to find the easiest solution.

As with most big systemic issues, many aspects of shortcut learning are long known [Ponce et al., 2006; Torralba and Efros, 2011]. For example, [Torralba and Efros, 2011] demonstrate that datasets have strong built-in biases by demonstrating that models can easily tell apart images from different datasets, and that models trained on one dataset often do not generalize to other datasets. The author's message about the limitations of individual benchmarks and the need to carefully evaluate the value and biases of each dataset ring as true today, as they did back when the study was published. But the progress DNNs enabled on challenging real-world tasks created the illusion of human-like performance [He et al., 2015a; Wang et al., 2021; He et al., 2020; Esteva et al., 2017; Silver et al., 2016; Vinyals et al., 2019; Berner et al., 2019] and led to overly optimistic predictions such as fully self-driving cars within years [Kaufman, 2014; LaFrance, 2015; Kalanick, 2015; Ross, 2016; Hassler, 2016]. As a result, the number of issues and challenges that appeared over time was surprising for many. After publication, the term shortcut learning quickly caught on in the community to describe these problems, and a number of publications referred to it to motivate their research. DeGrave et al. [2021] find that under close investigation, methods developed to detect Covid-19 from chest radiographs mostly learn shortcuts. D'Amour et al. [2020] discuss underspecification, the observation that many different models and sets of weights can solve the same task, as a central factor behind poor out of domain performance and shortcut learning. Mitchell [2021] discusses challenges that make AI research hard and create the illusion of human-like performance in models which are far from that goal. Hermann and Lampinen [2020] investigate which features are learned and why. Firestone [2020] argues that understanding the differences between machine and human cognition requires differentiating between "performance" and "competence". Whether as a unifying principle or in the form of the various errors it causes, shortcut learning has become a central topic in current machine learning research.

5 Discussion

The central topic of this dissertation is the difficulty of evaluating broad capabilities with benchmarks, which I investigated from two sides:

In the first part of the dissertation (Section 3), I introduced a new capability, one-shot object detection, and used it as an example to demonstrate how different benchmarks can evaluate very different aspects of a capability. In the first paper, we introduced a benchmark with heavily cluttered scenes made out of characters from different alphabets [Michaelis et al., 2018a]. We found that the trained models generalized well to novel characters, but struggled with clutter, even though characters could easily be told apart by color. As a next step, we introduced a benchmark with natural images of typical everyday scenes [Michaelis et al., 2018b]. In contrast to the first benchmark, detection and segmentation became easier, while generalization to novel categories became harder. Finally, we used the insight that the dataset in the first benchmark has a lot more categories, and used datasets with hundreds and thousands of categories, to demonstrated that in natural scenes good generalization cap be achieved by using a broad training dataset [Michaelis et al., 2020].

In the second part of the dissertation (Section 4), I broadened the view and investigated the role of the benchmark-capability gap in generalization failures of DNNs. In the paper [Geirhos et al., 2020], we identified a common pattern as the source of these generalization failures, shortcut learning. Using spurious correlations in the benchmark design and data, models can perform well on the benchmark without learning the underlying capability. This overfitting to the benchmark happens all across machine learning and leads to brittle models whose capabilities are hard to predict.

These results show the limits of benchmarks at evaluating capabilities. But they also demonstrate the potential benchmarks have as a tool for understanding and overcoming said limitations. Take the four issues mentioned in the introduction (Section 1), all of which can be better understood in the context of our results: Few-shot learning is mostly challenging when few the pre-training dataset has few categories and can potentially be solved with broader datasets (Sections 3.4 - 3.6). Noise and bad weather have such a strong impact on perception algorithms because they change image texture, on which object recognition models overly rely (Sections 4.1.1 & 4.2). And classification by background is a shortcut in standard image classification benchmarks (Section 4).

In the following sections, I will embed these results into the existing literature, to discuss more broadly why benchmarking capabilities is so challenging, how benchmarking could be improved and how investing in better evaluation enables building better models.

5.1 Evaluating Capabilities is Hard

Measurement is a key part of science and engineering. Physicists build huge machines to understand the building blocks of our universe. But any measurement can be invalidated by errors. A single incorrectly plugged in optical fiber led to a measurement that showed neutrinos moving faster than light, a result that would have contradicted Einsteins theory of relativity and caused a big stir in the physics community at that time [Strassler, 2012].

In machine learning the main tool of measurement are benchmarks and like in physics flaws can lead to false conclusions. This can be seen in the first part of the dissertation. The capability to generalize to novel categories from a single example strongly depends on the number of categories in the dataset. The same model trained and tested in the same way but on two different datasets can have very different generalization capabilities, as our experiments on COCO and LVIS show (Sections 3.5 & 3.6). Had we only looked at results for one of the datasets, we might have misjudged the abilities of our model.

What separates machine learning, and especially AI, from physics is that much of it is concerned with human capabilities. In physics, quantities are usually well-defined and results can be predicted very accurately from theory. In contrast, capabilities can be hard to define. Let us take object recognition as an example. Algorithms for image classification and object detection perform increasingly well, even surpassing human performance in some tasks [He et al., 2015a; Shankar et al., 2020]. At the same time, they are unable to handle the simple characters in *cluttered Omniglot* and fail when a bit of noise or distortions are added to and image [Geirhos et al., 2018; Hendrycks and Dietterich, 2019; Michaelis et al., 2019]. Which of the following interpretations is correct? Are CNNs good at recognizing objects and simply struggle with the delicate shapes of the Omniglot characters and extreme levels of clutter? Or is something going fundamentally wrong that cannot be measured with the standard image classification and object detection benchmarks (e.g., models having a texture bias [Geirhos et al., 2019])? What our results show is, that there is much more to understand, than the excellent performance of recent computer vision models on image classification tasks indicates.

That human capabilities are hard to evaluate is not a new insight, and misconceptions have a long history. Around 1900 the horse “Clever Hans” challenged the existing assumptions about animal cognition by being apparently able to solve arithmetic problems. A commission of researchers later found that the horse used subtle cues from his trainer, small changes in posture and movements, which were too subtle to notice. Importantly, these cues are not downright fraud but partly unconscious reactions of the horse’s instructor, and it took an extensive investigation to uncover them [Pfungst, 1911]. This is the phenomenon of shortcut learning the second part of the dissertation discusses (Section 4.2). Like “Clever Hans” models can latch onto spurious correlations such as background [Beery et al., 2018] or texture [Geirhos et al., 2019] in object recognition tasks. Similar errors can be found all across machine learning. In reinforcement learning, anecdotes about models exploiting the environment or objective pile up [Lehman et al., 2018], such as the example of a Tetris agent that learned to survive indefinitely by hitting the pause button [Murphy Vii, 2013]. In natural language processing, these failures have recently even been compared to the Clever Hans Effect [Lapuschkin et al., 2019; Heinzerling, 2020].

The key insight of Geirhos et al. [2020] (Section 4.2) is that this is not an issue of a few individual benchmarks. Most–if not all–current benchmarks can be solved in unintended ways. But these issues are often overlooked, because good performance on typical human tasks is quickly attributed to the model acquiring a capability, a tendency called anthropomorphism. Just like Clever Hans fooled the people of his time, models can fool us because we interpret their behavior in light of our own experiences. Solving this will be difficult because evaluating capabilities is notoriously hard. This can be seen in the results in this dissertation, but it also can be understood from a more principled perspective. In order to turn broad capabilities into measurable tasks, they have to be constrained in some form. While the perfect task may exist, normal tasks sacrifice some aspects of a capability and thus can not cover everything we associate with that capability. In other words, while capabilities imply meaning to a human, tasks can only measure form [Bender and Koller, 2020]. Therefore, good performance on a task does ensure acquiring the meaning people assign to it. There is an imbalance, that in the presence of a capability, a task can tell something about the extent of that capability, but performance on the task alone cannot prove the presence of the capability. The first recommendation of Geirhos et al. [2020] can be deducted from this imbalance: One should not assume a model learned a capability only because it excels on a specific task or benchmark.

Because benchmarks are so central in machine learning, the difference between benchmark performance and assumed capability has severe consequences. It can lead to wrong estimates about the capabilities of methods and the progress of research [Mitchell, 2021]. In applications, it can lead to errors and cause harm [O'Neil, 2016]. And the resulting problems can directly feed back into development. Often the task and metric define what is solved and how it is solved [Hardt and Recht, 2021, Chapter 8]. As a result, benchmarks can become closed worlds where solutions matter only within the context of the specific benchmark [Torralba and Efros, 2011]. Badly selected benchmarks can result in solving irrelevant problems [Wagstaff, 2012]. And even a set of well-designed benchmarks can potentially be problematic if it limits the diversity of approaches that are explored [Brooks, 1990]. Incorrect measurement becomes especially problematic when harmful biases or stereotypes are involved. Whether it is facial recognition models failing to identify people of color [Buolamwini and Gebru, 2018], text generation models reproducing stereotypes [Brown et al., 2020] or a hiring tool rejecting women [Dastin, 2018], machine learning systems can cause harm if their failures and the shortcuts they use are not detected.

5.2 How to Measure Better?

Designing benchmarks that better evaluate the associated capabilities will require dedicated effort. The prevalence of shortcut learning makes clear, that is not sufficient to evaluate performance on a single benchmark, to predict competence at a capability. But what is necessary to make better predictions? In Geirhos et al. [2020], we make three central recommendations (see Section 4.2.3): Interpreting results carefully, testing o.o.d. generalization and understanding what makes a solution easy to learn. The previous section already discussed why any evaluation of a capability requires careful interpretation. In the following paragraphs, I will discuss how benchmarks and evaluation will have to change to integrate the other two points.

First of all, the data used in current benchmarks should be scrutinized. Data has a history of being overlooked [Hardt and Recht, 2021], and research is often more driven by the available datasets than what is relevant in practice Wagstaff [2012]. But in the experiments in the first part of the dissertation, we find that data matters a lot. What works for synthetic datasets may not work for natural image datasets and vice versa [Michaelis et al., 2018a,b]. Data also plays an important role in shortcut learning, and improving datasets is probably the easiest and most effective way to identify and avoid shortcuts. To reduce shortcut opportunities, datasets should have as few biases as possible [Torralba and Efros, 2011], especially avoiding harmful biases [Mehrabi et al., 2021; Barocas et al., 2019]. Datasets and benchmarks should also be improved over time, to address issues that arise in their use [Beyer et al., 2020; Tsipras et al., 2020; Hardt and Recht, 2021, Chapter 8]. Before being applied in the real world, methods should be rigorously tested on multiple datasets [Welty et al., 2019; Bowman and Dahl, 2021; Henderson et al., 2018]. And to detect shortcuts early, models should be evaluated on o.o.d. generalization tasks by default [Li et al., 2017a; Michaelis et al., 2019; Yu et al., 2020b; Djolonga et al., 2021; Hendrycks et al., 2021a; Koh et al., 2021]. To make all of this feasible, new tools will be necessary to develop [Wang et al., 2020a], document [Gebru et al., 2021] and improve [Gupta et al., 2019] high-quality datasets. And to make testing models across a range of tasks a standard practice, machine learning toolkits should evolve to easily allow evaluation on multiple benchmarks.

Another important consideration are the tasks that are used for evaluation. Some shortcuts can likely be avoided by task design. For example, solving object classification by background [Beery et al., 2018] is likely much harder in object detection. But new shortcuts will appear, because even well-designed tasks can not solve the central issue that measuring a broad capability with a single number is a simplification that is bound to miss important aspects of the capability. Thus, like with datasets, it is probably a good idea to evaluate methods on multiple tasks. However, today, most methods are trained and tested on a single task, because they are developed for a specific task design, and changing the task design requires changing the method. The typical outcome is, that a small set of tasks become canonical, while most other tasks are forgotten over time [Dehghani et al., 2021]. The desired solution would be flexible models that can handle different tasks by design. Our Siamese Faster R-CNN model is a step in that direction, because the reliance on a reference allows applying it to new categories or datasets without retraining (Sections 3.5 & 3.6). A much bigger step are recent language modes, for which tasks can be specified in the prompt. Using this mode of evaluation, the static GPT-3 model [Brown et al., 2020] can be used for all kinds of applications [OpenAI, 2020], ranging from summarization to generating JavaScript code for user interfaces [Shameem, 2020]. And in computer vision, CLIP can recognize any category in ImageNet by probing it with a sentence such as "an image of a zebra" [Radford et al., 2021].

But even with the best data, the best tasks and models that can handle multiple tasks, running a number of benchmarks will likely never be sufficient to tell if a model acquired a capability. To have a capability includes being able to apply this capability to new tasks. And while prompt-based methods can in theory be evaluated on any task, they are in practice evaluated on the same well known benchmarks, just in a slightly different setting. Thus, evaluation will likely have to evolve to include methods that can generate new problems on the fly. One such task would be the Turing Test [Turing, 1950], in which a person has a conversation with a machine. The problem with this kind of interactive tests is that they cannot be easily standardized. To avoid research groups arbitrarily claiming they have "passed the Turing Test" [Westaway, 2014] it would be necessary to adopt best practices from areas such as psychology [Nesselroade and Cattell, 2013], but getting statistically significant results can be challenging and expensive. In psychology, false positive results from under-powered studies have become a major issue [Simmons et al., 2011]. Well-designed transfer tasks will likely become a necessary addition to traditional benchmarks as models come ever closer to human abilities, but they will have their own challenges and issues.

So far, I discussed how better evaluation practices can help to detect and avoid shortcuts. But to understand the inductive biases of models and why they learn certain solutions, it is also important to actively experiment with models. Many of the results in this dissertation rely on active analysis. In Section 3.6 we created subsets of datasets with varying numbers of samples and categories to study how one-shot generalization depends on these parameters. The *cluttered Omniglot* benchmark introduced in Section 3.4 was designed specifically to be able to study the effect of clutter on object identification and segmentation. And the discovery of most of the shortcuts discussed in this dissertation go back to such experiments. One problem is that it can be hard to design specific experiments. For example, there are no good measures for clutter in natural scenes [Wolfe et al., 2011; Rosenholtz et al., 2007], thus making a study of clutter in natural images hard. But, with some ingenuity and the help of modern data generation tools such as generative models, very interesting tasks can be created, such as the texture and shape cue conflict task used to identify the texture bias of CNNs by Geirhos et al. [2019]. While assessing the performance of methods on a set of standardized benchmarks will remain an integral part of machine learning research, deep understanding of the methods will require additional active investigations. This type of investigative work is of course already done, but often gets less credit than the development of new methods [Rahimi, 2017; Birhane et al., 2021].

While specific developments are hard to predict, the general direction is clear. To move forward in the face of shortcuts, evaluation will have to take on greater importance. Some changes will be easy to make. For example, in one-shot object detection and likely for most few-shot learning tasks, controlling the number of categories used for training is important to judge the generalization capabilities of a method. In other cases, changes will be more difficult. There are many aspects of robustness, thus while a number of robustness benchmarks exist [Hendrycks and Dietterich, 2019; Hendrycks et al., 2021b,a], some types of robustness have surely been overlooked. And for something like adversarial robustness no fixed benchmark can exist, but benchmarks have to continuously evolve to include the most recent defense mechanisms [Carlini et al., 2019; Brendel et al., 2019]. In yet other areas it is unclear, how the desired capability can be evaluated at all. Object recognition is such a case. While traditional object recognition benchmarks such as ImageNet [Russakovsky et al., 2015] are susceptible to shortcuts such as texture bias [Geirhos et al., 2019] or classification by background [Beery et al., 2018], that does not mean that benchmarks which cover these shortcuts, e.g., by including measures of texture bias [Hermann et al., 2020], evaluate object recognition. Rather, as our conflicting results on *clutteredOmniglot* and COCO show, the extent of CNN object recognition and the way to evaluate it is quite unclear.

The good news is that the research community has made a similar step before when it switched from qualitative evaluation on individual images to quantitative evaluation on datasets [Efros, 2020; LeCun et al., 1998; Martin et al., 2001], and it is doing it again, addressing many of the above issues. In computer vision, a number of recent studies investigated the limitations of ImageNet [Recht et al., 2019; Kornblith et al., 2019; Engstrom et al., 2020; Beyer et al., 2020; Tsipras et al., 2020; Taori et al., 2020; Djolonga et al., 2021] and proposed new test sets to evaluate robustness [Geirhos et al., 2018; Hendrycks and Dietterich, 2019; Hendrycks et al., 2021b,a; Barbu et al., 2019]. Many new datasets were created in the last years [Krishna et al., 2016; Zhou et al., 2017; Gupta et al., 2019; Kuznetsova et al., 2020; Yu et al., 2020a; Caesar et al., 2019; Wang et al., 2019b; Sun et al., 2020; Wang et al., 2018, 2019a] and a lot of attention was paid to harmful biases in datasets [Buolamwini and Gebru, 2018; Prabhu and Birhane, 2021; Mehrabi et al., 2021]. This led to best practices for dataset generation [Hutchinson et al., 2021; Jo and Gebru, 2020], as well as tools to analyze [Wang et al., 2020a] and document datasets [Gebru et al., 2021]. Projects such as the Robust Vision Challenge [Zendel et al., 2020], which measures performance on multiple benchmarks, and Dynabench [Kiela et al., 2021], an evolving benchmark with multiple rounds of new samples, explore new evaluation strategies. And the incentives to publish new benchmarks and evaluation strategies grow, with NeurIPS offering a benchmarks and dataset track for the first time last year [Vanschoren and Yeung, 2021].

Despite these encouraging steps, improving evaluation techniques that encapsulate the underlying capability so well that they can not be fooled by shortcut learning remains a daunting task. Scrutinizing each result whether it is a shortcut, questioning every benchmark, whether it measures what it is supposed to measure, and hunting for possible failures can appear to be a tedious exercise. Thus, it may make sense to change the question from how current benchmarks can be improved to what it takes to evaluate broad capabilities.

5.3 Is it Possible to Measure Reality?

What is necessary to develop an evaluation method for broad capabilities? A surprisingly relevant mental model for this question can be found in the almost 2500-year-old *Allegory of the Cave* by Plato [Plato and Reeves, 2004]. It describes a group of prisoners who are kept their whole life in a cave. They are fixed in their location and all they see is a shadow play that is performed for them on one of the caves walls. Plato argues that "what the prisoners would take for true reality is nothing other than the shadows" [Plato and Reeves, 2004, 515c]. And if one of the prisoners was freed and could leave the cave he would be confused and blinded by the bright daylight and, at first, "unable to see a single one of the things now said to be truly real" [Plato and Reeves, 2004, 516a]. Similar to how the prisoners in the cave only experience the world through shadow plays, current object recognition benchmarks rely solely on static 2D images. And like the prisoners in the cave who are not be prepared for the real world by the shadow plays, models fail outside their training distribution. The degree of realism between a shadow play and a photograph may be different, but both are representations of reality and not reality itself.

In the previous section, I discussed how current benchmarks could be adapted to reduce shortcuts and better evaluate the underlying capabilities. In the image of the cave, this corresponds to creating a more realistic shadow play. But to get rid of most shortcuts altogether, the solution may be to measure performance in the real world outside the cave. Many shortcuts that work on the current benchmarks would instantly fail in reality.

But evaluating models in the real world is easier said than done. First, there is the reality gap in the data. Humans experience a physical three-dimensional world that changes continuously over time. They can interact with this world and experience cause and effect of their actions, and these interactions shape perception [Grezes and Decety, 2002]. Re-creating the world with all details and across all sensor modalities is a daunting task that is likely impossible to achieve with current computing resources. To get a sense of the scale, consider that most CNNs are trained on images smaller than one megapixel, while our eyes sample the world with roughly 500 megapixels at 30 Hz. And even if this kind of processing power was available, a lot of the training and testing process would likely have to be done in simulators to be efficient, thus requiring extremely realistic representations of reality. Current simulators are far from this level of realism, and models trained on synthetic data usually perform worse on real data [Tobin et al., 2017]. But the reality gap is not limited to the data. The tasks humans usually perform are very different from typical machine learning tasks, such as image classification. For example, making coffee in the morning requires going through a number of steps, the easiest of which include detecting the coffee maker, coffee beans and a mug. The metric is difficult as well. Of course, one could just check if the coffee was made. But how many mugs can be broken during the process? How long can it take, and does the coffee have to taste good?

This leads to a central challenge: In reality, a lot of things are ambiguous. Of course, most physical properties are objective, and tasks such as depth estimation have a well-defined solution in almost all cases. But many questions do not have objective answers. Object recognition has many aspects that are subjective. A zebra is a zebra, but it is also an animal, a mammal, a living being et cetera. The same object can have different function in different contexts. In many cases, this hierarchical structure can be evaluated by accepting all the above answers. But what about attributes. Stripes are stripes, but what is a cute zebra in a zoo can be a very problematic zebra in your kitchen. Many attributes depend on context and can vary from person to person. It is not even always clear, what constitutes an object. The zebra is an object, but so is its head, ear, and each single hair in its fur. Any type of semantic labelling has ambiguities that stem from the fact that our definitions can vary and depend on context. Humans are excellent at handling these ambiguities, and therefore it is a reasonable goal to develop models which can do the same. But measuring if models achieved this goal will be a challenge. And the problems do not stop there. Being able to solve numerous ambiguous tasks is not enough to master reality. If you encounter a zebra in your kitchen on the way to the coffee maker, you suddenly face a number of new tasks, many of which you likely have never done before. As discussed in the previous sections, solving new tasks is a crucial element of measuring deep understanding, because almost any fixed task can be solved by a shortcut. However, designing a measurement system which can generate infinitely many reasonable tasks is a challenge whose difficulty is hard to predict.

While some of these issues may be overcome with engineering and improvements in computer power, there are also fundamental issues with modelling reality. Sometimes reproducing reality is not desired, especially when data has known biases, such as for example past hiring decisions which are usually biased towards hiring men [Dastin, 2018]. In other cases, learning the exact solution may be inefficient or not even possible. Shortcuts can be helpful to save energy in case where a true solution is not needed or not obtainable.

Often simple solutions are great as long as their limits are known. Newton's mechanics are superior to quantum mechanics at describing the movement of planets, but not suited to describe the behavior of electrons and atoms.

As the above points show, measuring reality is at best not trivial and at worst impossible. But it can serve as a guiding system for developing new forms of measurement. Benchmarks are one way to make reality measurable by taking a fixed sample of data points from reality with clearly defined labels. To make progress from there, the first steps may be to make the data more real or to add some of the flexibility reality requires into the task. This can be quite similar to the recommended made in the previous section. The central point I want to make is not that the only solution is to measure reality, but that it is important to keep the goal of measuring reality in mind, when developing new methods of measurement. Ultimately, the objective is a question of trade-offs. Humans are surprisingly good at handling tradeoffs, contradictions and ambiguities, therefore building models which can handle them is a reasonable goal. But, developing evaluation methods and models which can handle these tradeoffs will be a challenge.

5.4 How to Build Better Models?

The discussion in the last sections has made clear that current benchmarks are far from measuring human-level capabilities. This leads to all kinds of issues because it allows models to learn shortcuts instead of acquiring the underlying capability. So it is clear that the limitations of current benchmarks are at least partly responsible for the current human-machine gap. But what is their role in closing this gap?

The results in Michaelis et al. [2020] (Section 3.6) demonstrate, that careful analysis of benchmark results can help identify which changes are necessary to overcome shortcomings of existing methods. This principle is especially applicable in the case of shortcut learning. Once a shortcut is identified, targeted solutions can be developed. For example, Geirhos et al. [2019] find that image augmentation using style transfer [Gatys et al., 2016], which exchanges the texture features in an image, leads to models with a shape bias. But identifying and blocking shortcuts one-by-one will likely fail. There are usually a number of shortcuts present, and closing all of them may quickly turn into a "Whack a Mole" situation. Even if a shortcut has been identified and blocked it is not given, that this solution performs better in other settings [Hermann et al., 2020]. And a model that does not make certain mistakes is not the same as a model which has acquired the underlying capability. But through evaluation and careful analysis can still lead to improvements, if they help to identify larger patterns. Using style transfer as a data augmentation does not only induce a shape bias, but also reduces the impact of other image distortions, which usually affect textures much more than object shape [Geirhos et al., 2019; Michaelis et al., 2019]. And using a broad dataset for training does not only make it harder to memorize the training categories, but also provides the model with additional context, which the model can use to better understand the world [Michaelis et al., 2020; Kolesnikov et al., 2019].

But there is an alternative direction that proved very fruitful in recent years, and that is simply scaling up models and datasets. This idea that simply scaling up models and datasets may deliver better results than understanding and solving fundamental problems has been called the "bitter lesson" of AI research [Sutton, 2019]. That more data is a powerful way to improve models is long known, and is sometimes called the unreasonable effectiveness of data [Halevy et al., 2009; Sun et al., 2017]. But recently, breakthroughs in self-supervised learning [Oord et al., 2018; Chen et al., 2020a; Devlin et al., 2019; Radford et al., 2019] and model architecture [Vaswani et al., 2017; Kolesnikov et al., 2019; Liu et al., 2022] allowed training models of unprecedented scale on datasets of unprecedented scale [Devlin et al., 2019; Brown et al., 2020; Radford et al., 2021; Jia et al., 2021; Ramesh et al., 2021; Baevski et al., 2022]. These do not only outperform all previous methods on standard benchmarks, but are also more robust and excel at few-shot generalization [Djolonga et al., 2021; Taori et al., 2020; Kolesnikov et al., 2019; Brown et al., 2020; Radford et al., 2021; Jia et al., 2021]. Scaling has been shown to solve issues such as classification by background [Kolesnikov et al., 2019]. It can facilitate long term planning in reinforcement learning [OpenAI, 2018]. And it led to zero-shot language models which can perform new tasks simply by providing the right prompt [Brown et al., 2020]. The relationship between data, model size, training compute and performance is so clear, that they can be described with exponential "scaling laws" [Kaplan et al., 2020; Henighan et al., 2020]. In its extreme, this has led to the formulation of the scaling hypothesis [Gwern, 2020]: Simply scaling models and data will lead to human level intelligence.

The question is how far this can go? So far, even the biggest models trained on the largest datasets still take shortcuts [Bommasani et al., 2021]. In Plato's *Allegory of the Cave*, the shadow plays the prisoners see are not sufficient to understand the reality outside the cave [Plato and Reeves, 2004]. It is unclear if Plato had made the same argument if the prisoners were shown images with a projector, but it is not unlikely. It is known that 3D information [Kestenbaum et al., 1987], motion [Spelke, 1990] and interaction [Grezes and Decety, 2002] are elemental to human object perception and images can not accurately represent them. Or intuitively, if you see a zebra moving, the difference between its shape and texture is directly apparent. It is not unlikely that as for benchmarks (see discussion in Section 5.3), training data and tasks have to come as close as possible to reality to give models any chance to learn broad capabilities. The tasks that are used to train many of the most powerful recent models already added complexity in the way they model language [Brown et al., 2020; Radford et al., 2021].

One reason why scaling had such success not only on the training tasks, but also at improving generalization is that for all their flaws, progress on benchmarks seems to be surprisingly well aligned with progress on the underlying capability. As I discussed in the last sections, evaluating a capability with a benchmark is extremely hard, if not impossible. But this does not mean that progress on a benchmark cannot be aligned with progress towards the associated capability. And in fact, for many popular benchmarks this appears to be the case. Standard benchmark accuracy is often the most predictive factor for performance in generalization tasks, whether it is transfer learning [Kornblith et al., 2019; Kolesnikov et al., 2019; Djolonga et al., 2021], few-shot learning [Kolesnikov et al., 2019; Brown et al., 2020], or o.o.d. generalization [Taori et al., 2020; Djolonga et al., 2021]. But it is clear that they miss central aspects of their associated capabilities, so how far can they be used as a tool to measure progress? In the last part of the allegory, Plato considers what would happen if a person returned to the cave. Having seen the real world, the shadows would have lost their meaning as a way to understand the world and predict the future. He would no longer be eager to be the best at interpreting the shadows, and "if he had to compete once again with the perpetual prisoners in recognizing the shadows [...] he [would likely] provoke ridicule" [Plato and Reeves, 2004, 517a]. Just as the person at the end of Plato's Allegory who is led back into the cave understands the shadow plays less well than his peers in the cave, models may seem to move backwards on the existing benchmarks when they come closer to acquiring broad capabilities. For example, reducing the shape bias of a model leads to worse, rather than better, performance on ImageNet [Hermann et al., 2020]. And the human baseline for ImageNet performance is much lower than that of recent models [He et al., 2015a]. While benchmarks are an amazing way to measure progress "within the cave", it may be necessary to let go of them in order to move out of the cave.

It is unclear, how far machines are from achieving human-level capabilities. As demonstrated in this dissertation, the current set of narrow benchmarks will not be sufficient to measure the gap. And shortcut learning is a clear indicator, that current models are missing significant aspects of the capabilities they are supposed to learn. At the same time, progress in areas such as few-shot learning and robustness is fast and, in the face of shortcut learning maybe surprisingly, current benchmarks are still indicative of this progress. Whatever the key ingredients for the route ahead are, it is an exciting time to do AI research and develop machine learning models, because we will likely encounter the results in our everyday lives.

Acknowledgements

This dissertation is the result of four amazing years in the labs of Matthias Bethge and Alex Ecker. Thus, first and foremost I want to thank Matthias, Alex and Wieland, for creating this environment and offering me an opportunity to become a part of it. The energy and excitement of everyone, from the professors and postdocs over my fellow PhD students to the administrative staff, made working there an amazing experience. I am truly grateful I got to know them, and am happy many turned from colleagues to friends over time. I especially want to thank my collaborators and co-authors. A special role took Robert, Jörn, Evi and Ivan, with whom I worked as equal partners and often shared first authorship. The collaborations with them were some of the most fun and productive experiences I had. I am also deeply grateful to the administrative staff. The support Heike, Tina, Melanie, and many others provided with everything from contracts to calendars was simply phenomenal.

What made the lab special applies in a similar form to the wider ML community in Tübingen. The many bright and passionate people everywhere were a never ending source of ideas and motivation. So thanks to the IMPRS-IS graduate school that hosted my PhD, the University and MPI who hosted us physically and the many institutes and centers that shaped this surrounding. A special thanks go to the senior co-authors I worked with, especially Felix Wichmann and Rich Zemel, and the external members of my Thesis Advisory Committee, Andreas Geiger and Michael Black, for their mentorship and birds-eye perspective.

The last thank-you goes to my friends and family, who supported me through all ups and downs and make my life wonderful.

6 Bibliography

Anurag Arnab and Philip H. S. Torr. Pixelwise instance segmentation with a dynamically instantiated network. In *CVPR*, 2017.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *TPAMI*, 39(12):2481–2495, 2017.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. *arXiv:2202.03555*, 2022.

Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *CVPR*, 2017.

David Balduzzi, Karl Tuyls, Julien Perolat, and Thore Graepel. Re-evaluating evaluation. In *NeurIPS*, 2018.

Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *NeurIPS*, 2019.

Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. `http://www.fairmlbook.org`.

Peyman Bateni, Raghav Goyal, Vaden Masrani, Frank Wood, and Leonid Sigal. Improved few-shot visual classification. In *CVPR*, 2020.

Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, 2010.

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *ECCV*, 2018.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Emily M Bender and Alexander Koller. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *ACL*, 2020.

Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv:1912.06680*, 2019.

Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *ICLR*, 2019.

Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv:2006.07159*, 2020.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint ECML PKDD*, 2013.

Abeba Birhane, Pratyusha Kalluri, Dallas Card, William Agnew, Ravit Dotan, and Michelle Bao. The values encoded in machine learning research. *arXiv:2106.15590*, 2021.

Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv:2004.10934*, 2020.

Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *ICCV*, 2017.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv:2108.07258*, 2021.

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Nazanin Mohammadi Sepahvand, Edward Raff, Kanika Madan, Vikram Voleti, et al. Accounting for variance in machine learning benchmarks. *Machine Learning and Systems*, 2021.

Samuel Bowman and George Dahl. What will it take to fix benchmarking in natural language understanding? In *Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4843–4855, 2021.

Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *ICLR*, 2019.

Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *ICLR*, 2018.

Wieland Brendel, Jonas Rauber, Matthias Kümmerer, Ivan Ustyuzhaninov, and Matthias Bethge. Accurate, reliable and fast robustness evaluation. *NeurIPS*, 2019.

Rodney A Brooks. Elephants don't play chess. *Robotics and Autonomous Systems*, 6(1-2):3–15, 1990.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv:2005.14165*, 2020.

Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *ACM FAccT*, 2018.

Santiago A Cadena, George H Denfield, Edgar Y Walker, Leon A Gatys, Andreas S Tolias, Matthias Bethge, and Alexander S Ecker. Deep convolutional models improve predictions of macaque v1 responses to natural images. *PLoS computational biology*, 15(4):e1006897, 2019.

Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, 2017.

Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool. The 2019 davis challenge on vos: Unsupervised multi-object segmentation. *arXiv:1905.00737*, 2019.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv:1903.11027*, 2019.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv:1902.06705*, 2019.

Ding-Jie Chen, He-Yen Hsieh, and Tyng-Luh Liu. Adaptive image transformer for one-shot object detection. In *CVPR*, 2021a.

Hao Chen, Yali Wang, Guoyou Wang, and Yu Qiao. LSTD: A Low-Shot Transfer Detector for Object Detection. *AAAI*, 2018.

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv:1906.07155*, 2019a.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 40(4):834–848, 2017.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *ICML*, 2020a.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*, 2020b.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *ICLR*, 2019b.

Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollár. Tensormask: A foundation for dense object segmentation. In *ICCV*, 2019c.

Yinbo Chen, Xiaolong Wang, Zhuang Liu, Huijuan Xu, and Trevor Darrell. A new meta-baseline for few-shot learning. In *ICCV*, 2021b.

Zhiyuan Chen and Bing Liu. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3):1–207, 2018.

Christine Chin and David E Brown. Learning in science: A comparison of deep and surface approaches. *Journal of Research in Science Teaching*, 37(2):109–138, 2000.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005.

Kenneth Ward Church. Emerging trends: I did it, i did it, i did it, but. . . *Natural Language Engineering*, 23(3):473–480, 2017.

Kenneth Ward Church and Joel Hestness. A survey of 25 years of evaluation. *Natural Language Engineering*, 25(6):753–767, 2019. doi: 10.1017/S1351324919000275.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019.

Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In *Conference on Recommender Systems*, 2019.

Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015.

Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun. Instance-sensitive fully convolutional networks. In *ECCV*, 2016a.

Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016b.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *JMLR*, 2020.

Jeffrey Dastin. Amazon scraps secret AI recruiting tool that showed bias against women, 2018. URL `https://reut.rs/2Od9fPr`. (Accessed: 2021-12-04).

Alex J DeGrave, Joseph D Janizek, and Su-In Lee. Ai for radiographic covid-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 3(7):610–619, 2021.

Mostafa Dehghani, Yi Tay, Alexey A. Gritsenko, Zhe Zhao, Neil Houlsby, Fernando Diaz, Donald Metzler, and Oriol Vinyals. The benchmark lottery. *arXiv:2107.07002*, 2021.

Sebastian Deterding, Miguel Sicart, Lennart Nacke, Kenton O'Hara, and Dan Dixon. Gamification. using game-design elements in non-gaming contexts. In *Extended Abstracts on Human Factors in Computing Systems*, pages 2425–2428. ACM, 2011.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. *arXiv:1909.02729*, 2019.

Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D'Amour, Dan Moldovan, et al. On robustness and transferability of convolutional neural networks. In *CVPR*, 2021.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A Smith. Show your work: Improved reporting of experimental results. In *Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*, 2019.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

Xuanyi Dong, Liang Zheng, Fan Ma, Yi Yang, and Deyu Meng. Few-Example Object Detection with Model Communication. *TPAMI*, 41(7):1641–1654, 2018.

Vincent Dumoulin, Neil Houlsby, Utku Evci, Xiaohua Zhai, Ross Goroshin, Sylvain Gelly, and Hugo Larochelle. A unified few-shot classification benchmark to compare transfer and meta learning approaches. In *NeurIPS Datasets and Benchmarks Track*, 2021.

Alexei Efros. Imagining a post-dataset era. Talk at the ICML 2020 Workshop on Continual Learning, 2020. URL `https://slideslive.com/38930883/imagining-a-postdataset-era`.

Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *ICML*, 2019.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Jacob Steinhardt, and Aleksander Madry. Identifying statistical bias in dataset replication. In *ICML*, 2020.

Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639): 115–118, 2017.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, 88(2):303–338, 2010.

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR Workshop on Generative-Model Based Vision*, 2004.

Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *TPAMI*, 28(4):594–611, 2006.

Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2009.

Robert Fergus, Pietro Perona, and Andrew Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML*, 2017.

Chaz Firestone. Performance vs. competence in human–machine comparisons. *PNAS*, 117(43):26562–26571, 2020.

Shanghua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip HS Torr. Res2net: A new multi-scale backbone architecture. *TPAMI*, 2019.

Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *NeurIPS*, 2015.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12): 86–92, 2021.

Robert Geirhos, Carlos M. Medina Temme, Jonas Rauber, Heiko H Schütt, Matthias Bethge, and Felix A Wichmann. Generalisation in humans and deep neural networks. arXiv:1808.08750, 2018.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *ICLR*, 2019.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *arXiv:2004.07780*, 2020.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*, 2014.

Google Inc. Google images, 2020. URL `https://images.google.com/`. (Accessed: 2020-12-29).

Kristen Grauman and Trevor Darrell. Unsupervised learning of categories from sets of partially matching image features. In *CVPR*, 2006.

Julie Grezes and Jean Decety. Does visual perception of object afford action? evidence from a neuroimaging study. *Neuropsychologia*, 40(2):212–222, 2002.

Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019.

Gwern. The scaling hypothesis, 2020. URL `https://www.gwern.net/Scaling-hypothesis`. (Accessed: 26.12.2021).

Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *Intelligent Systems*, 2009.

Moritz Hardt and Benjamin Recht. *Patterns, predictions, and actions: A story about machine learning.* `https://mlstory.org`, 2021. (Accessed: 2021-12-15).

Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *CVPR*, 2017.

Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *ECCV*, 2014.

Susan Hassler. 2017: The year of self-driving cars and trucks. *IEEE Spectrum*, 2016. URL `https://spectrum.ieee.org/2017-the-year-of-selfdriving-cars-and-trucks`. (Accessed: 2021-12-04).

Elad Hazan. *Introduction to online convex optimization.* Independently published, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *TPAMI*, 37(9):1904–1916, 2015b.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *ICLR*, 2020.

Benjamin Heinzerling. Nlp's clever hans moment has arrived. *Journal of Cognitive Science*, 2020.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *AAAI*, 2018.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, 2021b.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv:2010.14701*, 2020.

Katherine Hermann and Andrew Lampinen. What shapes feature representations? exploring datasets, architectures, and training. *NeurIPS*, 2020.

Katherine Hermann, Ting Chen, and Simon Kornblith. The origins and prevalence of texture bias in convolutional neural networks. *NeurIPS*, 2020.

Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. One-shot object detection with co-attention and co-excitation. In *NeurIPS*, 2019.

Kuang-Jui Hsu, Yen-Yu Lin, and Yung-Yu Chuang. Deepco^3: Deep instance co-segmentation by co-peak search and co-saliency detection. In *CVPR*, 2019.

Houdong Hu, Yan Wang, Linjun Yang, Pavel Komlev, Li Huang, Xi Chen, Jiapei Huang, Ye Wu, Meenaz Merchant, and Arun Sacheti. Web-scale responsive visual search at bing. In *KDD*, 2018a.

Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to segment every thing. In *CVPR*, 2018b.

Yuqing Hu, Vincent Gripon, and Stéphane Pateux. Leveraging the feature distribution in transfer-based few-shot learning. In *ICANN*, 2021.

Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *CVPR*, 2017.

Ben Hutchinson, Andrew Smart, Alex Hanna, Emily Denton, Christina Greer, Oddur Kjartansson, Parker Barnes, and Margaret Mitchell. Towards accountability for machine learning datasets: Practices from software engineering and infrastructure. In *Conference on Fairness, Accountability, and Transparency*, 2021.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *NeurIPS*, 2019.

Joern-Henrik Jacobsen, Jens Behrmann, Richard Zemel, and Matthias Bethge. Excessive invariance causes adversarial vulnerability. In *ICLR*, 2018.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *EMNLP*, 2017.

Shuqiang Jiang, Yaohui Zhu, Chenlong Liu, Xinhang Song, Xiangyang Li, and Weiqing Min. Dataset bias in few-shot image recognition. *arXiv:2008.07960*, 2020.

Yushi Jing and Shumeet Baluja. Visualrank: Applying pagerank to large-scale image search. *TPAMI*, 30 (11):1877–1890, 2008.

Yushi Jing, David Liu, Dmitry Kislyuk, Andrew Zhai, Jiajing Xu, Jeff Donahue, and Sarah Tavel. Visual search at pinterest. In *KDD*, 2015.

Eun Seo Jo and Timnit Gebru. Lessons from archives: Strategies for collecting sociocultural data in machine learning. In *Conference on Fairness, Accountability, and Transparency*, 2020.

Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *TPAMI*, 34(7):1409–1422, 2011.

Travis Kalanick. Tweet, 2015. URL `https://twitter.com/travisk/status/564072341395632128`. (Accessed: 2021-12-04).

Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. *arXiv:1812.01866*, 2018.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv:2001.08361*, 2020.

Alexander C. Kaufman. Elon musk: We'll have driverless cars by 2023, 2014. URL `https://www.huffpost.com/entry/tesla-driverless-cars_n_5990136`. (Accessed: 2021-12-04).

Roberta Kestenbaum, Nancy Termine, and Elizabeth S Spelke. Perception of objects and object boundaries by 3-month-old infants. *British Journal of Developmental Psychology*, 5(4):367–383, 1987.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, et al. Dynabench: Rethinking benchmarking in nlp. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2021.

Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. Edge-labeling graph neural network for few-shot learning. In *CVPR*, 2019.

Alexander Kirillov, Evgeny Levinkov, Bjoern Andres, Bogdan Savchynskyy, and Carsten Rother. Instancecut: From edges to instances with multicut. In *CVPR*, 2017.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *PNAS*, 114(13):3521–3526, 2017.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese Neural Networks for One-shot Image Recognition. *ICML*, 2015.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *ICML*, 2021.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Large scale learning of general visual representations for transfer. *arXiv:1912.11370*, 2019.

Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *CVPR*, 2019.

Nikolaus Kriegeskorte. Deep neural networks: A new framework for modeling biological vision and brain information processing. *Annual Review of Vision Science*, 2015.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. In *arXiv:1602.07332*, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 128(7):1956–1981, 2020.

Adrienne LaFrance. The high-stakes race to rid the world of human drivers. *The Atlantic*, 2015. URL `https://www.theatlantic.com/technology/archive/2015/12/driverless-cars-are-this-centurys-space-race/417672/`. (Accessed: 2021-12-04).

Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking Clever Hans predictors and assessing what machines really learn. *Nature Communications*, 10(1):1096, 2019.

Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *NeurIPS*, 1990.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 2015.

Kai-Fu Lee and Chen Qiufan. *AI 2041*. Currency, 2021.

Joel Lehman, Jeff Clune, Dusan Misevic, Christoph Adami, Lee Altenberg, Julie Beaulieu, Peter J. Bentley, Samuel Bernard, Guillaume Beslon, David M. Bryson, Patryk Chrabaszcz, Nick Cheney, Antoine Cully, Stéphane Doncieux, Fred C. Dyer, Kai Olav Ellefsen, Robert Feldt, Stephan Fischer, Stephanie Forrest, Antoine Frénoy, Christian Gagné, Leni K. Le Goff, Laura M. Grabowski, Babak Hodjat, Frank Hutter, Laurent Keller, Carole Knibbe, Peter Krcah, Richard E. Lenski, Hod Lipson, Robert MacCurdy, Carlos Maestre, Risto Miikkulainen, Sara Mitri, David E. Moriarty, Jean-Baptiste Mouret, Anh Nguyen, Charles Ofria, Marc Parizeau, David P. Parsons, Robert T. Pennock, William F. Punch, Thomas S. Ray, Marc Schoenauer, Eric Shulte, Karl Sims, Kenneth O. Stanley, François Taddei, Danesh Tarapore, Simon Thibault, Westley Weimer, Richard Watson, and Jason Yosinksi. The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities. *arXiv:1803.03453*, 2018.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. Question and answer test-train overlap in open-domain question answering datasets. In *EACL*, 2021.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, 2017a.

Weihao Li, Omid Hosseini Jafari, and Carsten Rother. Deep object co-segmentation. In *ACCV*, 2018.

Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, 2017b.

Jimmy Lin. The neural hype and comparisons against weak baselines. In *ACM SIGIR Forum*, 2019.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *CVPR*, 2017a.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017b.

Zachary C Lipton and Jacob Steinhardt. Troubling trends in machine learning scholarship: Some ml papers suffer from flaws that could mislead the public and stymie future research. *Queue*, 2019.

Shu Liu, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. Sgn: Sequential grouping networks for instance segmentation. In *ICCV*, 2017.

Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. In *ICLR*, 2019.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *CVPR*, 2022.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

Timo Luddecke and Alexander Ecker. The role of data for one-shot semantic segmentation. In *CVPR Workshop on Learning From Limited or Imperfect Data*, 2021.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *JMLR*, 11(1):19–60, 2010.

Gary Marcus. Deep learning: A critical appraisal. *arXiv:1801.00631*, 2018.

Benjamin Marie, Atsushi Fujita, and Raphael Rubino. Scientific credibility of machine translation research: A meta-evaluation of 769 papers. In *Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, 2021.

David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001.

James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419, 1995.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys*, 2021.

Claudio Michaelis, Matthias Bethge, and Alexander S. Ecker. One-Shot segmentation in clutter. *arXiv:1803.09597*, 2018a.

Claudio Michaelis, Ivan Ustyuzhaninov, Matthias Bethge, and Alexander S. Ecker. One-Shot instance segmentation. *arXiv:1811.11507*, 2018b.

Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *NeurIPS Workshop on Machine Learning for Autonomous Driving*, 2019.

Claudio Michaelis, Matthias Bethge, and Alexander S. Ecker. Closing the Generalization Gap in One-Shot Object Detection. *arXiv:2011.04267*, 2020.

Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv:1603.00831*, 2016.

George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

Melanie Mitchell. Why ai is harder than we think. In *Mind Design 3*. MIT Press, 2021.

Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *NeurIPS*, 2014.

James L. Morgan. Learnability considerations and the nature of trigger experiences in language acquisition. *Behavioral and Brain Sciences*, 12(2):352–353, 1989. doi: 10.1017/S0140525X00049050.

Tom Murphy Vii. The first level of super mario bros. is easy with lexicographic orderings and time travel... after that it gets a little tricky. In *SIGBOVIK*, 2013.

Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *ECCV*, 2020.

Preetum Nakkiran. A discussion of'adversarial examples are not bugs, they are features': Adversarial examples are just bugs, too. *Distill*, 4(8):e00019–5, 2019.

Akhila Narla, Brett Kuprel, Kavita Sarin, Roberto Novoa, and Justin Ko. Automated classification of skin lesions: from pixels to practice. *Journal of Investigative Dermatology*, 138(10):2108–2110, 2018.

John R Nesselroade and Raymond B Cattell. *Handbook of multivariate experimental psychology*. Springer Science & Business Media, 2013.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, 2015.

Carlton Wayne Niblack, Ron Barber, Will Equitz, Myron D Flickner, Eduardo H Glasman, Dragutin Petkovic, Peter Yanker, Christos Faloutsos, and Gabriel Taubin. Qbic project: querying images by content, using color, texture, and shape. In *Storage and Retrieval for Image and Video Databases*. International Society for Optics and Photonics, 1993.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv:1803.02999*, 2018.

Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015.

Curtis G Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. In *NeurIPS Datasets and Benchmarks Track*, 2021.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016.

Cathy O'Neil. *Weapons of Math Destruction.* Crown Books, 2016.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv:1807.03748*, 2018.

OpenAI. Openai five, 2018. URL `https://openai.com/blog/openai-five/`. (Accessed: 2021-12-22).

OpenAI. Openai api. https://openai.com/api/, 2020. (Accessed: 2022-3-30).

Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *CVPR*, 2015.

Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, 2018.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *KDD*, 2014.

Oskar Pfungst. *Clever Hans: (the horse of Mr. Von Osten.) a contribution to experimental animal and human psychology.* Holt, Rinehart and Winston, 1911.

James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008.

Pedro O Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. *NeurIPS*, 2015.

Pedro O Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *ECCV*, 2016.

Matthew Pitropov, Danson Evan Garcia, Jason Rebello, Michael Smart, Carlos Wang, Krzysztof Czarnecki, and Steven Waslander. Canadian adverse driving conditions dataset. *International Journal of Robotics Research*, 40(4-5):681–690, 2021.

Plato and C. D. C. Reeves. *Republic.* Hackett Publishing Company, Inc, 2004.

Jean Ponce, Tamara L Berg, Mark Everingham, David A Forsyth, Martial Hebert, Svetlana Lazebnik, Marcin Marszalek, Cordelia Schmid, Bryan C Russell, Antonio Torralba, et al. Dataset issues in object recognition. In *Toward Category-Level Object Recognition.* Springer, 2006.

Vinay Uday Prabhu and Abeba Birhane. Large image datasets: A pyrrhic win for computer vision? In *WACV*, 2021.

Marcelo OR Prates, Pedro H Avelar, and Luís C Lamb. Assessing gender bias in machine translation: a case study with google translate. *Neural Computing and Applications*, 32(10):6363–6381, 2020.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *ICLR*, 2020.

Ali Rahimi. Neurips 2017 test-of-time award presentation, 2017. URL `https://www.youtube.com/watch?v=Qi1Yry33TQE`. (Accessed: 2020-11-26).

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Mark Chen, Rewon Child, Vedant Misra, Pamela Mishkin, Gretchen Krueger, Sandhini Agarwal, and Ilya Sutskever. Dall·e: Creating images from text. *OpenAI blog*, 2021.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, 2019.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *ICLR*, 2018.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM Computing Surveys*, 54(9), 2021.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. *NeurIPS*, 2019.

Michael Roberts, Derek Driggs, Matthew Thorpe, Julian Gilbey, Michael Yeung, Stephan Ursprung, Angelica I Aviles-Rivero, Christian Etmann, Cathal McCague, Lucian Beer, et al. Common pitfalls and recommendations for using machine learning to detect and prognosticate for covid-19 using chest radiographs and ct scans. *Nature Machine Intelligence*, 3(3):199–217, 2021.

Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.

Ruth Rosenholtz, Yuanzhen Li, and Lisa Nakano. Measuring visual clutter. *Journal of Vision*, 7(2):17–17, 2007.

Philip E. Ross. Ford: Robotaxis in 2021, self-driving cars for consumer 2025. *IEEE Spectrum*, 2016. URL `https://spectrum.ieee.org/ford-robotaxis-in-2021-selfdriving-cars-for-consumer-2025`. (Accessed: 2021-12-04).

Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *CVPR*, 2006.

Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. *CVPR*, 2013.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *IJCV*, 2018.

Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M Aroyo. "everyone wants to do the model work, not the data work": Data cascades in high-stakes ai. In *Conference on Human Factors in Computing Systems*, 2021.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, 2016.

Othman Sbai, Camille Couprie, and Mathieu Aubry. Impact of base dataset design on few-shot image classification. In *ECCV*, 2020.

Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Sharathchandra Pankanti, Rogerio Feris, Abhishek Kumar, Raja Giryes, and Alex M. Bronstein. RepMet: Representative-based metric learning for classification and one-shot object detection. In *CVPR*, 2019.

Karen Scouller. The influence of assessment method on students' learning approaches: Multiple choice question examination versus assignment essay. *Higher Education*, 35(4):453–472, 1998.

David Sculley, Jasper Snoek, Alex Wiltschko, and Ali Rahimi. Winner's curse? on pace, progress, and empirical rigor. In *ICLR Workshops*, 2018.

Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *ICLR*, 2014.

Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-Shot Learning for Semantic Segmentation. *BMVC*, 2017.

Sharif Shameem. "this is mind blowing. with gpt-3, i built a layout generator where you just describe any layout you want, and it generates the jsx code for you. w h a t", 2020. URL `https://twitter.com/sharifshameem/status/1282676454690451457`. (Accessed: 2022-3-30).

Janelle Shane. Do neural nets dream of electric sheep?, 2018. URL `https://aiweirdness.com/post/171451900302/do-neural-nets-dream-of-electric-sheep`. (Accessed: 2019-8-7).

Vaishaal Shankar, Rebecca Roelofs, Horia Mania, Alex Fang, Benjamin Recht, and Ludwig Schmidt. Evaluating machine accuracy on imagenet. In *ICML*, 2020.

Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, 2019.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *CVPR Workshops*, 2014.

Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Joseph P Simmons, Leif D Nelson, and Uri Simonsohn. False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, 22(11): 1359–1366, 2011.

Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

Josef Sivic, Bryan C Russell, Alexei A Efros, Andrew Zisserman, and William T Freeman. Discovering objects and their location in images. In *ICCV*, 2005.

Arnold WM Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *TPAMI*, 22(12):1349–1380, 2000.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical Networks for Few-shot Learning. In *NeurIPS*, 2017.

Elizabeth S Spelke. Principles of object perception. *Cognitive Science*, 14(1):29–56, 1990.

Courtney J Spoerer, Patrick McClure, and Nikolaus Kriegeskorte. Recurrent convolutional neural networks: a better model of biological object recognition. *Frontiers in Psychology*, 8:1551, 2017.

Matt Strassler. Opera: What went wrong, 2012. URL `https://profmattstrassler.com/articles-and-posts/particle-physics-basics/neutrinos/neutrinos-faster-than-light/opera-what-went-wrong/`. (Accessed: 2022-4-23).

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018.

Richard Sutton. The bitter lesson. *Incomplete Ideas (blog)*, 2019.

Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. *NeurIPS*, 2013.

Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *ICML*, 2019.

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *NeurIPS*, 2020.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017.

Tesla Inc. A tragic loss, 2016. URL `https://www.tesla.com/de_DE/blog/tragic-loss`. (Accessed: 2021-12-04).

Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *ECCV*, 2020.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IROS*, 2017.

Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, 2011.

Anne M Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 1980.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, et al. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *ICLR*, 2020.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. From imagenet to image classification: Contextualizing progress on benchmarks. In *ICML*, 2020.

Alan M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

Ivan Ustyuzhaninov, Claudio Michaelis, Wieland Brendel, and Matthias Bethge. One-shot Texture Segmentation. *arXiv:1807.02654*, 2018.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *CVPR*, 2018.

Joaquin Vanschoren and Serena Yeung. Announcing the neurips 2021 datasets and benchmarks track, 2021. URL `https://neuripsconf.medium.com/announcing-the-neurips-2021-datasets-and-benchmarks-track-644e27c1e66c`. (Accessed: 2022-3-30).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NeurIPS*, 2016.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Kiri L Wagstaff. Machine learning that matters. In *ICML*, 2012.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. arXiv:1804.07461, 2018.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv:1905.00537*, 2019a.

Angelina Wang, Arvind Narayanan, and Olga Russakovsky. Revise: A tool for measuring and mitigating bias in visual datasets. In *ECCV*, 2020a.

Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2020b.

Peng Wang, Xinyu Huang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *TPAMI*, 42(10):2702–2719, 2019b.

Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv:1911.04623*, 2019c.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys*, 53(3):1–34, 2020c.

Zirui Wang, Adams Wei Yu, Orhan Firat, and Yuan Cao. Towards zero-label language learning. *arXiv:2109.09193*, 2021.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

Chris Welty, Praveen Paritosh, and Lora Aroyo. Metrology for ai: From benchmarks to instruments. *arXiv:1911.01875*, 2019.

Davis Wertheimer and Bharath Hariharan. Few-shot learning with localization in realistic settings. In *CVPR*, 2019.

Luke Westaway. Why 'outdated' turing test is no longer the gold standard of ai, 2014. URL `https://www.cnet.com/news/why-outdated-turing-test-is-no-longer-the-gold-standard-of-ai`. (Accessed: 2021-11-14).

Julia K Winkler, Christine Fink, Ferdinand Toberer, Alexander Enk, Teresa Deinlein, Rainer Hofmann-Wellenhof, Luc Thomas, Aimilios Lallas, Andreas Blum, Wilhelm Stolz, et al. Association between surgical skin markings in dermoscopic images and diagnostic performance of a deep learning convolutional neural network for melanoma recognition. *JAMA Dermatology*, 155(10):1135–1141, 2019.

Jeremy M. Wolfe, George A. Alvarez, Ruth Rosenholtz, Yoana I. Kuzmova, and Ashley M. Sherman. Visual search for arbitrary objects in real scenes. *Attention, perception & psychophysics*, 73(6):1650–1671, 2011.

Xiongwei Wu, Doyen Sahoo, and Steven CH Hoi. Meta-rcnn: Meta learning for few-shot object detection. *OpenReview*, 2019a.

Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR*, 2013.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019b.

Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *TPAMI*, 41(9):2251–2265, 2018.

Kai Yuanqing Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. In *ICLR*, 2021.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

Daniel LK Yamins and James J DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, 19(3):356–365, 2016.

Daniel LK Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *PNAS*, 111 (23):8619–8624, 2014.

Fan Yang, Ajinkya Kale, Yury Bubnov, Leon Stein, Qiaosong Wang, Hadi Kiapour, and Robinson Piramuthu. Visual search at ebay. In *KDD*, 2017.

Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13–es, 2006.

Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2020a.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *CoRL*, 2020b.

Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, 8:58443–58469, 2020.

John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLoS Medicine*, 15(11), 2018.

Oliver Zendel, Hassan Abu Alhaija, Rodrigo Benenson, Marius Cordts, Angela Dai, Xavier Puig Fernandez, Andreas Geiger, Niklas Hanselmann, Nicolas Jourdan, Vladlen Koltun, Peter Kontschieder, Alina Kuznetsova, Yubin Kuang, Tsung-Yi Lin, Claudio Michaelis, et al. Robust vision challenge, 2020. URL `http://www.robustvision.net/`. (Accessed: 2022-3-30).

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *arXiv:2004.08955*, 2020.

Yanhao Zhang, Pan Pan, Yun Zheng, Kang Zhao, Yingya Zhang, Xiaofeng Ren, and Rong Jin. Visual search at alibaba. In *KDD*, 2018.

Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *arXiv:2110.04596*, 2021a.

Yuxiao Zhang, Alexander Carballo, Hanting Yang, and Kazuya Takeda. Autonomous driving in adverse weather conditions: A survey. *arXiv:2112.08936*, 2021b.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017.

Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, 2019.

One-Shot Segmentation in Clutter

Claudio Michaelis [1 2] **Matthias Bethge** [1 2 3 4] **Alexander S. Ecker** [1 2 4]

Abstract

We tackle the problem of one-shot segmentation: finding and segmenting a previously unseen object in a cluttered scene based on a single instruction example. We propose a novel dataset, which we call *cluttered Omniglot*. Using a baseline architecture combining a Siamese embedding for detection with a U-net for segmentation we show that increasing levels of clutter make the task progressively harder. Using oracle models with access to various amounts of ground-truth information, we evaluate different aspects of the problem and show that in this kind of visual search task, detection and segmentation are two intertwined problems, the solution to each of which helps solving the other. We therefore introduce *MaskNet*, an improved model that attends to multiple candidate locations, generates segmentation proposals to mask out background clutter and selects among the segmented objects. Our findings suggest that such image recognition models based on an iterative refinement of object detection and foreground segmentation may provide a way to deal with highly cluttered scenes.

1. Introduction

Humans are not only good at learning to recognize novel, unknown objects from a single instruction example (one-shot learning), but can also localize these objects in highly cluttered scenes and segment them from the background.

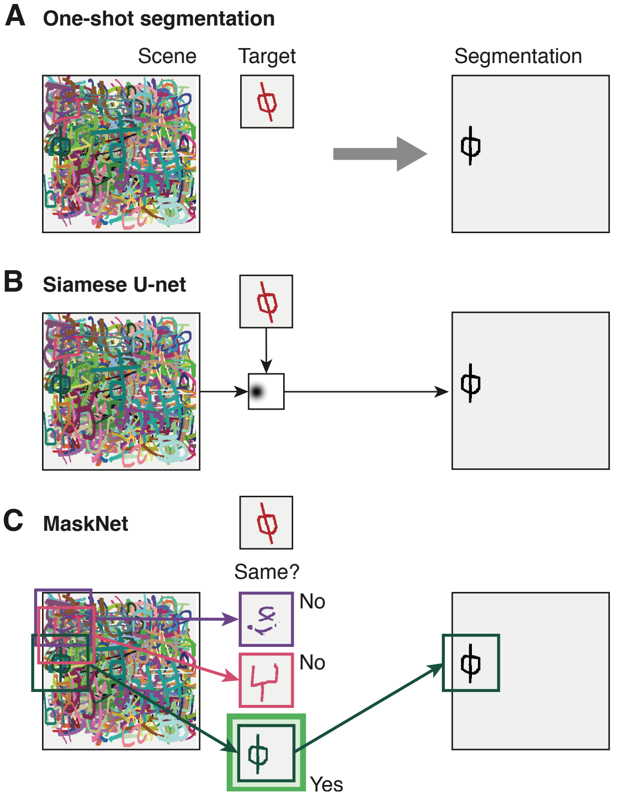


Figure 1. One-Shot Segmentation. **A,** Goal: find a *target* in a cluttered *scene* and produce a pixel-wise segmentation. **B,** Our *Siamese U-net* baseline localizes the target, then segments it. **C,** *MaskNet* generates proposals of segmented instances, masks the background, then computes the best match.

In the computer vision community, one-shot learning has recently received a lot of attention and substantial progress has been made in the context of image classification (Koch et al., 2015; Lake et al., 2015; Vinyals et al., 2016; Bertinetto et al., 2016; Snell et al., 2017; Triantafillou et al., 2017; Shyam et al., 2017). Segmentation, however, is still very much tied to classification, limiting its applicability to datasets with less than a few hundred semantic or object classes (or subsets thereof, e. g. the SceneParse150 benchmark on ADE20k (Zhou et al., 2017)). This stands in contrast to humans who can segment previously unseen objects simply by using contextual information.

In the present paper, we work towards closing this gap by tackling the problem of one-shot segmentation: Given a single instruction example (the *target*) and a cluttered image

[1] Centre for Integrative Neuroscience and Institute for Theoretical Physics, University of Tübingen, Germany [2] Bernstein Centre for Computational Neuroscience, Tübingen, Germany [3] Max Planck Institute for Biological Cybernetics, Tübingen, Germany [4] Center for Neuroscience and Artificial Intelligence, Baylor College of Medicine, Houston, TX, USA. Correspondence to: Alexander Ecker <alexander.ecker@uni-tuebingen.de>.

Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

with many objects (the *scene*), find the target in the scene and produce a pixel-wise segmentation (Fig 1A). This task is harder than the multi-way discrimination task often employed for one-shot learning because it additionally requires (a) localizing the target among a potentially large number of distractors and (b) segmenting the detected object. While a few groups have started working on variants of this task (Caelles et al., 2017; Shaban et al., 2017), no commonly employed benchmark has emerged yet.

Our contributions are as follows:

- We propose a new benchmark dataset: "cluttered Omniglot" (Fig. 1A). It is based on simple components – characters from Omniglot (Lake et al., 2015) – yet turns out to be hard for current state-of-the-art computer vision components. We publish the dataset, the code and our models.[1]
- We present a baseline for one-shot segmentation on cluttered Omniglot. It combines two principled yet simple components: a Siamese network for object detection and a U-net for segmentation (Fig. 1B).
- We identify clutter as a substantial problem for current computer vision systems and investigate it using various oracles – models with access to some ground truth information. Although the statistical complexity of the objects in cluttered Omniglot is low – color alone completely identifies each instance –, the dead leaves environment creates difficulties for both detection and segmentation due to the similar foreground and background statistics.
- We propose to solve this task by a form of object-based attention: we first generate and segment multiple object proposals, then mask out background and finally decide among the "cleaned-up" objects (Fig. 1C). We show that this approach, which we call *MaskNet*, improves both segmentation and localization.

Our paper is structured as follows: We start by describing the cluttered Omniglot dataset (Sec. 2), then explain our Siamese U-net baseline (Sec. 3) and MaskNet, our improved architecture (Sec. 4), as well as the oracles we use (Sec. 5). We then present our experimental results (Sec. 6), discuss related work (Sec. 7) and conclude (Sec. 8).

2. Cluttered Omniglot

Cluttered Omniglot is a visual search task: the goal is to find a previously unseen target character in a cluttered scene and to produce a pixelwise segmentation (Fig. 1A). It is based on the Omniglot dataset (Lake et al., 2015), which we chose for two reasons: First, it is a popular and well-studied dataset for one-shot learning. Second, the statistics of the individual objects in Omniglot are relatively simple. Nevertheless, we show below that cluttered Omniglot presents a serious challenge to convolutional neural networks. Thus, we think of this dataset as the essence of the clutter problem.

Each sample in the dataset consists of three images: a target, a scene and a segmentation map. *Targets* are individual characters from Omniglot, rescaled to 32×32 pixels and colored in a random RGB color. *Scenes* are 96×96 pixel collages of multiple (4–256) randomly drawn Omniglot characters, one of which is the target (Fig. 2). The characters are sequentially "dropped" into the image like dead leaves, occluding any characters previously drawn at the same pixel locations. Each character is placed at a random location, has a random RGB color and is transformed with a random affine transformation of up to 20° rotation, 10° shearing and scaling between 16 and 64 pixels. At the end, a random instance of the target character is added. This instance is always fully visible and not occluded. We specifically avoid occlusion of the target instance, so we do not confound the effect of visual clutter with that of occlusion.

We split the dataset into three splits: training, validation and one-shot. As in the original work on Omniglot (Lake et al., 2015), we use the *background* set for training and validation, while we use the *evaluation* set for testing one-shot performance. For simplicity, we use only the first ten drawers in each alphabet for the training set and the other ten drawers for the validation and one-shot sets.

The difficulty of this task depends on the number of distractors (Wolfe, 1998). We show below (Section 6.1) that our baseline scores a close-to-perfect Intersection over Union (IoU) for the easiest version with just four distractors, similar to the accuracies of high-performing architectures designed for one-shot discrimination on Omniglot (Koch et al., 2015; Vinyals et al., 2016; Snell et al., 2017; Triantafillou et al., 2017; Shyam et al., 2017). In contrast, performance drops below 40% IoU for the hardest version with 256 distractors.

For each difficulty level, we generate a training set consisting of 2 million samples and validation and one-shot sets consisting of 10,000 samples each. Note that the entire dataset is generated using a total of 9640 (6590) character instances for the training (one-shot) set.

3. Baseline: Siamese U-net

Intuitively, the one-shot segmentation task can be broken down into two steps: detect the target in the scene and segment it. We implement a baseline that performs the detection part with a Siamese net applied in sliding windows over the scene to produce a heat map of candidate locations (Fig. 3A). The segmentation mask is then generated by a

[1] https://github.com/michaelisc/cluttered-omniglot

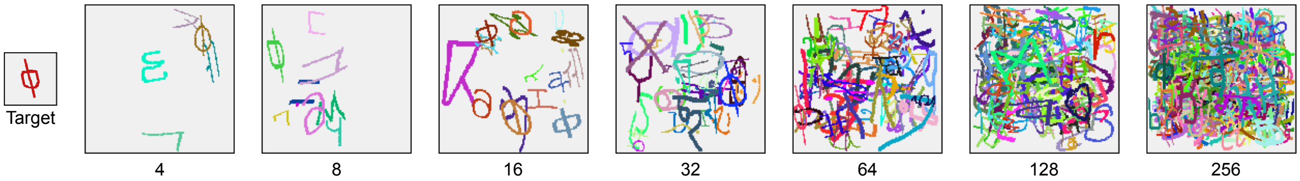


Figure 2. Multiple *scenes* form *cluttered Omniglot* with a common *target* and varying amounts of clutter defined by the numbers of characters in each scene.

deconvolutional net with skip connections from the encoder.

3.1. Encoder

The encoder is inspired by Siamese networks. It consists of two parallel fully convolutional neural networks that process the target ($32 \times 32 \times 3$) and the scene image ($96 \times 96 \times 3$), respectively (Fig. 3A). All convolutions use 3×3 kernels with "same" padding, followed by layer normalization (Ba et al., 2016) and ReLUs. An exception is made in the last two layers, which use 2×2 and 1×1 kernels respectively (the size of the feature maps of the target encoder in these layers) (Fig. 3C). Before each but the first convolution, the image is downsampled by a factor of two using average pooling. This architecture produces an embedding of the target in form of a 384-dimensional vector (1×1 spatially). The scene image is processed analogously. To retain a higher resolution in the last layer, we do not use downsampling in the last two layers of the scene encoder. Instead we us a dilation factor of 2 for the convolutions in the second-to-last layer. This results in a 12×12 pixel encoding with – as for the target – 384 feature maps.

Although the encoder is inspired by Siamese networks, we found in initial experiments that untying the weights improves performance and therefore do not use weight sharing between the two paths (see also Bertinetto et al., 2016). This result could potentially be attributed to the differing statistics of the clean target and the cluttered scene image.

3.2. Target matching

To get an estimate of the target's location in the scene, we compute the cosine similarity in the embedding space given by the encoder. We do so by taking the pixelwise inner product of the scene embedding with that of the target (Fig. 3C), which is implemented by a 1×1 convolution using the target embedding as the filter. This step can be thought of as applying a Siamese network in sliding windows over the scene image (with a stride of 8, the stride of the final layer of the scene encoder). The output is a 12×12 heatmap, which can be seen as a (subsampled) pixel-level likelihood that the target is at a given location within the scene.

This heatmap does not contain any information about what the target is. To inform the decoder about the target that should be segmented, we compute the outer tensor product of the heatmap with the target embedding. Thus, the final output of the matching step is a $12 \times 12 \times 384$ tensor, which encodes at each location the direction of the target in embedding space, weighted by how likely the encoder considers the target to be at that location. As all other layers, this output is normalized using layer normalization.

3.3. Decoder

The segmentation part of our baseline model is inspired by the U-net architecture (Ronneberger et al., 2015). The decoder is essentially a mirror image of the encoder: six convolutional layers with 3×3 kernels and "same" padding, followed by layer normalization, ReLU and – for the third, fourth and fifth layer – nearest neighbor upsampling by a factor of two to incrementally increase the image size to the original 96×96 pixels (Fig. 3C). The input to each convolutional layer in the decoder is the concatenation of the previous layer's output and the output of the corresponding layer in the encoder (skip connections). The final layer of the decoder outputs two feature maps, which are combined into a segmentation map by taking the pixelwise softmax.

3.4. Training

During training, we minimize the binary cross-entropy between the ground truth segmentation and the network's prediction. The cross-entropy is computed pixelwise and averaged across all pixels. The weights are initialized randomly from a Gaussian distribution following the MSRA initialization scheme (He et al., 2015). We regularize the weights using L_2 weight decay with a factor of 10^{-9}. We train the network for 20 epochs using Adam (Kingma & Ba, 2014) with a batch size of 250 and an initial learning rate of 5×10^{-4}. After 10, 15 and 17 epochs, we divide the learning rate by 2.

3.5. Evaluation

We evaluated the baseline model using intersection over union (IoU). Therefore the generated segmentation maps are binarized using a threshold or 0.3, which was determined to be optimal across models and datasets.

4. MaskNet: Segment first, decide later

MaskNet (Fig. 3B) adds two additional processing stages to the baseline. Instead of generating the segmentation in a single pass through the U-net, we let the decoder attend to different locations. We branch off at the target matching stage and generate multiple object proposals with associated instance segmentations. We then decide which of these proposals is the best match. This last stage reduces to the one-shot multi-way discrimination task for image classification, and we solve it using a Siamese net.

4.1. Proposal network

We modify our Siamese U-net to turn it into a targeted proposal network (Fig.3B+C). Its output is a set of segmentation proposals (96×96 pixels). To this end, we modify the target matching step: instead of computing the heatmap by an inner product of target and scene embeddings, we simply set it to a one-hot map encoding a single location (Fig.3C, orange block). We then use the simplest possible strategy for selecting candidate locations: sweeping all possible locations, thus generating 144 proposals (Fig.3B). While there are certainly more elaborate ways of generating proposals, we opt for simplicity over efficiency. Similar to the target matching step in the baseline network, these one-hot heatmaps are multiplied with the target embedding and normalized using layer normalization. Thus, for each proposal, the decoder is seeded by an embedding of the target confined to a single pixel within the 12×12 spatial grid and generates a segmentation mask for the target at this location (or background if the target is not present).

4.2. Decision stage

The decision stage takes multiple object proposals as input and uses a Siamese network to pick the one that most closely resembles the target (Fig. 3B). This step is essentially a 144-way one-shot discrimination task. The key ingredient here is the input: instead of just taking crops from the scene, we use the generated segmentations to mask out background clutter and perform the discrimination on “clean” objects (Fig. 3B & Fig. 1C). To do so, we binarize the segmentation proposals using a threshold of 0.3 and extend them to RGB colors by simply coloring them white. For each proposal, we compute the center of mass of the segmentation mask and extract a 32×32 pixel crop centered on this point. We found this solution using the mask directly to perform slightly better then applying it to the image. These crops are then fed into an encoder with the same architecture as the one used for the target (i. e. outputs a 384-dimensional embedding). As in Siamese networks (Koch et al., 2015), we use the sigmoid of a weighted sum of the L1 distance between two embeddings as a similarity measure. The full segmentation map corresponding to the crop that is most similar to the target is the final output.

4.3. Training

We train proposal network and discriminator separately, by initializing the weights (where possible) from the Siamese U-net baseline and then fine-tuning (Sec. 3.4). All other weights are initialized randomly as for the baseline. We use the same optimizer and regularization as before. We train for five epochs, dividing the learning rate by two after two, three and four epochs, respectively.

To train the proposal network, we generate eight proposals for each training sample: four positive ones as above and four negative ones, which are drawn from random locations. We then fine-tune encoder and decoder using the same pixelwise cross-entropy loss as above using the ground truth segmentation for the positive samples and “background” as the label for the negative ones. The initial learning rate is set to 5×10^{-5} and the batch size is 50.

To train the discriminator, we fix the target encoder, train the encoder for the segmented patches by initializing with the weights of the target encoder and fine-tuning, and train the weights for the weighted L_1 distance. For each training sample, we generate four segmentation proposals: one centered at one of the four locations around the center of mass of the target and three at other random positions. We minimize the binary cross-entropy of the same/different task for each proposal. The initial learning rate is set to 2.5×10^{-4} and the batch size is 250.

4.4. Evaluation

To evaluate MaskNet, we use intersection over union (IoU) as for the baseline. As before, we apply a threshold of 0.3 to the predicted segmentation mask. In addition, we evaluate the localization accuracy of the network independent of the quality of the generated segmentation masks. To do so, we use the center of mass of the chosen segmentation proposal as the prediction of the target's location. We count all predictions that are within five pixels of the ground truth location (also center of mass) as correct and report localization accuracy in percent correct.

5. Oracles

We evaluate two oracles that have access to ground truth segmentation masks of all characters in the scene. Being able to define such oracles is a useful feature of cluttered Omniglot, which allows us to test the quality of individual model components.

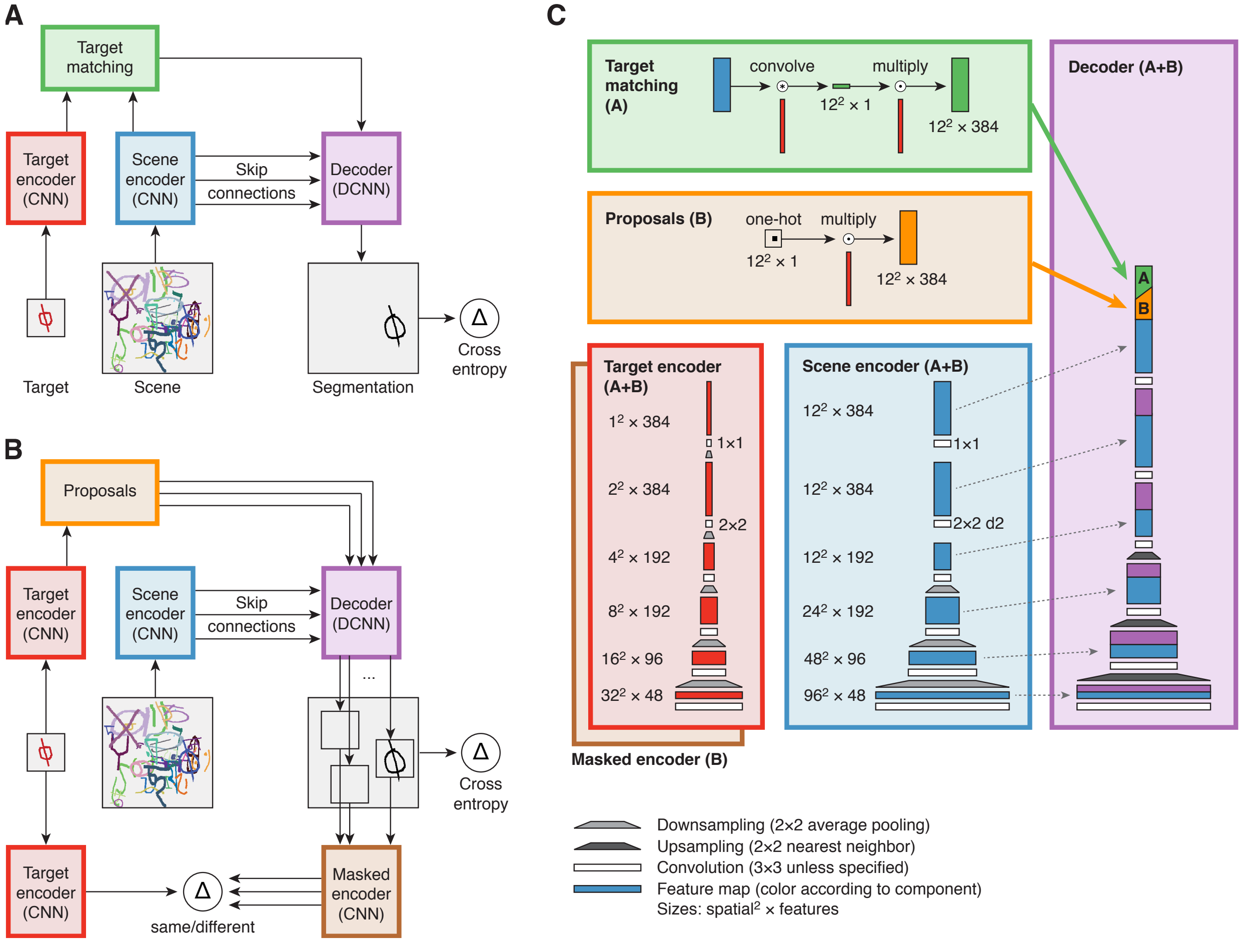


Figure 3. Architectures and details. **A,** Siamese U-net baseline (Section 3). **B,** MaskNet (Section 4). **C,** Close-up of the individual components, showing architecture details.

5.1. Pre-segmented discriminator

The *pre-segmented discriminator* operates on individual characters that have been pre-segmented and cropped to the same size as the target. Specifically, we use the fact that the characters are uniformly colored to segment each character and extract a 32×32 pixel crop centered on its center of mass. The task of this oracle is the same as for the decision step of MaskNet (Sec. 4.2) and can be reduced to the widely used one-shot multi-way discrimination, hence the name *discriminator*. We implement it by a Siamese network using the same encoder as before (Sec. 3.1) comparing the generated embeddings with a weighted L_1 distance, followed by a sigmoid (Koch et al., 2015). The pre-segmented discriminator lets us assess the additional difficulty (if any) introduced by (a) the random affine transformations in cluttered Omniglot and (b) the potentially large number of candidate characters to decide among.

5.2. Cluttered discriminator

The *cluttered discriminator* does not pre-segment characters. Instead it takes the same crops as the pre-segmented discriminator, but keeps the cluttered background intact. The rest is identical to the pre-segmented discriminator. Thus, the cluttered discriminator performs the one-shot multi-way discrimination on cluttered crops. By comparing its performance to that of the pre-segmented version, we can directly assess the effect of clutter on discrimination.

5.3. Training

We train both discriminators by minimizing the binary cross-entropy in the same/different task. In each training step, four crops are sampled: one containing the target and three randomly selected ones. Each crop is compared with the target and the average cross-entropy is computed. Initialization, regularization and optimization are done in the same way as for the baseline (Sec. 3.4). A batch size of 250 and an initial learning rate of 5×10^{-4} are chosen. Like the baseline, the

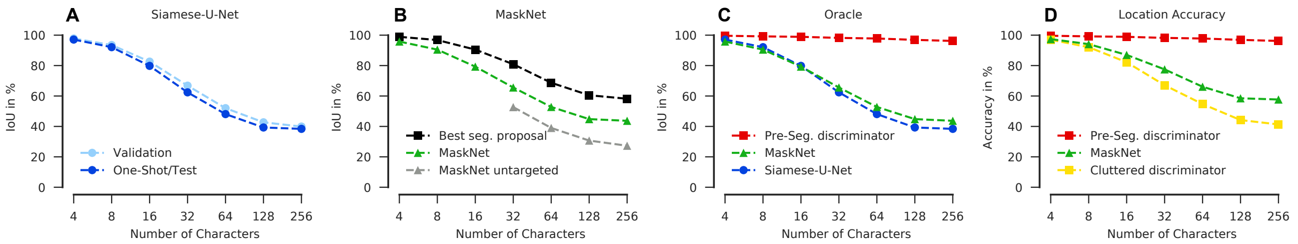


Figure 4. Performance of various model architectures and oracles on cluttered Omniglot. Performance is measured as intersection over union (IoU) for segmentation (A–C) or localization accuracy (D); higher is better. All results (except A) are measured on the one-shot sets. **A,** IoU of the Siamese-U-Net on validation (light blue) and one-shot set (dark blue). **B,** MaskNet with targeted (green) and un-targeted proposals (grey) and the best segmentations generated by the proposal network (black). **C,** Comparison of Siamese-U-Net (blue), MaskNet (green) and an oracle: the pre-segmented discriminator (red), which has access to ground truth locations and segmentation masks of all characters (but not to class labels). **D,** Localization accuracy of MaskNet (green) in comparison to the cluttered (yellow) and the pre-segmented discriminator (red).

discriminators are trained for 20 epochs and the learning rate is divided by 2 after epochs 10, 15 and 17.

5.4. Evaluation

We evaluate the pre-segmented discriminator using the same two metrics used for MaskNet: IoU and localization accuracy. To evaluate IoU, we use the ground truth segmentations associated with the best-matching crop. Due to the access to ground truth segmentations, IoU is equivalent to the percentage of correct decisions in the discrimination task. To evaluate localization accuracy, we take the same measure as for MaskNet: The Euclidean distance between the center of each crop and the true location of the target thresholded at 5 pixels. For the cluttered discriminator, we evaluate only localization accuracy.

6. Results

We used the same encoder and decoder architectures for all experiments. Both consist of six convolutional layers interleaved with pooling, dilation or upsampling operations (see Fig. 3C and Sec. 3.1). All comparisons between architectures are therefore independent of the expressiveness of encoder and decoder, but rely only on the different approaches to segmentation and detection. All reported results are evaluated on the one-shot set unless specified otherwise.

6.1. Baseline

We start by characterizing the difficulty of the one-shot segmentation task on cluttered Omniglot by evaluating the performance of our baseline model (Section 3) on both, the one-shot and the validation set across all difficulty levels.

We first consider the results on the validation set (Fig. 4A, light blue). The validation set contains characters seen during training, but drawn by a different set of drawers (see Section 2). For a small number of distractors, the network performs well – as expected, because the characters are mostly isolated within the scene. Performance is above 90% IoU, similar to discrimination performance in one-shot five-way discrimination on regular Omniglot (Koch et al., 2015; Vinyals et al., 2016; Snell et al., 2017; Triantafillou et al., 2017; Shyam et al., 2017). However, performance drops substantially with increasing number of distractors ($< 40\%$ for 256 distractors).

Table 1. One-shot segmentation accuracy (IoU in %) across different amounts of clutter (number of characters per image).

MODEL	4	8	16	32	64	128	256
PATTERN MATCHING	62.2	50.4	41.7	36.9	32.6	29.0	28.6
P-SEG. DISCRIMINATOR	99.6	99.2	98.9	98.2	97.8	96.9	96.2
BEST SEG. PROPOSAL	98.9	96.8	90.5	80.9	68.7	60.5	58.2
SIAMESE U-NET	**97.1**	**92.1**	**79.8**	62.4	48.1	39.3	38.4
MaskNet	95.8	90.5	79.3	**65.6**	**52.8**	**44.8**	**43.7**
MASKN. UNTARGETED	-	-	-	52.7	39.0	30.7	27.3

On the one-shot set – that is, characters from alphabets not seen during training – performance is on average only 3% worse than validation performance (Fig. 4A, blue), showing that the network has indeed learned the right metric to identify previously unseen letters and segment them.

6.2. Clutter reduces performance more than the number of comparisons

The performance drop of our baseline model with increasing number of distractors could have two reasons. First, the scenes are highly cluttered, which may cause problems for the detection of the target. Second, the large number of comparisons may simply increase the probability of making a mistake by chance (n-way discrimination with large n). To understand the influence of these factors, we constructed two oracles, which both have access to the ground truth locations of all characters in the scene (Sec. 5). Both models

Table 2. One-shot localization accuracy (in %) across different amounts of clutter (number of characters per image).

Model	4	8	16	32	64	128	256
P-Seg. discriminator	99.6	99.2	98.9	98.2	97.8	96.9	96.2
Clutt. discriminator	97.0	92.1	82.2	67.1	54.7	44.2	41.3
MaskNet	**97.4**	**94.1**	**87.0**	**77.5**	**66.1**	**58.5**	**57.7**

extract crops centered at the location of each character in the scene and perform a discrimination task between these crops and the target.

The pre-segmented discriminator has access not only to the ground truth location but also the segmentation mask of each character, allowing it to pre-segment all crops. The resulting task is essentially the classical one-shot n-way discrimination task. The only difference is that it is a bit easier since many characters in the background are highly occluded, whereas the target is always unoccluded. Remarkably, the performance of the pre-segmented discriminator remains above 95% IoU even for the most cluttered scenes with 256 characters (Fig. 4C+D, red), demonstrating that our encoder can solve the task in an uncluttered environment.

The cluttered discriminator has access to only the ground truth locations. It cannot segment the characters and has to perform the n-way discrimination on cluttered crops. In contrast to the pre-segmented discriminatior its performance takes a substantial hit with increased clutter (Fig. 4D, yellow). Thus we conclude that the difficulty of cluttered Omniglot arises due to clutter rather than the potentially large number of candidate characters in the scene.

6.3. Template matching is not sufficient

A lot of work on one-shot learning has used Omniglot, but we are not aware of any work evaluating simple approaches like template matching. As a sanity check, we implemented a template matching procedure for our task based on the pre-segmented discriminator.[2] Accuracy ranged from 62% for 4 characters to 29% for 256 characters (Table 1).[3] Despite the highly simplified setting with oracle information available, template matching performs not only worse than the pre-segmented discriminator ($99 - 96\%$), but even worse than our baseline on the full task ($97 - 38\%$). Thus, template matching is not a viable solution for (cluttered) Omniglot.

6.4. Background masking improves performance

Motivated by the superb discrimination performance on pre-segmented objects, we developed MaskNet, a novel model that operates in three steps (Sec. 4). First, we generate a number of object proposals. Next, we generate corresponding object segmentations which mask out the background. In the last step, we perform discrimination on these segmented objects to decide which one to pick. This model outperforms the baseline (Fig. 4B+C, green line), suggesting that segmenting objects (and masking out background) before classifying them is beneficial when processing highly cluttered scenes. Nevertheless, there is still a large margin to the performance of the pre-segmented oracle. We investigate the reasons for this margin below.

6.5. Quality of segmentation limits performance

A crucial feature of MaskNet (and perhaps its main weakness) is that the final discriminator can only be as good as the segmentations it receives as input. We therefore evaluate the quality of these segmentations. To this end, we evaluate the maximal IoU among all proposals, which is equivalent to assuming a perfect discriminator that always picks the correct character. We find that indeed the instance segmentations of the proposals appear to be a limiting factor: for the most cluttered scenes the proposal with the highest IoU achieves only around 60% on average (Fig. 4B, black).

6.6. Targeted segmentations improve performance

Next, we test whether it is necessary to seed the decoder with an embedding of the target, instead of just seeding it with a location and segment the most salient character at that location. To this end, we remove the target multiplication step from MaskNet's proposal network and simply seed the decoder with the spatial one-hot encoding (Section 4.1). Using this non-targeted proposal network instead of the targeted one reduces performance (Fig. 4B, grey), showing that it is important to supply the decoder with information what to segment.

6.7. Performing segmentation improves localization

So far, we have focused our evaluation of MaskNet's performance on segmentation. Interestingly, though, segmenting objects also helps if we are interested only in localizing the target rather than segmenting it. To provide evidence for this claim, we compare the localization performance of MaskNet to that of the cluttered discriminator. For the cluttered discriminator, we simply use the location of the crop it chooses as the prediction for the target's location. For MaskNet, we use the center of mass of its predicted segmentation mask. We then compute the localization accuracy (Sec. 4.4) of these predictions to the ground truth center of mass of the target. Indeed, MaskNet predicts the location of the target more accurately than the cluttered discriminator (Fig. 4D and Table. 2), showing that segmenting objects to mask out background clutter improves localization.

[2] We generated 9317 transformed versions of the target (11 rotations, 7 shearing angles, 11x11 x/y scales), convolved them with each segmented, binarized character and picked the best match.

[3] For comparison: on the standard 5-way one-shot task on Omniglot, we achieved 84% accuracy using template matching.

7. Related Work

7.1. One-shot discrimination

One-shot learning has been explored mostly in the context of multi-way discrimination for image classification. Lake et al. (2015) developed the Omniglot dataset for this purpose and approach it using a generative model of stroke patterns. Most competing approaches learn an embedding to compute a similarity metric (Koch et al., 2015; Vinyals et al., 2016; Snell et al., 2017; Triantafillou et al., 2017). Bertinetto et al. (2016) train a meta network that predicts the weights of a discriminator in a single feedforward step. Another approach compares image parts in an iterative fashion (Shyam et al., 2017).

7.2. Semantic/instance segmentation

Most recent approaches to segmentation use an encoder/decoder architecture (Noh et al., 2015; Badrinarayanan et al., 2017). The encoders are usually high-performing architectures for image classification [e. g. AlexNet (Krizhevsky et al., 2012), VGG (Simonyan & Zisserman, 2015), ResNet (He et al., 2016)]. The main differences lie in the decoder design. Where early works converted high-level representations into pixelwise labels using upsampling in combination with linear transformation (Long et al., 2015) or conditional random fields (Chen et al., 2014; 2018), recent approaches rely on more complex decoders [DeconvNet (Noh et al., 2015), SegNet (Badrinarayanan et al., 2017), RefineNet (Lin et al., 2017)] and introduce skip connections from the encoder. The U-net architecture (Ronneberger et al., 2015), which uses skip connections is a particularly simple and elegant general-purpose architecture for dense labeling and image-to-image problems (e. g. Isola et al., 2016).

More recent work focuses on multi-scale pooling (Zhao et al., 2017) and dilated convolutions (Chen et al., 2017). These architectures improve performance, but simplify the decoders, relying more on upsampling. While this approach works well on datasets such as MS-COCO, it renders them infeasible for segmenting on Omniglot, where characters have fine detail at the pixel level.

Our proposal network is inspired by Mask R-CNN (He et al., 2017), which achieved state-of-the-art performance on MS-COCO by splitting object detection and instance segmentation into two consecutive steps. Similarly, our class-agnostic segmentation is inspired by the work of Hong et al. (2015) and Mask R-CNN (He et al., 2017). Also related is work on class-agnostic segmentation using extreme point annotations (Maninis et al., 2017; Papadopoulos et al., 2017): while these works inform the segmentation by clicks in the image, our architecture seeds the decoder with a location information at the embedding layer.

7.3. One-shot segmentation

One-shot segmentation has emerged only recently. Caelles et al. (2017) tackle the problem of segmenting an unseen object in a video based on a single (or a few) initial labeled frame(s). The work by Shaban et al. (2017) is very similar to our approach, except that they use logistic regression with a large stride and upsampling for the decoder and tackle Pascal VOC (Everingham et al., 2012).

7.4. Other related problems

Co-segmentation (Faktor & Irani, 2013; Quan et al., 2016; Sharma, 2017) is somewhat related to one-shot segmentation, as the common object in multiple images has to be segmented. However, objects are typically quite salient (otherwise the problem is not well defined). We can think of cluttered Omniglot as an asymmetric co-segmentation problem with one object-centered and one scene image.

Apparel recognition (Hadi Kiapour et al., 2015; Zhao et al., 2016; Cheng et al., 2017) and particular object retrieval (Razavian et al., 2014; Tolias et al., 2016; Li et al., 2017; Siméoni et al., 2017) are related in the sense that the goal is to find objects specified by one image in other images. However, both problems are primarily about image retrieval rather than segmentation of objects within these images. One exception is the work of Zhao et al. (2016) in which co-segmentation is performed on pieces of clothing.

8. Conclusions

We explored one-shot segmentation in cluttered Omniglot and found increasing clutter to quickly diminish performance even though characters can be easily identified by color. Thus clutter is a serious problem for current state-of-the-art CNN architectures. As a first step towards solving this problem, we showed that segmenting objects first improves detection when scenes are cluttered. We aimed for a proof of principle and thus used the simplest model possible, which performs only one iteration of segmentation and then decides directly based upon this first segmentation. Fully recurrent architectures that iteratively refine detection and segmentation by cycling through this process multiple times could lead to even larger performance gains.

As we focus on the role of clutter, we specifically designed cluttered Omniglot to have relatively simple object statistics but various levels of clutter. An interesting avenue for future work would be to specifically investigate cluttered image regions in real-world datasets such as Pascal VOC, MS-COCO or ADE20k. Both, the task and our MaskNet architecture should be directly applicable to these datatsets, for instance by searching for unseen object categories in natural scenes could be done by replacing our encoder by a state-of-the-art ImageNet classifier.

Acknowledgements

This work was supported by the German Research Foundation (DFG) through Collaborative Research Center (CRC 1233) "Robust Vision" and DFG grant EC 479/1-1, and by the Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DoI/IBC) contract number D16PC00003. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/IBC, or the U.S. Government.

References

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer Normalization. arXiv:1607.06450 [cs, stat], 2016. URL http://arxiv.org/abs/1607.06450.

Badrinarayanan, V., Kendall, A., and Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. TPAMI, 39(12):2481–2495, 2017. doi: 10.1109/TPAMI.2016.2644615.

Bertinetto, L., Henriques, J. F., Valmadre, J., Torr, P., and Vedaldi, A. Learning feed-forward one-shot learners. In NIPS, pp. 523–531. 2016.

Caelles, S., Maninis, K.-K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., and Van Gool, L. One-shot video object segmentation. In CVPR, 2017.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. arXiv:1412.7062 [cs], 2014. URL http://arxiv.org/abs/1412.7062.

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation. arXiv:1706.05587 [cs], 2017. URL http://arxiv.org/abs/1706.05587.

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. TPAMI, 2018. doi: 10.1109/TPAMI.2017.2699184.

Cheng, Z.-Q., Wu, X., Liu, Y., and Hua, X.-S. Video2shop: Exact Matching Clothes in Videos to Online Shopping Images. In CVPR, pp. 4048–4056, 2017.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012), 2012. URL http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

Faktor, A. and Irani, M. Co-segmentation by Composition. In ICCV, pp. 1297–1304, 2013. URL http://ieeexplore.ieee.org/document/6751271/.

Hadi Kiapour, M., Han, X., Lazebnik, S., Berg, A. C., and Berg, T. L. Where to buy it: Matching street clothing photos in online shops. In ICCV, pp. 3343–3351, 2015. URL http://www.cv-foundation.org/openaccess/content_iccv_2015/html/Kiapour_Where_to_Buy_ICCV_2015_paper.html.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In ICCV, pp. 1026–1034, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In CVPR, pp. 770–778, 2016.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask R-CNN. In ICCV, pp. 2980–2988, October 2017. doi: 10.1109/ICCV.2017.322.

Hong, S., Noh, H., and Han, B. Decoupled Deep Neural Network for Semi-supervised Semantic Segmentation. In NIPS, pp. 1495–1503. 2015.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. Image-to-Image Translation with Conditional Adversarial Networks. arXiv:1611.07004 [cs], 2016. URL http://arxiv.org/abs/1611.07004.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs], 2014. URL http://arxiv.org/abs/1412.6980.

Koch, G., Zemel, R., and Salakhutdinov, R. Siamese Neural Networks for One-shot Image Recognition - oneshot1.pdf. ICML, 2015.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In NIPS, pp. 1097–1105, 2012.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. Science, 350(6266):1332–1338, 2015. URL http://science.sciencemag.org/content/350/6266/1332.

Li, W., Wang, L., Li, W., Agustsson, E., Berent, J., Gupta, A., Sukthankar, R., and Van Gool, L. WebVision Challenge: Visual Learning and Understanding

With Web Data. arXiv:1705.05640 [cs], 2017. URL http://arxiv.org/abs/1705.05640.

Lin, G., Milan, A., Shen, C., and Reid, I. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In CVPR, 2017.

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In CVPR, pp. 3431–3440, 2015.

Maninis, K.-K., Caelles, S., Pont-Tuset, J., and Van Gool, L. Deep Extreme Cut: From Extreme Points to Object Segmentation. arXiv:1711.09081 [cs], 2017. URL http://arxiv.org/abs/1711.09081.

Noh, H., Hong, S., and Han, B. Learning deconvolution network for semantic segmentation. In ICCV, pp. 1520–1528, 2015.

Papadopoulos, D. P., Uijlings, J. R., Keller, F., and Ferrari, V. Extreme clicking for efficient object annotation. In ICCV, 2017.

Quan, R., Han, J., Zhang, D., and Nie, F. Object co-segmentation via graph optimized-flexible manifold ranking. In CVPR, pp. 687–695, 2016.

Razavian, A. S., Azizpour, H., Sullivan, J., and Carlsson, S. CNN features off-the-shelf: an astounding baseline for recognition. In CVPR Workshops, pp. 512–519, 2014.

Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Medical Image Computing and Computer-Assisted Intervention, pp. 234–241. Springer, 2015. URL https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28.

Shaban, A., Bansal, S., Liu, Z., Essa, I., and Boots, B. One-Shot Learning for Semantic Segmentation. BMVC, 2017.

Sharma, A. One Shot Joint Colocalization and Cosegmentation. arXiv:1705.06000 [cs], 2017. URL http://arxiv.org/abs/1705.06000.

Shyam, P., Gupta, S., and Dukkipati, A. Attentive Recurrent Comparators. arXiv:1703.00767 [cs], 2017. URL http://arxiv.org/abs/1703.00767.

Siméoni, O., Iscen, A., Tolias, G., Avrithis, Y., and Chum, O. Unsupervised deep object discovery for instance recognition. arXiv:1709.04725 [cs], 2017. URL http://arxiv.org/abs/1709.04725.

Simonyan, K. and Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. ICLR, 2015. URL http://arxiv.org/abs/1409.1556.

Snell, J., Swersky, K., and Zemel, R. Prototypical Networks for Few-shot Learning. In NIPS, pp. 4080–4090. 2017.

Tolias, G., Sicre, R., and Jégou, H. Particular object retrieval with integral max-pooling of CNN activations. ICLR, 2016. URL http://arxiv.org/abs/1511.05879.

Triantafillou, E., Zemel, R., and Urtasun, R. Few-Shot Learning Through an Information Retrieval Lens. In NIPS, pp. 2252–2262. 2017.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., and others. Matching networks for one shot learning. In NIPS, pp. 3630–3638, 2016.

Wolfe, J. M. Visual search. Attention, 1:13–73, 1998.

Zhao, B., Wu, X., Peng, Q., and Yan, S. Clothing Cosegmentation for Shopping Images With Cluttered Background. Transactions on Multimedia, 18(6):1111–1123, 2016. URL http://ieeexplore.ieee.org/document/7423747/.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. Pyramid scene parsing network. In CVPR, pp. 2881–2890, 2017.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. Scene parsing through ADE20k dataset. In CVPR, 2017.

One-Shot Instance Segmentation

Claudio Michaelis Ivan Ustyuzhaninov Matthias Bethge Alexander S. Ecker
University of Tübingen
claudio.michaelis@uni-tuebingen.de

Abstract

We tackle the problem of one-shot instance segmentation: Given an example image of a novel, previously unknown object category (the *reference*), find and segment all objects of this category within a complex scene (the *query image*). To address this challenging new task, we propose Siamese Mask R-CNN. It extends Mask R-CNN by a Siamese backbone encoding both reference image and scene, allowing it to target detection and segmentation towards the reference category. We demonstrate empirical results on MS-COCO highlighting challenges of the one-shot setting: while transferring knowledge about instance segmentation to novel object categories works very well, targeting the detection network towards the reference category appears to be more difficult. Our work provides a first strong baseline for one-shot instance segmentation and will hopefully inspire further research into more powerful and flexible scene analysis algorithms. Code is available at: https://github.com/bethgelab/siamese-mask-rcnn

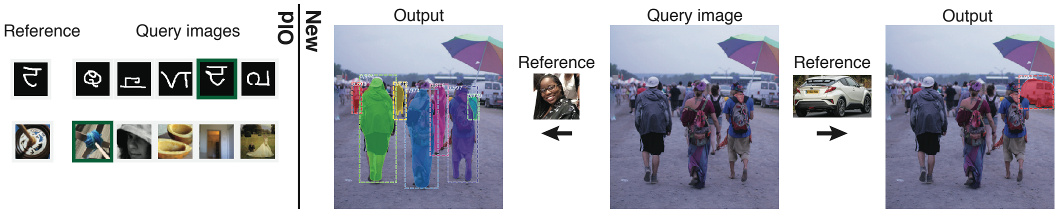


Figure 1: **Left:** Classical one-shot learning tasks are phrased as multi-class discrimination on datasets such as Omniglot and *mini*Imagenet. **Right:** We propose one-shot instance segmentation on MS-COCO. The bounding boxes and instance masks are outputs of our model.

1 Introduction

Humans do not only excel at acquiring novel concepts from a small number of training examples (*few-shot learning*), but can also readily point to such objects (*object detection*) and draw their outlines (*instance segmentation*). Conversely strong machine vision algorithms exist which can detect and segment a limited number of object categories in complex scenes [46, 32, 21]. However in contrast to humans they are unable to incorporate new object concepts for which only a small number of training examples are provided. Enabling these object detection and segmentation systems to perform few-shot learning would be extremely useful for many real-world applications for which no large-scale annotated datasets like MS-COCO [33] or OpenImages [26] exist. Examples include autonomous agents such as household, service or manufacturing robots, or detecting objects in images collected in scientific settings (e. g. medical imaging or satellite images in geosciences).

Computer vision has made substantial progress in few-shot learning in the last years [27, 57, 16, 52, 35]. However, the field has focused on image classification in a discriminative setting, using

Preprint. Under review.

datasets such as Omniglot [27] and MiniImagenet [62] (see Figure 1, left). As a consequence, these approaches are limited to rather simple object-centered images and cannot trivially handle object detection.

In this paper, we combine few-shot learning and instance segmentation in one task: We learn to detect and segment arbitrary objects in complex real-world scenes based on a single visual example (Figure 1, right). That is, we want our system to be able to find people and cars even though it has been provided with only one (or a few) labeled examples for each of those object categories.

To evaluate the success of such a system, we formulate the task of one-shot instance segmentation: Given a scene image and a previously unknown object category defined by a single reference instance, generate a bounding box and a segmentation mask for every instance of that category in the image. This task can be seen as an example-based version of the typical instance segmentation setup and is closely related to the everyday problem of visual search which has been studied extensively in human perception [54, 64].

We show that a new model, *Siamese Mask R-CNN*, which incorporates ideas from metric learning (Siamese networks [25]) into Mask R-CNN [21], a state-of-the-art object detection and segmentation system (Figure 2), can learn this task and acquire a similarity metric that allows it to generalize to previously unknown object categories.

Our main contributions are:

- We introduce one-shot instance segmentation, a novel one-shot task, requiring object detection and instance segmentation based on a single visual example.
- We present *Siamese Mask R-CNN*, a system capable of performing one-shot instance segmentation.
- We establish an evaluation protocol for the task and evaluate our model on MS-COCO.
- We show that, for our model, targeting the detection towards the reference category is the main challenge, while segmenting the correctly identified objects works well.

2 Background

Object detection and instance segmentation. In computer vision, object detection is the task of localizing and classifying individual objects in a scene [14]. It is usually formalized as: Given an image (*query image*), localize all objects from a fixed set of categories and draw a bounding box around each of them. Current state-of-the-art models use a convolutional neural network (the *backbone*) to extract features from the query image and subsequently classify the detected objects into one of the n categories (or background). Most models either directly use the backbone features to predict object locations and categories (*single stage*) [34, 44–46, 32] or first generate a set of class-agnostic object proposals which are subsequently classified (*two stage*) [18, 17, 49, 21].

Segmentation tasks require labeling all pixels belonging to a certain semantic category (*semantic segmentation*) or object instance (*instance segmentation*). While both tasks seem closely related, they in fact require quite different approaches: Semantic segmentation models perform pixel-wise classification and are usually implemented using fully convolutional architectures [36, 40, 50, 68, 8]. In contrast, instance segmentation is more closely related to object detection, as it requires identifying individual object instances [20, 10, 41, 47, 21]. It therefore inherits the difficulties of object detection, which make it a significantly harder task than semantic segmentation. Consequently, the current state-of-the-art instance segmentation model (*Mask R-CNN*) [21] is an extension of a successful object detection model (*Faster R-CNN*) [49].

Few-shot learning The goal of few-shot learning is to find models which can generalize to novel categories from few labeled examples [30, 27]. This capability is usually evaluated through a number of *episodes*. Each episode consists of a few examples from novel categories (the *support set*) and a small test set of images from the same categories (the *query set*). When the support set contains k examples from n categories, the problem is usually referred to as an *n-way, k-shot* learning problem. In the extreme case when only a single example per category is given, this is referred to as one-shot learning.

There are two main approaches to solve this task: either train a model to learn a metric, based on which examples from novel categories can be classified (*metric learning*) [25, 62, 57, 63] or to learn a good learning strategy which can be applied in each episode (*meta learning*) [16, 29, 39, 38, 59, 48, 58, 52]. To train these models, the categories in a dataset are usually split into *training* categories used to train the models and *test* categories used during the evaluation procedure. Therefore, the few-shot model will be trained and tested on different categories, forcing it to generalize to novel categories.

3 One-shot object detection and instance segmentation on MS-COCO

The goal of one-shot object detection and instance segmentation is to develop models that can localize and segment objects from arbitrary categories when provided with a single visual example from that category. To this end, we 1) replace the widely used category-based object detection task by an example-based task setup and 2) split the available object categories into a training set and a non-overlapping test set, which is used to evaluate generalization to unknown categories. We use the popular MS-COCO dataset, which consists of a large variety of complex scenes with multiple objects from abroad range of categories and often challenging conditions like clutter.

Task setup: example-based instance segmentation. We define one-shot detection and segmentation as follows: Given a *reference image* showing a close-up example of a novel object category, find and segment all instances of objects belonging to this category in a separate *query image*, which shows an entire visual scene containing many objects (Figure 1, right). The main difference between this task and the usual object detection setup is the change from a category-based to an example-based setup. Instead of requiring to localize objects from a number of fixed categories, the example-based task requires to detect objects from a single category, which is defined through a reference image. The reference image shows a single object instance of the category that is to be detected, cropped to its bounding box (see Figure 1 for two examples). It is provided without mask annotations.

Split of categories for training and testing. To be able to evaluate performance on novel categories, we split the 80 object categories in MS-COCO into 60 *training* and 20 *test* categories. Following earlier work on Pascal VOC [56], we generate four such training/test splits by including every fourth category into the test split, starting with the first, second, third or fourth category, respectively (see Table A1 in the Appendix).

Because we use complex scenes which can contain objects from many categories, it is not feasible to ensure that the training images contain no instances of held-out categories. However, we do not provide any annotations for these categories during training and never use them as references. In other words, the model will see objects from the test categories during training, but is never provided with any information about them. This setup differs from the typical few-shot learning setup, in which the model never encounters any instance of the novel objects during training. However, in addition to being the only feasible solution, we consider this setup quite realistic for an autonomous agent, which may encounter unlabeled objects multiple times before they become relevant and label information is provided. Think of a household robot seeing, but not recognizing, a certain type of toy in various parts of the apartment multiple times before you instruct it to go pick it up for you.

Evaluation procedure. We propose to evaluate task performance using the following procedure:

1. Choose an image from the test set
2. Draw a random reference image for each of the (novel) test categories present in the image
3. Predict bounding boxes for each reference image separately
4. Assign the computed predictions to the category of the corresponding reference image
5. Repeat this process for all images in the test set
6. Compute mAP50 [14] using the standard tools from object detection [1] [1]

The same steps as above apply in the case of instance segmentation, with the difference that a segmentation mask instead of a bounding box is required for each predicted object.

Our evaluation procedure is simplified somewhat, because we ensure that the reference categories are actually present in each image used for evaluation. For a real-world application of such a system, this

[1] We chose to use mAP50 (mAP @ 50% Bounding Box IoU [14]) instead of the COCO metric mAP (mean of mAP @ 50, 55, ..., 95% Bounding Box IoU [33]), because we think it more directly reflects the result we are primarily interested in: whether our model can find novel objects based on a single reference image. For results using the MS-COCO metric see Appendix Section A6

restriction would have to be removed. However, we found the task to be very challenging already with this simplification, so we believe it is justified for the time being.

Connection to few-shot learning and object detection. Our evaluation procedure lends from other few-shot setups that typically evaluate in episodes. Each episode consists of a support set (the training examples for the novel categories) and a query set (the images to be classified). In our case, an episode consists of the detection of objects of one novel category in one image. In this case, the support set is the set of examples from the category to be detected (the *references*) while the query set is a single image (the *query image*). Compared to object detection, the classifier is turned into a binary verification conditioned on the reference image(s). Compared to the typical few-shot learning setup, there are two key differences: First, as only one category is given, the task is not a discrimination task between the given categories, but a verification task between the given category and all other object categories. Second the query image may not only contain objects from the novel category given by the reference, but also other objects from known and unknown categories.

Connection to other related tasks. Our setup differs from a number of related paradigms. In contrast to recent work on few-shot object detection [13, 7, 24, 55], we formulate our task as an example-based search task rather than learning an object detector from a small labeled dataset. This allows us to directly apply our model on novel categories without any retraining. We also extend all of these approaches by additionally asking the system to output segmentation masks for each instance and focus on the challenging MS-COCO dataset. Similarly our task shares similarities with zero-shot object detection [3, 42, 11, 69], but with the crucial difference that in zero-shot detection the reference category is defined by a textual description instead of an image.

A range of one-shot segmentation tasks exist, including one-shot semantic segmentation [56, 43, 12, 37], texture segmentation [61], medical image segmentation [67] and recent work on co-segmentation [28][2]. The key difference is that the models developed for these tasks output pixel-level semantic classifications rather than instance-level masks and, thus, cannot distinguish individual object instances. In co-segmentation very recent work [23] explores instance co-segmentation, but not in a few-shot setting. Two studies segment instances in a few-shot setting, but with different task setups: (1) in one-shot video segmentation [5, 6], object instances are tracked across a video sequence; (2) in one-shot instance segmentation of homogeneous object clusters [65] a model is proposed which segments, e. g., a pile of bricks into the individual instances based on a video pan of one of the bricks. Both of these setups are closer to particular object retrieval [60, 53, 19], as they localize instances of a particular object rather than instances of the same object category, as is the focus of our work.

4 Siamese Mask R-CNN

The key idea of one-shot instance segmentation is to detect and segment object instances based on a single visual example of some object category. Thus, our system has to deal with arbitrary, potentially previously unknown object categories which are defined only through a single reference image, rather than with a fixed set of categories for which extensive labeled data was provided during training. To solve this problem, we take a metric-learning approach: we learn a similarity metric between the reference and image regions in the scene. Based on this similarity metric, we then generate object proposals and classify them into matches and non-matches. The key advantage of this approach is that it can be directly applied to objects of novel categories without the need to retrain or fine-tune the learned model.

To compute the similarity metric we use Siamese networks, a classic metric learning approach [4, 9, 25]. We combine this form of similarity judgment with the domain knowledge built into current state-of-the-art object detection and instance segmentation systems by integrating it into Mask R-CNN [21]. In the following paragraphs we provide a quick recap of Mask R-CNN before describing the changes we made to integrate the Siamese approach and how we compute the similarity metric. We build our implementation upon the Matterport Mask R-CNN library [2]. The details can be found in Appendix A2 and in our code[3].

[2] Most co-segmentation work (e.g. [51, 15]) uses the same object categories during training and test time and therefore does not operate in the few-shot setting

[3] `https://github.com/bethgelab/siamese-mask-rcnn`

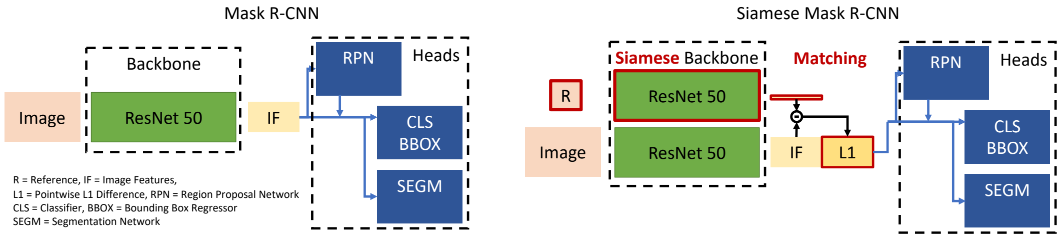

Mask R-CNN
Backbone
Image
ResNet 50
IF
RPN
Heads
CLS
BBOX
SEGM
R = Reference, IF = Image Features,
L1 = Pointwise L1 Difference, RPN = Region Proposal Network
CLS = Classifier, BBOX = Bounding Box Regressor
SEGM = Segmentation Network
Siamese Mask R-CNN
Siamese Backbone
Matching
R
Image
ResNet 50
ResNet 50
IF
L1
RPN
Heads
CLS
BBOX
SEGM

Figure 2: Comparison of Mask R-CNN and Siamese Mask R-CNN. The main differences (marked in red) of our model are (1) the Siamese backbone which jointly encodes the image and reference, and (2) the matching of those embeddings to target the region proposal and classification heads towards the reference category.

Mask R-CNN. Mask R-CNN is a two-stage object detector that consists of a backbone feature extractor and multiple heads operating on these features (see Figure 2). The heads consist of two stages. First, the region proposal network (RPN) is applied convolutionally across the image to predict possible object locations in the scene. The most promising region proposals are then cropped from the backbone feature maps and used as inputs for the bounding box classification (CLS) and regression (BBOX) head as well as the instance masking head (MASK).

Siamese network backbone. To integrate the reference information into Mask R-CNN, the same backbone (ResNet50 [22] with Feature Pyramid Networks (FPN) [31]) is used with shared weights to extract features from both the reference and the scene.

Feature matching. To obtain a measure of similarity between the reference and different regions of the query image, we treat each (x,y) location of the encoded features of the query image as an embedding vector and compare it to the embedding of the reference image. This procedure can be viewed as a non-linear template matching in the embedding space instead of the pixel space. The matching procedure works as shown in Figure 3:

1. Average pool the features of the reference image to an embedding vector. In the few-shot case (more than one reference) compute the average of the reference features as in prototypical networks [57].
2. Compute the absolute difference between the reference embedding and that of the scene at each (x,y) position.
3. Concatenate this difference to the scene representation.
4. Reduce the number of features with a 1×1 convolution.

The resulting features are then used as a drop-in replacement for the original Mask R-CNN features [4]. The key difference is that they do not only encode the content of the scene image, but also its similarity to the reference image, which forms the basis for the subsequent heads to generate object proposals, classify matches vs. non-matches and generate instance masks.

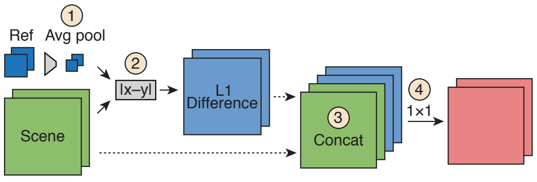


Figure 3: Sketch of the matching procedure.

Head architecture Because the computed features can be used as a drop-in replacement for the original features, we can use the same region proposal network and ROI pooling operations as Mask R-CNN. We can also use the same classification and bounding box regression head as Mask R-CNN, but change the classification from an 80-way category discrimination to a binary match/non-match discrimination and generate only a single, class-agnostic set of bounding box coordinates. Similarly, for the mask branch we predict only a single instance mask instead of one per potential category.

[4] As we use a backbone with feature pyramid networks (FPN) we get features at multiple resolutions. We therefore simply apply the described matching procedure at each resolution independently.

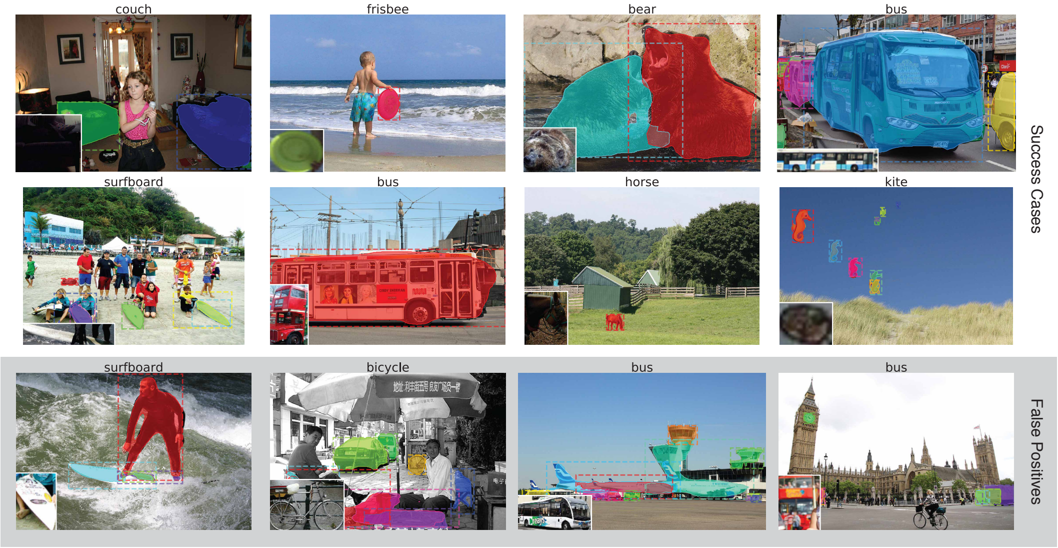


Figure 4: Examples of Siamese Mask R-CNN operating in the one-shot setting, i.e. segmenting novel objects which are not known from training (split S_2). The only information our model has about these categories is one reference image (shown in the lower-left corner of each example; the categories in the titles are just for the reader). The top two rows show success cases while the last row displays some results with a lot of false positives. Best viewed with zoom and color.

5 Experiments

We train Siamese Mask R-CNN jointly on object detection and instance segmentation in the example-based setting using the training set of MS-COCO. We train one model on each of the four category splits defined in Section 3 and evaluate the trained models on both known (train) and unknown (test) categories using the MS-COCO validation set. In the following paragraphs, we highlight the most important changes between our training and evaluation protocol and that of Mask R-CNN. The full training and evaluation details are given in Appendix A3 and A4.

Training. We first pre-train the ResNet backbone on a reduced subset of ImageNet, which contains only images from the 687 ImageNet categories that have no correspondence in MS-COCO. We do this to avoid using any label information about the test categories during pre-training.

We then proceed by training episodically. For each image in a minibatch, we pick a random reference category among the training categories present in the image. We then crop a random instance of this category out of another random image in the training set. We keep only the annotations of this category; all other objects are treated as background.

Evaluation. We evaluate our model using the procedure described in Section 3. Each category split is evaluated separately. The final score is the mean of the scores from all four splits. This evaluation procedure is stochastic due to the random selection of references. We thus repeat the evaluation five times and report the average and 95% confidence intervals.

Baseline: random boxes. As a simple sanity check, we evaluate the performance of a model predicting random bounding boxes and segmentation masks. To do so, we take ground-truth bounding boxes and segmentation masks for the category of the reference image, and randomly shift the boxes around the image (assigning a random confidence value for each box between 0.8 and 1). We keep the ground-truth segmentation masks intact in the shifted boxes. This procedure allows us to get random predictions while keeping certain statistics of the ground-truth annotations (*e.g.* number of boxes per image, their sizes, etc.).

	Categories used in training		Novel categories		Random
	1-shot	5-shot	1-shot	5-shot	
Object detection	37.6 ± 0.2	41.3 ± 0.1	16.3 ± 0.1	18.5 ± 0.1	1.2 ± 0.1
Instance segmentation	34.9 ± 0.1	38.4 ± 0.1	14.5 ± 0.1	16.8 ± 0.1	0.5 ± 0.1

Table 1: Results on MS-COCO (in % mAP50 with 95% confidence intervals). Three settings are reported: Evaluating on training (train), novel (test) categories and randomly drawn boxes (random). We run our models with one or five references per category and image (shots).

6 Results

Example-based detection and segmentation. We begin by applying the trained Siamese Mask R-CNN model to detect objects from the categories used for training. In this setting, all of the training examples are used to learn the metric, but the detection is based only on the similarity to one (or five) instance(s) from the reference category. IWith one reference, we achieve 37.6% and 34.9% mAP50 for object detection and instance segmentation, respectively. With five references, we achieve 41.3% and 38.4%, respectively (Table 1). We also report the 95% confidence interval estimated from five evaluation runs to quantify the variability introduced by to the random selection of reference images. The variation is below 0.2 percentage points in all cases, which suggests that evaluating five times is sufficient to handle the variability. We observe some additional variation between the splits, which seems to stem mostly from the over-representation of the person category (see Appendix Table A2 for results of each split).

One-shot instance segmentation. Next, we report the results of evaluating Siamese Mask R-CNN on novel categories not used for training, showcasingits ability to generalize to the 20 held-out categories that have not been annotated during training. With one reference (one-shot), the average detection mAP50 score for the test splits is 16.3%, while the segmentation performance is 14.5% (Table 1). While these values are significantly lower than those for the training categories, they still present a strong baseline and are far from chance (1.2%/0.5% for detection/segmentation) despite the difficulty of the one-shot setting. When using five references (five-shot), the performance improves to 18.5% and 16.7%, respectively. Taken together, these results suggest that the metric our model has learned allows some generalization outside of the training categories, but a substantial degree of overfitting on the those categories remains.

Qualitative analysis. The first two rows of Figure 4 show some examples of successful detection and segmentation of objects from novel categories. These examples allow us to get a feeling for the difficulty of the task: the reference inputs are quite different from the instances in the query image, often showing different perspectives, usually very different instances of the category and sometimes only parts of the reference object. Also note that the ground truth segmentation mask is not used to pre-segment the reference.

To generate bounding boxes and segmentation masks, the model can thus use only its general knowledge about objects. It has to rely on the metric learned on the categories annotated during training to decide whether the reference and the query instances belong to the same category. For instance, the bus and the horse in the second row of Figure 4 are incomplete and the network has never been provided with ground truth bounding boxes or instance masks for either horses or buses. Nevertheless, it still finds the correct object in the query image and segments the entire object.

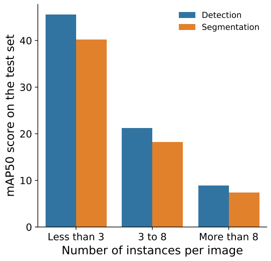


Figure 5: Results on split S_2 (in % mAP50) separated by the number of instances per image.

We also show examples of failure cases in the last row of Figure 4. The picture that emerges from both successful and failure cases is that the network produces overall good bounding boxes and segmentation masks, but often fails at targeting them towards the correct category. We elaborate more on the challenges of the task in the following paragraphs.

False positives when evaluating on novel categories. There is a marked drop in model performance between evaluating on the categories used during training and the novel categories, suggesting some degree of overfitting to the training categories. If this is indeed the case, we would expect false positives to be biased towards these categories and, in particular, towards those categories that are most frequent in the training set. Qualitatively, this bias seems indeed to exist (Figure 4). We verified this assumption quantitatively by computing a confusion matrix between categories (Appendix Figure A1). The confusion matrix shows that objects from the training categories are often falsely detected when searching for objects of the novel categories. Among the most commonly falsely detected categories are people, cars, airplanes and clocks which are overrepresented in the dataset.

Effect of image clutter. Previous work on synthetic data found that cluttered scenes are especially challenging in example based one-shot tasks [37]. This effect is also present in the current context. Both detection and segmentation scores are substantially higher for images with a small number of total instances (Figure 5), underscoring the importance of extending the model to robustly process cluttered scenes.

7 Related work

As outlined in section 3, our approach lies at the intersection of few-shot/metric learning, object detection/visual search, and instance segmentation. Each of these aspects has been investigated extensively. The novelty of our approach is the combination of all these aspects. A number of very recent and, to a large extent concurrent, works have started addressing few-shot detection. We review the most closely related work below. We are not aware of any previous work on category-based few-shot instance segmentation.

Dong et al. [13] train a semi-supervised few-shot detector on the 20 categories of Pascal VOC using roughly 80 annotated images, supplemented by a large set of unlabeled images. They train a set of models, each of which generates training labels for the other models by using high-confidence detections in the unlabeled images. The low-shot transfer detector (LSTD) [7] fine-tunes an object detector on a transfer task with new categories using two novel regularization terms: one for background depression and one for knowledge transfer from the source domain to the target domain. Kang et. al. [24] extend a single-stage object detector – YOLOv2 [45] – by a category-specific feature reweighting that is predicted by a meta model, allowing them to incorporate novel classes with few examples. Schwartz et. al. [55] replace the classification branch of Faster R-CNN with a metric learning module, which evaluates the similarity of each predicted box to a set of prototypes generated from the few provided examples. Very recent concurrent work [66] evaluates the same task as we do for object detection on Pascal VOC using Faster R-CNN, although they employ separate feature fusions in the RPN and classifier head instead of the unified matching we employ. Recent works on zero-shot detection [3, 42, 11, 69] use a similar approach to ours to target the detection towards a novel category, except that they learn a joint embedding for the query image and a textual description (instead of a visual description) of this novel category.

8 Discussion

We introduced the task of *one-shot instance segmentation* which requires models to generalize to object categories that have not been labeled during training in the challenging setting of instance segmentation. To address this task we proposed *Siamese Mask R-CNN*, a model combining a state-of-the-art instance segmentation model (Mask R-CNN) with a metric learning approach (Siamese networks). This model can detect and segment objects from novel categories based on a single reference image. While our approach is not as successful on novel categories as on those used for training, it performs far above chance, showcasing it's ability to generalize to categories outside of the training set. Generally, it is expected from any reasonable learning system that it should perform better on object categories for which it has been trained with thousands of examples than for those encountered in a few-shot setting. Considering the difficulty of this problem, the performance of our model should provide a strong baseline and we hope that our work provides a first step towards visual search algorithms with human like flexibility.

Acknowledgements

We want to thank Mengye Ren, Jake Snell, James Lucas, Marc Law, Richard Zemel and Eshed Ohn-Bar for helpful discussion. This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) via grant EC 479/1-1 to A.S.E and the Collaborative Research Center (Projektnummer 276693517 – SFB 1233: Robust Vision), by the German Federal Ministry of Education and Research through the Tübingen AI Center (FKZ 01IS18039A), by the International Max Planck Research School for Intelligent Systems (C.M. and I.U.), and by the Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DoI/IBC) contract number D16PC00003. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/IBC, or the U.S. Government.

References

[1] Coco website, leaderboard and api. `http://cocodataset.org/`.

[2] Waleed Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. `https://github.com/matterport/Mask_RCNN`, 2017.

[3] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-Shot Object Detection. *ECCV*, 2018.

[4] Jane Bromley, James W. Bentz, Léon Bottou, Isabelle Guyon, Yann LeCun, Cliff Moore, Eduard Säckinger, and Roopak Shah. Signature Verification Using A "Siamese" Time Delay Neural Network. *IJPRAI*, 1993.

[5] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, 2017.

[6] Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool. The 2019 DAVIS Challenge on VOS: Unsupervised Multi-Object Segmentation. *arXiv:1905.00737*, 2019.

[7] Hao Chen, Yali Wang, Guoyou Wang, and Yu Qiao. LSTD: A Low-Shot Transfer Detector for Object Detection. *AAAI*, 2018.

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *TPAMI*, 2018.

[9] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005.

[10] Jifeng Dai, Kaiming He, and Jian Sun. Convolutional Feature Masking for Joint Object and Stuff Segmentation. In *CVPR*, 2015.

[11] Berkan Demirel, Ramazan Gokberk Cinbis, and Nazli Ikizler-Cinbis. Zero-shot object detection by hybrid region embedding. In *BMVC*, 2018.

[12] Nanqing Dong and Eric P Xing. Few-Shot Semantic Segmentation with Prototype Learning. In *BMVC*, 2018.

[13] Xuanyi Dong, Liang Zheng, Fan Ma, Yi Yang, and Deyu Meng. Few-Example Object Detection with Model Communication. *TPAMI*, 2018.

[14] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 2010.

[15] Alon Faktor and Michal Irani. Co-segmentation by Composition. In *ICCV*, 2013.

[16] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML*, 2017.

[17] Ross Girshick. Fast r-cnn. In *ICCV*, 2015.

[18] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*, 2014.

[19] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end Learning of Deep Visual Representations for Image Retrieval. *IJCV*, 2016.

[20] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous Detection and Segmentation. In *ECCV*, 2014.

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[23] Kuang-Jui Hsu, Yen-Yu Lin, and Yung-Yu Chuang. Deepco^3: Deep instance co-segmentation by co-peak search and co-saliency detection. In *CVPR*, 2019.

[24] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. *arXiv:1812.01866*, 2018.

[25] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese Neural Networks for One-shot Image Recognition. *ICML*, 2015.

[26] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Shahab Kamali, Matteo Malloci, Jordi Pont-Tuset, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://storage.googleapis.com/openimages/web/index.html*, 2017.

[27] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[28] Weihao Li, Omid Hosseini Jafari, and Carsten Rother. Deep Object Co-Segmentation. In *ACCV*, 2018.

[29] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-SGD: Learning to Learn Quickly for Few-Shot Learning. *arXiv:1707.09835*, 2017.

[30] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *TPAMI*, 2006.

[31] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *CVPR*, 2017.

[32] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. *ICCV*, 2017.

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014.

[34] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single Shot MultiBox Detector. In *ECCV*, 2016.

[35] Yaoyao Liu, Qianru Sun, An-An Liu, Yuting Su, Bernt Schiele, and Tat-Seng Chua. LCC: learning to customize and combine neural networks for few-shot learning. *arXiv:1904.08479*, 2019.

[36] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[37] Claudio Michaelis, Matthias Bethge, and Alexander Ecker. One-Shot Segmentation in Clutter. In *ICML*, 2018.

[38] Tsendsuren Munkhdalai and Adam Trischler. Metalearning with Hebbian Fast Weights. *arXiv:1807.05076*, 2018.

[39] Tsendsuren Munkhdalai and Hong Yu. Meta Networks. In *ICML*, 2017.

[40] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015.

[41] Pedro O. Pinheiro, Ronan Collobert, and Piotr Dollar. Learning to Segment Object Candidates. In *NIPS*, 2015.

[42] Shafin Rahman, Salman Khan, and Fatih Porikli. Zero-Shot Object Detection: Learning to Simultaneously Recognize and Localize Novel Concepts. In *ACCV*, 2018.

[43] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alyosha Efros, and Sergey Levine. Conditional Networks for Few-Shot Semantic Segmentation. *arXiv:1806.07373*, 2018.

[44] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *CVPR*, 2016.

[45] Joseph Redmon and Ali Farhadi. YOLO9000: Better, Faster, Stronger. In *CVPR*, 2017.

[46] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. *arXiv:1804.02767*, 2018.

[47] M. Ren and R. S. Zemel. End-to-End Instance Segmentation with Recurrent Attention. In *CVPR*, 2017.

[48] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-Learning for Semi-Supervised Few-Shot Classification. In *ICLR*, 2018.

[49] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[50] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*, 2015.

[51] Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *CVPR*, 2006.

[52] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-Learning with Latent Embedding Optimization. In *ICLR*, 2019.

[53] Amaia Salvador, Xavier Giro-i Nieto, Ferran Marques, and Shin'ichi Satoh. Faster R-CNN Features for Instance Search. In *CVPR DeepVision workshop*, 2016.

[54] Andries F Sanders and Mieke Donk. Visual search. In *Handbook of perception and action*, volume 3, pages 43–77. Elsevier, 1996.

[55] Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Sharathchandra Pankanti, Rogerio Feris, Abhishek Kumar, Raja Giryes, and Alex M. Bronstein. RepMet: Representative-based metric learning for classification and one-shot object detection. In *CVPR*, 2019.

[56] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-Shot Learning for Semantic Segmentation. *BMVC*, 2017.

[57] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical Networks for Few-shot Learning. In *NIPS*, 2017.

[58] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *CVPR*, 2019.

[59] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H S Torr, and Timothy M Hospedales. Learning to Compare: Relation Network for Few-Shot Learning. *CVPR*, 2018.

[60] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of CNN activations. *ICLR*, 2016.

[61] Ivan Ustyuzhaninov, Claudio Michaelis, Wieland Brendel, and Matthias Bethge. One-shot Texture Segmentation. *arXiv:1807.02654*, 2018.

[62] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NIPS*, 2016.

[63] Yong Wang, Xiao-Ming Wu, Qimai Li, Jiatao Gu, Wangmeng Xiang, Lei Zhang, and Victor O. K. Li. Large Margin Few-Shot Learning. *arXiv:1807.02872*, 2018.

[64] Jeremy M. Wolfe, George A. Alvarez, Ruth Rosenholtz, Yoana I. Kuzmova, and Ashley M. Sherman. Visual search for arbitrary objects in real scenes. *Attention, perception & psychophysics*, 73(6):1650–1671, August 2011.

[65] Zheng Wu, Ruiheng Chang, Jiaxu Ma, Cewu Lu, and Chi-Keung Tang. Annotation-free and one-shot learning for instance segmentation of homogeneous object clusters. In *IJCAI*, 2018.

[66] Tengfei Zhang, Yue Zhang, Xian Sun, Hao Sun, Menglong Yan, Xue Yang, and Kun Fu. Comparison network for one-shot conditional object detection. *arXiv:1904.02317*, 2019.

[67] Amy Zhao, Guha Balakrishnan, Frédo Durand, John V. Guttag, and Adrian V. Dalca. Data augmentation using learned transforms for one-shot medical image segmentation. In *CVPR*, 2019.

[68] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.

[69] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. Zero shot detection. In *TCSVT*, 2019.

Changes to previous version

Compared to the previous version (submitted to arxiv on 28 Nov 2018) this version additionally includes:

- a different evaluation procedure evaluating each split 5-times and reporting the mean and 95% confidence interval.
- five-shot results using a prototypical approach to accomodate multiple reference images.
- a background section introducing information and notation of object detection and few-shot learning tasks.
- discussion of concurrent work which was published on arxiv since the publication of the previous version [24, 66].
- detailed description of the training and evaluation process in the Appendix.
- results for all metrics evaluated on the MS-COCO leaderboard to the Appendix.

Additionally to adding content we reworked large parts of the text to clarify the task setup the way we present related tasks and the corresponding solutions. We also update some of the figures, mainly combining the two figures for qualitative analysis into one figure which includes good and bad examples, adding a comparison with traditional few-shot learning tasks to the introduction figure and making the color coding in the model figure easier to understand.

Appendix

A1 Training and testing categories

This section contains the description of the category splits from Section 3 from the main paper as well as a table of those categories.

A1.1 Splits S_1-S_4

To be able to evaluate performance on novel categories we hold out some categories during training. We split the 80 object categories in MS-COCO into 60 *training* and 20 *test* categories. Following earlier work on Pascal VOC [56], we generate four such training/test splits by including every fourth category into the test split starting with the first, second, third or fourth category, respectively. These splits are shown in Table A1 below.

A1.2 Rationale

Providing four splits with equally distributed held-out categories has two main advantages: It allows to test on all categories in MS-COCO (albeit with different models) while sub sampling the super categories [33] as evenly as possible. This approach assumes that we will know some objects from all broad object categories in the world and that we can infer the missing parts from this knowledge. This setup differs from tasks like *tiered*ImageNet [48] which require generalization to objects from vastly different categories.

A2 Implementation details

A2.1 Backbone

We use the standard architecture of ResNet-50 [22] without any modifications.

A2.2 Feature matching

- We use layers[5] `res2c_relu` (256 features), `res3d_relu` (512), `res4f_relu` (1024) and `res5c_relu` (2048) of the backbone as a feature representation of the inputs. For brevity, we refer to these layers as C_2, C_3, C_4 and C_5.

[5] Using the notation from here: https://ethereon.github.io/netscope/#/gist/db945b393d40bfa26006

S_1		S_2		S_3		S_4	
1	Person	2	Bicycle	3	Car	4	Motorcycle
5	Airplane	6	Bus	7	Train	8	Truck
9	Boat	10	Traffic light	11	Fire Hydrant	12	Stop sign
13	Parking meter	14	Bench	15	Bird	16	Cat
17	Dog	18	Horse	19	Sheep	20	Cow
21	Elephant	22	Bear	23	Zebra	24	Giraffe
25	Backpack	26	Umbrella	27	Handbag	28	Tie
29	Suitcase	30	Frisbee	31	Skis	32	Snowboard
33	Sports ball	34	Kite	35	Baseball bat	36	Baseball glove
37	Skateboard	38	Surfboard	39	Tennis rocket	40	Bottle
41	Wine glass	42	Cup	43	Fork	44	Knife
45	Spoon	46	Bowl	47	Banana	48	Apple
49	Sandwich	50	Orange	51	Broccoli	52	Carrot
53	Hot dog	54	Pizza	55	Donut	56	Cake
57	Chair	58	Couch	59	Potted plant	60	Bed
61	Dining table	62	Toilet	63	TV	64	Laptop
65	Mouse	66	Remote	67	Keyboard	68	Cell phone
69	Microwave	70	Oven	71	Toaster	72	Sink
73	Refrigerator	74	Book	75	Clock	76	Vase
77	Scissors	78	Teddy bear	79	Hair drier	80	Toothbrush

Table A1: Category splits (S_1 – S_4, Section 3) of MS-COCO.

- FPN generates multi-scale representations P_i, $i = \{2, 3, 4, 5, 6\}$ consisting of 256 features (for all i) as follows. P_5 is a result of applying a 1×1 conv layer to C_5 (to get 256 features). P_i ($i = \{2, 3, 4\}$) is a sum of a 1×1 conv layer applied to C_i and up-sampled (by a factor of two on each side) P_{i+1}. P_6 is a down-sampled P_5 (by a factor of two on each side).
- The final similarity scores between the input scene and the reference at scale i are computed by obtaining P_i^{scene} and P_i^{ref} as described above, applying global average pooling to P_i^{ref}, and computing pixel-wise differences $D_i = \text{abs}(P_i^{\text{scene}} - \text{pool}(P_i^{\text{ref}}))$.
- The final feature representations containing information about similarities between the scene and the reference are computed by concatenating P_i^{scene} and D_i, and applying a 1×1 conv layer, outputting 384 features.

A2.3 Region Proposal Network (RPN)

- We use 3 anchor aspect ratios (0.5, 1, 2) at each pixel location for the 5 scales (32, 64, 128, 256, 512) $i = \{2, \ldots, 6\}$ defined above, resulting in $3 \times (32^2 + \ldots + 512^2) \approx 1$M proposals in total.
- The architecture is a $3 \times 3 \times 512$ conv layer, followed by the 1×1 conv outputting k times number of anchors per location (three in our case) features (corresponding to proposal logits for $k = 2$ or to bounding box deltas for $k = 4$).

A2.4 Classification and bounding box regression head

The classification head produces same/different classifications for each proposal and performs bounding box regression.

- Inputs: the computed bounding boxes (outputs of the RPN) are cropped from P_i, reshaped to 7×7, and concatenated for $i = \{2, \ldots, 5\}$. Only 6000 top scoring anchors are processed for efficiency.
- Architecture: two fc-layers (1024 units with ReLU) followed by a logistic regression into 2 classes (same as reference or not).
- Bounding box regression is part of the classification branch, but uses a different output layer. This output layer produces fine adjustments (deltas) of the bounding box coordinates (instead of class probabilities).
- Non-maximum suppression (NMS; threshold 0.7) is applied to the predicted bounding boxes.

A2.5 Segmentation head

- Inputs: the computed bounding boxes are cropped from P_i, reshaped to 14×14, and concatenated for $i = \{2, \ldots, 5\}$.
- Architecture: four 3×3 conv layers (with ReLU and BN) followed by a transposed conv layer with 2×2 kernels and stride of 2, and a final 1×1 conv layer outputting two feature maps consisting of logits for foreground/background at each spatial location.

A3 Training details

This section contains a detailed description of the training procedure. To make this section more readable and have all relevant information in one place it contains a few duplications with Section 5

Pre-training backbone. We pre-train the ResNet backbone on image classification on a reduced subset of ImageNet, which contains images from the 687 ImageNet categories without correspondence in MS-COCO – hence we refer to it as *ImageNet-687*. Pre-training on this reduced set ensures that we do not use any label information about the test categories at any training stage.

Training Siamese Mask R-CNN. We train the models using stochastic gradient descent with momentum for 160,000 steps with a batch size of 12 on 4 NVIDIA P100 GPUs in parallel. With this setup training takes roughly a week. We use an initial learning rate of 0.02 and a momentum of 0.9. We start our training with a warm-up phase of 1,000 steps during which we train only the heads. After that, we train the entire network, including the backbone and all heads, end-to-end. After 120,000 steps, we divide the learning rate by 10.

Construction of mini-batches. During training, a mini-batch contains 12 sets of reference and query images. We first draw the query images at random from the training set and pre-process them in the following way: (1) we resize an image so that the longer side is 1024 px, while keeping the aspect ratio, (2) we zero-pad the smaller side of the image to be square 1024×1024, (3) we subtract the mean ImageNet RGB value from each pixel. Next, for each image, we generate a reference image as follows: (1) draw a random category among all categories of the background set present in the image, (2) crop a random instance of the selected category out of any image in the training set (using the bounding box annotation), and (3) resize the reference image so that its longer side is 192 px and zero-pad the shorter side to get a square image of 192×192. To enable a quick look-up of reference instances, we created an index that contains a list of categories present in each image.

Labels. We use only the annotations of object instances in the query image that belong to the corresponding reference category. The annotations of all other objects are removed and subsequently they are treated as background.

Loss function. Siamese Mask R-CNN is trained on the same basic multi-task objective as Mask R-CNN: classification and bounding box loss for the RPN; classification, bounding box and mask loss for each RoI. There are a couple of differences as well. First, the classification losses consist of a binary cross-entropy of the match/non-match classification rather than an 80-way multinomial cross-entropy used for classification on MS-COCO. Second, we found that weighting the individual losses differently improved performance in the one-shot setting. Specifically, we apply the following weights to each component of the loss function: RPN classification loss: 2, RPN bounding box loss: 0.1, RoI classification loss: 2, RoI bounding box loss: 0.5 and mask loss: 1.

Exact hyper parameter details Complex systems like Mask R-CNN require a large set of hyper parameters to be set for optimal training performance. We mentioned all changes we made to the hyperparameter settings of the implementation we extended [2]. For the full list of hyperparameter settings and exact details of our loss function implementation and data handling please refer to the code: `https://github.com/bethgelab/siamese-mask-rcnn`

A4 Evaluation details

This section contains a detailed description and discussion of the evaluation procedure. As with the training section it contains a few duplications with the corresponding Section 3 from the main paper in order to have all information in one place.

A4.1 Category selection

The evaluation is performed on the MS Coco 2017 validation set (which corresponds to the 2014 minval set). The evaluation is performed for 4 subtasks, each using 60 categories for training and the remaining 20 categories for one-shot evaluation. Those 20 categories are selected by choosing every 4th category, therefore the ith split is constructed by: $[i + 4 * k \text{ for } k \text{ in range } (20)]$. An explicit listing of all 4 splits can be found in Table A1 above.

A4.2 Evaluation procedure

Each of the subtasks is evaluated over the whole validation set using the corresponding set of categories. Therefore for each image the present categories from the current split are determined. Then for each present category a reference instance is randomly chosen from the whole evaluation set (those references are chosen individually for each image). The model is then evaluated for each of the references and the predictions of each of these runs is assigned to the corresponding category. If no category from the current split is present the image is skipped. After running this over all images the results contain predicted bounding boxes for each image but only for the categories of the selected split. These collected results can then be fed to a slightly modified version of the official MS-COCO analysis tools [1] which can handle specific category subsets to get the final mAP50 scores.

```
for image in images do
    present categories = get one shot categories(image);
    for category in present categories do
        ref = get random instance (category, images);
        results[image, category] = model.predict (ref, image);
    end
end
mAP50 = evaluate mAP50(results, one shot categories);
```

Algorithm 1: Pseudocode for evaluation procedure

A4.3 Noise induced by random reference sampling

Because only one reference is sampled per category and image the predictions can be rather noisy (especially in the one-shot case). For our model the std of the predicted results is $\pm 1\%$. To get a good prediction of the actual mean we run the evaluation of each split 5 times thus reaching reaching a standard error of the mean of less than $\pm 0.2\%$.

A4.4 Comment on the evaluation procedure

We specifically chose to evaluate our model only on the categories present in each image. We think, that this scenario can realistically be assumed in real world tasks as a whole-image classification network can be used to pre-select if the reference category is present in an image before running the bounding box and instance segmentation prediction network.

This choice, however, makes the task substantially easier than evaluating each image for all categories. It does not punish false positives as hard as the other task does. However, as visible in our results, false positives play an important role even in our simpler task, which leads us to the conclusion, that our task setup is still sufficiently difficult.

A4.5 Note on non-maximum suppression

We use non-maximum suppression (NMS) on the predictions of each image/references combination individually and not on the combined output of an image after running the detection for all references because at test time the system needs to be able to detect and segment objects based on only a single reference example of each category separately.

A4.6 Choice of evaluation metric

We chose to use mAP50 instead of the so called "coco metric" mAP. mAP50 is evaluated at a single Intersection over Union (IoU) threshold of 50% between predicted and the ground truth bounding boxes (corresponding to around 70% overlap between two same-sized boxes/masks) while mAP is evaluated at IoU thresholds of [50%, 55%, ..., 95%] adding weight to exact bounding box/segmentation mask predictions.

We think, that mAP50 is the value most reflective of the result we are interested in: whether our model can find novel objects based on a single reference image. For instance segmentation the additional information about mask quality implicitly included in mAP might make sense. However we found, that correctly masking the sought objects was less of a problem for our model than correctly classifying them.

A5 Confusion matrix

To quantify the errors of our model we compute a confusion matrix over the 80 categories in MS-COCO using a model trained on split S^2 (Figure A1). The element (i, j) of this matrix corresponds to the AP50 value of detections obtained for reference images of category i, which are evaluated as if the reference images belonged to category j. If there were no false positives, the off-diagonal elements of the matrix would be zero. The sums of values in the columns show instances of categories that are most often falsely detected (the histogram of such sums is shown below the matrix). Among such commonly falsely predicted categories are people, cars, airplanes, clocks, and other categories that are common in the dataset.

A6 Additional results

In this section we discuss the noisiness of our evaluation approach and provide additional results including split-by-split values for the 95% confidence intervals we get from running the evaluation 5 times (Table A2) and the full results on all metrics evaluated on the MS-COCO leaderboard (`cocodataset.org/#detection-leaderboard`) for object detection (Tables A3 & A5) and instance segmentation (Tables A4 & A6).

A6.1 Noisiness of evaluation

The example based evaluation setting with a randomly drawn reference per category and image is naturally prone to be noisy. We therefore evaluate our models five times and take the mean of these 5 evaluations as our final result. We here want to discuss the amount of randomness generated by our evaluation procedure and the confidence of our mean.

We found the standard deviation of one-shot object detection and instance segmentation segmentation to be around 0.3% mAP50 while the standard deviation with five reference images is lower at 0.1% mAP50. The 95% confidence of the mean is around 0.1% (See Table 1. The rather small deviations can be seen as a result of the evaluation procedure which considers every image and reference category as a single instance. This ensures that there are many samples per category over test set.

A6.2 Results for each split

We show the results for each split (S^1-S^4) separately reporting mean and 95% confidence interval of five evaluation runs in Table A2. We find slight difference in performance between these split with split S^1 showing the biggest gap between evaluating on the training and test categories. We assume, that this is due to the strong over representation of the person category in MS-COCO [33]. With a lot of small instances and presence of persons in almost every image the removal of this category during training makes the dataset considerably easier, while requesting to detect them later is hard.

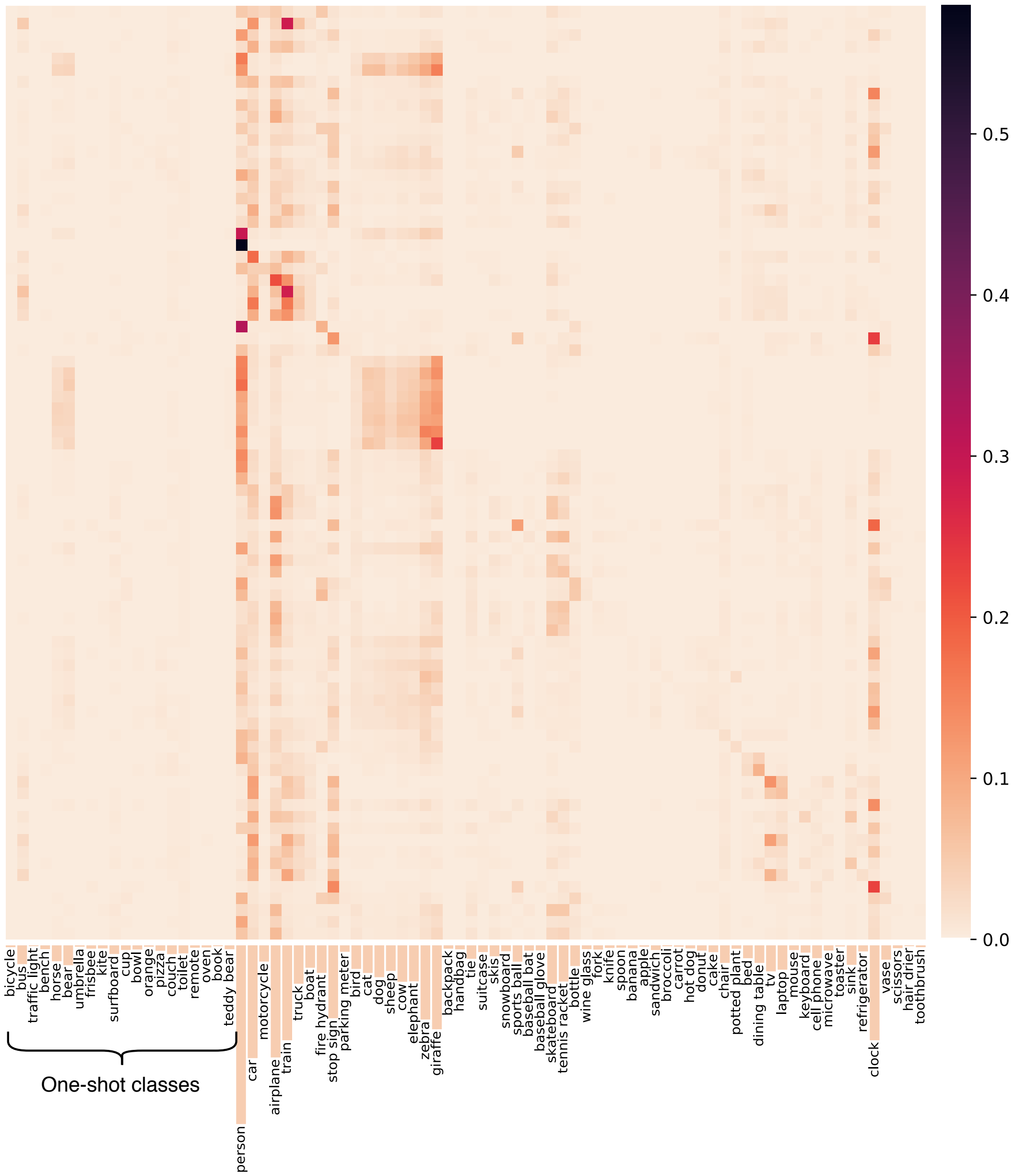


Figure A1: Confusion matrix for the Siamese Mask R-CNN model using split S_2 for one-shot evaluation. The element (i, j) shows the AP50 of using detections for category i and evaluating them as instances of category j. The histogram below the matrix shows the most commonly confused (or falsely predicted) categories.

A6.3 Full MS-COCO style results

In this section we report results on all metrics used at the MS-COCO leader board `cocodataset.org/#detection-leaderboard`. Beyond the mAP50 (AP^{50}) metric reported in the main paper these include the MS-COCO metric (AP) as well as other AP metrics at different thresholds (AP^{75}) and object sizes (AP^{S}, AP^{M}, AP^{L} each as subsets of AP) as well as recall metrics (AR) with varying numbers of detections (AR^{1}, AR^{10}, AR^{100}) and object sizes (AR^{S}, AR^{M}, AR^{L} each as parts of AR^{100}).

Object detection

Categories	Shots	S^1	S^2	S^3	S^4	$\emptyset$
Train	1	39.1 ± 0.1	36.6 ± 0.1	37.5 ± 0.1	37.2 ± 0.2	37.6 ± 0.1
	5	42.4 ± 0.1	40.5 ± 0.1	41.5 ± 0.1	40.9 ± 0.2	41.3 ± 0.1
Test	1	15.3 ± 0.2	16.8 ± 0.2	16.7 ± 0.2	16.4 ± 0.1	16.3 ± 0.1
	5	16.8 ± 0.1	20.0 ± 0.1	18.2 ± 0.1	19.0 ± 0.1	18.5 ± 0.1

Instance segmentation

Categories	Shots	S^1	S^2	S^3	S^4	$\emptyset$
Train	1	36.6 ± 0.1	33.5 ± 0.1	34.9 ± 0.1	34.5 ± 0.2	34.9 ± 0.1
	5	39.7 ± 0.1	37.3 ± 0.1	38.7 ± 0.1	37.9 ± 0.2	38.4 ± 0.1
Test	1	13.5 ± 0.2	14.9 ± 0.1	15.5 ± 0.2	14.2 ± 0.1	14.5 ± 0.1
	5	14.8 ± 0.1	18.0 ± 0.1	17.4 ± 0.1	16.9 ± 0.1	16.8 ± 0.1

Table A2: Results on MS Coco (in % mAP50 with 95% confidence intervals). In split S^i, every fourth category, starting at the i^{th}, is placed into the test set.

Model	AP	AP^{50}	AP^{75}	AP^{S}	AP^{M}	AP^{L}	AR^{1}	AR^{10}	AR^{100}	AR^{S}	AR^{M}	AR^{L}
full	21.8	35.5	23.4	11.1	21.8	30.8	19.9	37.6	39.2	22.2	41.0	56.5
train S^1	23.6	39.1	25.0	11.4	23.3	33.8	20.9	38.9	40.7	22.9	43.1	57.5
train S^2	21.9	36.6	23.5	11.4	22.6	31.1	19.9	37.9	39.4	22.7	41.9	57.1
train S^3	23.3	37.5	25.2	11.1	22.5	33.4	20.9	39.3	41.0	21.8	43.1	59.7
train S^4	22.7	37.2	24.2	11.9	21.6	31.7	20.1	38.5	40.4	23.2	42.4	56.7
test S^1	8.6	15.3	8.8	5.0	8.6	13.5	10.3	26.4	27.7	14.4	29.9	43.2
test S^2	9.8	16.8	10.1	5.7	8.4	14.8	12.2	26.7	27.7	13.9	27.6	43.9
test S^3	8.9	16.7	8.8	5.6	8.2	16.6	9.4	23.6	24.6	15.3	25.1	40.0
test S^4	9.1	16.4	9.2	5.4	9.4	14.0	10.9	25.7	27.4	14.5	30.7	43.2

Table A3: Full **one-shot detection** results on MS-COCO. train/test indicate evaluation on the training/test categories of split S^i respectively. Each value is the mean of 5 evaluation runs.

Model	AP	AP^{50}	AP^{75}	AP^{S}	AP^{M}	AP^{L}	AR^{1}	AR^{10}	AR^{100}	AR^{S}	AR^{M}	AR^{L}
full	19.3	33.1	19.9	9.3	19.4	27.5	17.9	33.5	34.9	19.4	36.9	49.9
train S^1	20.9	36.6	21.0	9.3	20.9	30.5	19.0	34.6	36.1	19.6	38.9	51.3
train S^2	18.9	33.5	19.4	9.2	19.3	27.4	17.8	33.2	34.5	19.6	36.7	50.0
train S^3	20.0	34.9	20.2	9.0	19.5	29.6	18.7	34.8	36.2	18.6	38.4	53.9
train S^4	20.0	34.5	20.9	9.9	19.0	28.6	18.2	34.3	35.7	20.3	37.5	51.2
test S^1	6.7	13.5	6.0	3.8	6.8	11.0	8.3	22.0	23.0	11.7	25.2	36.1
test S^2	8.5	14.9	8.7	4.7	7.4	12.8	10.8	23.5	24.5	11.7	24.7	39.3
test S^3	8.2	15.5	8.0	4.7	7.2	15.3	9.0	21.8	22.7	13.9	22.8	35.9
test S^4	7.3	14.2	6.6	3.8	7.9	11.8	9.3	22.3	23.8	12.0	28.2	38.2

Table A4: Full **one-shot segmentation** results on MS-COCO. train/test indicate evaluation on the training/test categories of split S^i respectively. Each value is the mean of 5 evaluation runs.

Model	AP	AP^{50}	AP^{75}	AP^{S}	AP^{M}	AP^{L}	AR^{1}	AR^{10}	AR^{100}	AR^{S}	AR^{M}	AR^{L}
full	24.9	40.5	26.7	13.3	25.0	35.9	21.8	40.1	41.8	23.9	44.3	59.1
train S^1	25.7	42.4	27.1	12.6	25.6	36.2	22.1	40.6	42.4	24.3	45.1	59.3
train S^2	24.3	40.5	26.1	12.8	25.1	35.3	21.4	39.7	41.3	24.1	44.2	59.9
train S^3	25.8	41.5	28.0	12.7	25.2	38.2	22.4	41.0	42.7	23.5	45.1	61.5
train S^4	25.1	40.9	26.8	12.9	23.8	36.3	21.5	40.3	42.3	24.7	44.5	59.1
test S^1	9.4	16.8	9.7	5.6	9.3	14.6	11.0	28.1	29.4	15.8	31.9	45.8
test S^2	11.7	20.0	12.1	6.3	9.7	19.3	13.3	29.1	30.3	15.1	30.7	48.3
test S^3	9.8	18.2	9.5	6.7	9.2	17.5	9.6	25.0	26.0	16.3	26.4	42.4
test S^4	10.6	19.0	10.6	5.8	10.4	16.6	11.8	27.8	29.6	14.8	33.1	47.5

Table A5: Full **five-shot detection** results on MS-COCO. train/test indicate evaluation on the training/test categories of split S^i respectively. Each value is the mean of 5 evaluation runs.

Model	AP	AP^{50}	AP^{75}	AP$^{\mathrm{S}}$	AP$^{\mathrm{M}}$	AP$^{\mathrm{L}}$	AR^{1}	AR^{10}	AR^{100}	AR$^{\mathrm{S}}$	AR$^{\mathrm{M}}$	AR$^{\mathrm{L}}$
full	22.0	37.8	22.9	11.4	22.2	32.4	19.7	35.8	37.2	21.0	39.5	52.6
train S^1	22.7	39.7	23.1	10.3	23.0	33.0	20.2	36.1	37.6	20.8	40.7	53.2
train S^2	21.0	37.3	21.7	10.5	21.7	31.2	19.2	34.9	36.3	20.7	38.7	52.9
train S^3	22.4	38.7	22.8	10.3	21.9	34.2	20.3	36.5	37.8	19.8	40.2	56.3
train S^4	22.1	37.9	23.2	10.6	20.8	32.9	19.5	35.9	37.6	21.4	39.3	53.6
test S^1	7.4	14.8	6.7	4.3	7.2	12.2	9.2	23.7	24.7	13.1	26.8	39.3
test S^2	10.2	18.0	10.5	5.1	8.6	17.2	12.0	26.0	27.0	12.5	27.7	44.1
test S^3	9.0	17.4	8.5	5.6	8.2	16.6	9.4	23.0	23.9	14.5	24.3	38.3
test S^4	8.5	16.9	7.8	4.1	8.8	14.4	10.3	24.3	25.9	12.3	30.4	42.0

Table A6: Full **five-shot segmentation** results on MS-COCO. train/test indicate evaluation on the training/test categories of split S^i respectively. Each value is the mean of 5 evaluation runs.

A Broad Dataset is All You Need for One-Shot Object Detection

Claudio Michaelis[1,§], Matthias Bethge[1] & Alexander S. Ecker[2]

[1] *University of Tübingen, Germany*
[2] *University of Göttingen, Germany*
[§] `claudio.michaelis@uni-tuebingen.de`

November 1, 2022

Abstract

Is it possible to detect arbitrary objects from a single example? A central problem of all existing attempts at one-shot object detection is the generalization gap: Object categories used during training are detected much more reliably than novel ones. We here show that this generalization gap can be nearly closed by increasing the number of object categories used during training. Doing so allows us to improve generalization from seen to unseen classes from 45% to 89% and improve the state-of-the-art on COCO by 5.4 %AP^{50} (from 22.0 to 27.5). We verify that the effect is caused by the number of categories and not the number of training samples, and that it holds for different models, backbones and datasets. This result suggests that the key to strong few-shot detection models may not lie in sophisticated metric learning approaches, but instead simply in scaling the number of categories. We hope that our findings will help to better understand the challenges of few-shot learning and encourage future data annotation efforts to focus on wider datasets with a broader set of categories rather than gathering more samples per category.

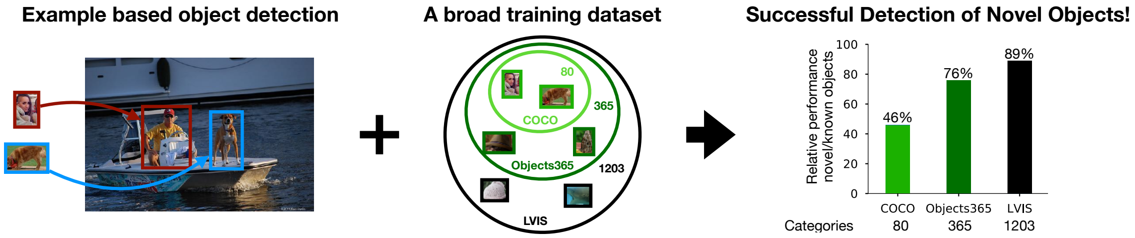


Figure 1: Example based object detectors can in theory detect any object based on an example image. However existing models trained on the datasets with few categories such as COCO perform significantly worse for novel than known objects. We here show that this generalization gap progressively shrinks when training the same models with more categories thus moving us closer to models which can actually detect any object.

1 Introduction

It's January 2021 and your long awaited household robot finally arrives. Equipped with the latest "Deep Learning Technology", it can recognize over 21,000 objects. Your initial excitement quickly vanishes as you realize that your casserole is not one of them. When you contact customer service they ask you to send some pictures of the casserole so they can fix this. They *tell you that the fix will be some time, though, as they need to collect about a thousand images of casseroles to retrain the neural network. While you are making the call your robot knocks over the olive oil because the steam coming from the pot of boiling water confused it. You start filling out the return form ...*

While not 100% realistic, the above story highlights an important obstacle towards truly autonomous agents: such systems should be able to detect novel, previously unseen objects and learn to recognize them based on (ideally) a single example. Solving this one-shot object detection problem can be decomposed into three subproblems: (1) designing a class-agnostic object proposal mechanism that detects both known and previously unseen objects; (2) learning a suitably general visual representation (metric) that supports recognition of the detected objects; (3) continuously updating the classifier to accommodate new object classes or training examples of existing classes. In this paper, we focus on the detection and representation learning part of the pipeline, and we ask: what does it take to learn a visual representation that allows detection and recognition of previously unseen object categories based on a single example?

We operationalize this question using an example-based visual search task (Fig. 1) that has been investigated before using handwritten characters (Omniglot; [29]) and real-world image datasets (Pascal VOC, COCO; [30, 18, 50, 10, 25]). Our central hypothesis is that scaling up the number of object categories used for training should improve the generalization capabilities of the learned representation. This hypothesis is motivated by the following observations. On (cluttered) Omniglot [29], recognition of novel characters works almost as well as for characters seen during training. In this case, sampling enough categories during training relative to the visual complexity of the objects is sufficient to learn a metric that generalizes to novel categories. In contrast, models trained on visually more complex datasets like Pascal VOC and COCO exhibit a large generalization gap: novel categories are detected much less reliably than ones seen during training. This result suggests that on the natural image datasets, the number of categories is too small given the visual complexity of the objects and the models retreat to a shortcut [12] – memorizing the training categories.

To test the hypothesis that wider datasets improve generalization, we increase the number of object categories during training by using datasets (LVIS, Objects365) that have a larger number of categories annotated. Our experiments support this hypothesis and suggest the following conclusions:

- The generalization gap between training and novel categories is a key problem in one-shot object detection.
- This generalization gap can be almost closed by increasing the number of categories used for training: going from 80 classses in COCO to 1200 in LVIS improves relative performance from 45% to 89%.
- A detailed analysis shows that number of categories, not the amount of data, is the driving force behind this effect.
- Closing the generalization gap allows using established methods from the object detection community (like e.g. stronger backbones) to improve performance on known and novel categories alike.
- We use these insights to improve state-of-the-art performance on COCO by **5.4** %AP^{50} (from 22 %AP^{50} to 27.5 %AP^{50}) using annotations from LVIS.

2 Related Work

Object detection Object detection - the task of detecting objects in complex, cluttered scenes - has seen huge progress since the widespread adoption of DNNs [13, 35, 16, 26, 6, 47, 4]. Similarly the number of datasets has grown steadily, fueled by the importance this task has for computer vision applications [9, 36, 28, 51, 32, 23, 14, 40]. However most models and datasets focus on scenarios where abundant examples per category are available.

Few-shot learning Algorithms for few-shot learning - learning a model from only a few examples - can broadly be separated into two categories: Metric learning[21, 44, 41] - learn a good representation and metric that generalizes to new data. And meta learning [11, 37] - learn a good way to learn a new task. However, recent work has shown that complex algorithmic approaches can be rivaled by improving and scaling simple approaches like transfer learning [7, 31, 8, 22].

Few-shot & one-shot object detection Recently, several groups have started to tackle few-shot learning for object detection. Two training and evaluation paradigms have emerged. The first is inspired by continual learning: incorporate a set of new categories with only a few labeled images per category into an existing classifier [20, 49, 46, 45]. The second one phrases the problem as an example-based visual search: detect objects based on a single example image [30, 18, 50, 10, 25, Fig. 1 left]. We refer to the former (continual learning) as *few-shot object detection*, since typically 10–30 images are used for experiments on COCO. In contrast, we refer to the latter (visual search) as *one-shot object detection*, since the focus is on the setting with a single example. In the present paper we work with this latter paradigm, since it focuses on the representation learning part of the problem and avoids the additional complexity of continual learning.

Methods for one-shot object detection Existing methods for one-shot object detection usually combine a standard object detection architecture with metric or meta-learning methods [2, 30, 18, 50, 10, 33, 25]. To better handle complex scenes and pose changes methods such as spatial awareness [25] or pose transforms [2, 33] have been proposed. A recent method uses a transformer to solve the matching problem [5]. We here use one of the most straightforward models, Siamese Faster R-CNN [30], to demonstrate that a change of the training data rather than the model architecture is sufficient to substantially reduce the generalization gap between known and novel categories.

Number of categories in few-shot learning Most of the few-shot learning literature focuses on developing algorithmic solutions to a set of existing small-scale benchmarks. In contrast a lot less attention has been payed to exploring new tasks or datasets. The influence of the training data was mostly observed indirectly, e.g. through better performance on datasets with more categories such as *tiered*ImageNet vs. *mini*ImageNet. We here flip the focus demonstrating that significant progress can be made by keeping the algorithm the same and only changing the training data. Concurrent studies confirm this finding that more categories help few-shot object detection [10] and few-shot image classification [38, 19]. We add to this by not only looking at few-shot performance but comparing it with performance on known categories (generalization gap). This allows us to uncover the functional relationship behind the effect (closing a shortcut).

3 Experiments

Models We mainly use Siamese Faster R-CNN, an example-based version of Faster R-CNN [35] similar to Siamese Mask R-CNN [30]. Briefly, it consists of a feature extractor, a matching step and a standard region proposal network and bounding box head (Fig. 2). The feature extractor (called backbone in object detection) is a standard ResNet-50 with feature pyramid networks [17, 26] which is applied to the image and reference with weight sharing. In the matching step the reference representation is compared to the image representation in a sliding window approach by computing a feature-wise L1 difference. The resulting similarity encoding representation is concatenated to the image representation and passed on to the region proposal network (RPN). The RPN proposes a set of bounding boxes which potentially contain objects. These boxes are then classified as containing an object from the reference class or something else (other object or background). Box coordinates are refined by bounding box regression and overlapping boxes are removed using non-maximum suppression.

We additionally developed Siamese RetinaNet, a single-stage detector based on RetinaNet [27]. The feature extraction and matching steps are identical to Siamese Faster R-CNN, but it uses the unified RetinaHead to jointly propose and classify bounding boxes. To counter the effect of too many negative samples, the classifier is trained with focal loss [27].

Training & Evaluation The example-based training is slightly different from the traditional object detection training paradigm. For each image a reference category is randomly chosen by picking a category with at least one instance in the image. A reference is retrieved by randomly selecting one instance from this category in another image and tightly cropping it. The labels for each bounding box are changed to 0 or 1 depending on whether the object is from the reference category or not. Annotations for objects from the held-out categories are removed from the dataset before training. At test time a similar procedure is chosen but instead of picking one category for each image, all categories with at least one object in the image are chosen [30] and one (1-shot) or five (5-shot) reference images are provided. Predictions are assigned their corresponding category label and evaluation is performed using standard tools and metrics.

Implementation We implemented Siamese Faster R-CNN and Siamese RetinaNet in mmdetection v1.0rc [6], which improved performance by more than 30% over the original Siamese Mask R-CNN [30, Table 4]. We keep all hyperparameters the same as in the standard Faster R-CNN implementation of mmdetection. Due to resource constraints we reduce the number of samples per epoch to 120k for Objects365.

Hyperparameters Our model is derived from mmdetection v1.0rc [6] and uses the same hyperparameters as used for Faster R-CNN and RetinaNet[1]. Please note that the default settings for Pascal VOC differ slightly from those for COCO training. We use the COCO hyperparameters for experiments on COCO, LVIS and Objects365 and Pascal VOC settings for Pascal VOC.

Datasets We use the four datasets shown in Table 1: COCO [28], Objects365 [40], LVIS [14] and Pascal VOC [9]. We use standard splits and test on the validation sets except for Pascal VOC where we test on the 2007 test set. Due to resource constraints, we evaluate Objects365 on a fixed subset of 10k images from the validation set.

Category splits Following common protocol for example-based detection [30, 39] we split the categories in each dataset into four splits using every fourth category as hold-out set and the other 3/4 categories for training. So on Pascal VOC there are 15 categories for training in each split, on COCO there are 60, on Objects365 274 and on LVIS 902. We train and test four models (one for each split)

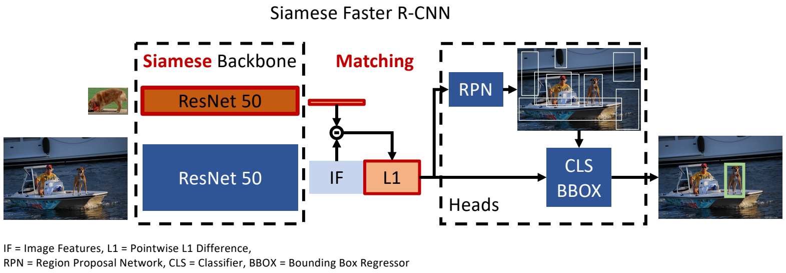

Siamese Faster R-CNN
Siamese Backbone
Matching
ResNet 50
ResNet 50
IF
L1
RPN
CLS
BBOX
Heads
IF = Image Features, L1 = Pointwise L1 Difference,
RPN = Region Proposal Network, CLS = Classifier, BBOX = Bounding Box Regressor

Figure 2: Siamese Faster R-CNN

[1] All details can be found in the respective configs: `https://github.com/open-mmlab/mmdetection/tree/5bf935e1b7621b234ddb34ef6c32b2b524243995/configs`

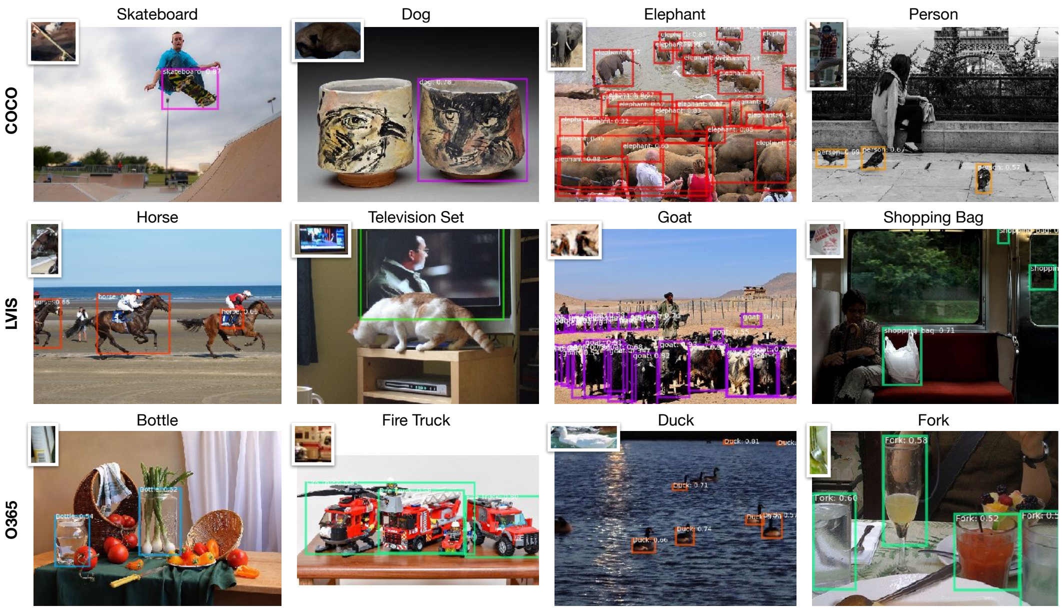


Figure 3: Example predictions on held-out categories (ResNet-50 backbone). The left three columns show success cases. The rightmost column shows failure cases in which objects are overlooked and/or wrongfully detected.

and report the mean over those four models, so performance is always measured on all categories. Computing performance in this way across all categories is preferable to using a fixed subset as some categories may be harder than others. During evaluation, the reference images are chosen randomly. We therefore run the evaluation five times, reporting the average AP^{50} over splits. The 95% confidence intervals for the average AP^{50} is below $\pm 0.2\%\text{AP}^{50}$ for all experiments.

4 Results

4.1 Generalization gap on COCO and Pascal VOC

We start by showing that objects of held-out categories are detected less reliably on COCO and Pascal VOC. On both datasets, Siamese Faster R-CNN shows strong signs of overfitting to the training categories (Fig. 4 & Table 2): despite setting a new stat-of-the-art performance is much higher than for categories held-out during training (COCO: 49.7 → 22.8 $\%\text{AP}^{50}$; Pascal VOC: 82.7 → 37.6 $\%\text{AP}^{50}$). We refer to this drop in performance as the *generalization gap*. This result is consistent with the literature: [18] – the previous state-of-the-art – report performance dropping 40.9 → 22.0 $\%\text{AP}^{50}$ on COCO

Dataset	Version	Classes	Images	Instances	Ins/Img	Cls/Img	Thr.
Pascal VOC	07+12	20	8k	23k	2.9	1.6	✓
COCO	2017	80	118k	860k	7.3	2.9	✓
LVIS	v1	1,203	100k	1.27M	≥12.8*	≥3.6*	✗
Objects365	v2	365	1.94M	28M	14.6	6.1	✓

Table 1: Dataset comparison. Thr. = Throughoutly annotated: every instance of every class is annotated in every image. *LVIS has potentially more objects and categories per image than are annotated due to the non-exhaustive labeling.

	COCO		Pascal VOC	
Categories→	Train	Held-Out	Train	Held-Out
Siam. Faster R-CNN	49.7	22.8	82.7	37.6
— empty Refs.	10.1	4.4	59.6	33.2

Table 2: On COCO and Pascal VOC there is a clear performance gap (AP^{50}) between categories used during training (Train) and held-out categories (Held-Out). A baseline getting a black image as reference which contains no information about the target category (– empty Refs.) performs surprisingly well on Pascal VOC but fails on COCO.

(see Table 4 below). Some newer models reportedly close the gap on Pascal VOC [50, 18, 25]; we will discuss Pascal VOC further in the next section. Example predictions show good localization (bounding boxes) even for unknown objects in cluttered scenes while classification errors make up the majority of mistakes (Fig. 3).

4.2 Pascal VOC is too easy to evaluate one-shot object detection models

Having identified this large generalization gap, we ask whether the models have learned a useful metric for one-shot detection at all or whether they rely on simple dataset statistics. Pascal VOC contains, on average, only 1.6 categories and 2.9 instances per image. In this case, simply detecting all foreground objects may be a viable strategy. To test how well such a trivial strategy would perform, we provide the model with uninformative references (we use all-black images). Interestingly, this baseline performs very well, achieving 59.6 $\%AP^{50}$ on training and 33.2 $\%AP^{50}$ on held-out categories (Table 2). For held-out categories, the difference to an example-based search is marginal (33.2 → 37.6 $\%AP^{50}$). This result demonstrates that on Pascal VOC the model mostly follows a shortcut and uses basic dataset statistics to solve the task.

In contrast, COCO represents a drastic increase in image complexity compared with Pascal VOC: it contains, on average, 2.9 categories and 7.3 instances per image. As expected, in this case the trivial baseline with uninformative references performs substantially worse than the example-based search (training: 49.7 → 10.1 $\%AP^{50}$; held-out: 22.8 → 4.4 $\%AP^{50}$; Table 2). Thus, the added image complexity in COCO forces the model to use the reference image for classification but the small set of categories is not sufficient to prevent memorizing the training categories.

4.3 Training on more categories reduces the generalization gap

We now turn to our main hypothesis that increasing the number of categories used during training could close the generalization gap identified above. To this end we use Objects365 and LVIS, two fairly new datasets with 365 and 1203 categories, respectively (much more than the 20/80 in Pascal VOC/COCO). Indeed, training on these wider datasets improves the relative performance on the held-out categories from 46% on COCO to 76% on Objects365 and up to 89% on LVIS (Fig. 1). In absolute numbers this means going from a 26.9 $\%AP^{50}$ gap on COCO to a 4.6 $\%AP^{50}$ gap on Objects365 and a 3.5 $\%AP^{50}$ gap on LVIS (Table 3) in the one-shot setting. Increasing the number of references to five (5-shot) improves performance on all datasets but leaves relative performance unchanged (Table 3, right columns).

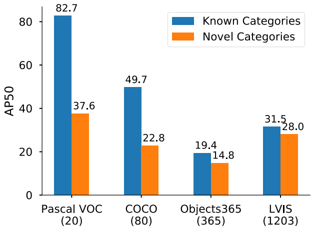


Figure 4: Performance on known and novel categories for different datasets.

This effect is not caused simply by differences between the datasets, as the following experiment shows: For both datasets (LVIS and Objects365), we train models on progressively more categories. When training on less than 100 categories (resembling training on COCO), a clear generalization gap is visible on both LVIS and Objects365 (Fig. 5A: leftmost data points). Increasing the number of training categories leads to better performance on the held-out categories, while performance on the training categories stays the same (LVIS) or decreases (Objects365). The effect is the same in the 5-shot setting but with a better baseline performance (Fig. A.1 in Appendix).

4.4 The number of categories is the crucial factor

The results so far show that increasing the number of categories used during training reduces the generalization gap and improves performance. However, this effect could also be caused by the fact that with more categories there is also more data available. Consider the situation where we train on 10% of the categories (90 in the case of LVIS). As we sample these categories uniformly from the dataset, we use only approximately 10% of the total number of instances. To control for this confound, we created training sets that match the number of instances: in this case we use only 10% of the instances in the dataset but sample them uniformly from all 900 training categories.

The results can be seen in Fig. 5B. Our example from above with 10% of the data corresponds to the leftmost datapoint in both plots. The model trained with more categories (green) clearly outperforms the model with more instances per category (blue). The same performance gap can be seen for any fraction of the data. Thus, for a given budget of instances (labels) it is better to cover more categories than to collect as many samples per category as possible.

4.5 Once the generalization gap is closed more powerful models benefit novel categories

If models indeed learn the distribution over categories, stronger models that can learn more powerful representations should perform better on known and novel categories alike. We test this hypothesis in two ways: first, by replacing the standard ResNet-50 [17] backbone with a more expressive ResNeXt-101 [48]; second, by using a three times longer training schedule.

The larger backbone does not improve performance on the held-out categories on COCO (Table 3). Instead the additional capacity is used to memorize the training categories, which is evident from the large improvement (6.7 %AP^{50}) in performance on the training categories, but only a small improvement (0.7 %AP^{50}) on the held-out categories. In contrast, on LVIS and Objects365 the gains of the bigger backbone are not confined to the training categories but apply to the one-shot setting as well. Only a small difference remains on Objects365 (3.0 %AP^{50} vs. 1.4 %AP^{50}).

Longer training schedules show the same pattern. For COCO, performance on the training categories improves while performance on held-out categories even gets a bit worse on a 3x schedule (Table 3). In

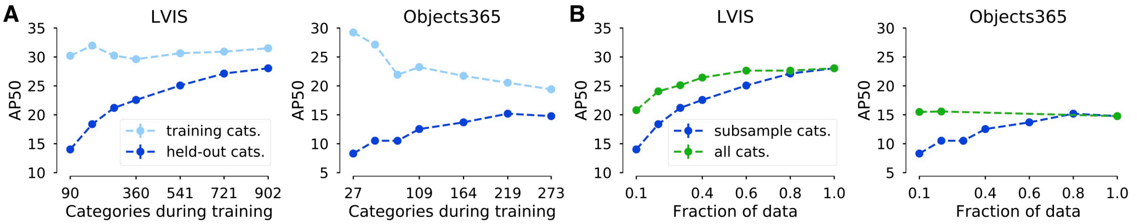


Figure 5: A. Experiment subsampling LVIS and Objects365 categories during training. When more categories are used during training performance on held-out categories (blue) improves while performance on the training categories (light blue) stays flat or decreases. **B.** Comparison of the performance on held-out categories if a fixed number of instances is chosen either from all categories (green) or from a subset of categories (blue). Having more categories is more important than having more samples per category. (1-shot results, for 5-shot see Fig. A.1)

COCO

			1-shot			5-shot		
Model	Backb.	Sched.	Train C.	Held-Out C.	Delta	Train C.	Held-Out C.	Delta
Siam. RetinaNet	R50	1x	50.6	18.9	31.7	55.5	22.1	33.4
Siam. FRCNN	R50	1x	49.7	22.8	26.9	54.9	27.6	27.3
Siam. FRCNN	R50	3x	51.7	21.9	29.8	57.6	26.7	30.9
Siam. FRCNN	X101	1x	56.4	23.5	32.9	61.9	28.6	33.3

LVIS

			1-shot			5-shot		
Model	Backb.	Sched.	Train C.	Held-Out C.	Delta	Train C.	Held-Out C.	Delta
Siam. RetinaNet	R50	1x	28.4	24.7	3.7	31.6	27.5	4.1
Siam. FRCNN	R50	1x	31.5	28.0	3.5	37.0	33.0	4.0
Siam. FRCNN	R50	3x	32.7	28.7	4.0	38.2	33.5	4.7
Siam. FRCNN	X101	1x	35.4	31.3	4.1	41.4	36.3	5.1

Objects365

			1-shot			5-shot		
Model	Backb.	Sched.	Train Cats.	Held-Out C.	Delta	Train C.	Held-Out C.	Delta
Siam. RetinaNet	R50	1x	19.7	14.5	5.2	23.4	17.2	6.2
Siam. FRCNN	R50	1x	19.4	14.8	4.6	25.7	19.9	5.8
Siam. FRCNN	R50	3x	22.0	16.5	5.5	27.7	20.9	6.8
Siam. FRCNN	X101	1x	25.0	17.9	7.1	30.6	22.4	8.2

Table 3: Effect of a three times longer training schedule and a larger backbone (ResNeXt-101 32x4d) on model performance across datasets. While larger models and longer training times lead to no or only minor improvements on held-out categories on COCO, they do have a larger effect on LVIS and Objects365.

contrast, performance on LVIS and Objects365 improves for both training and held-out categories alike, suggesting that the models do not overfit only the training categories.

4.6 Results hold for different model configurations

To test if our findings apply to single-stage detectors as well, we train and test Siamese RetinaNet on COCO, LVIS and Objects365 (Table 3). Results are very similar to Siamese Faster R-CNN. Siamese RetinaNet shows a slightly larger generalization gap on COCO (relative performance: Retina: 37% vs. FRCNN: 46%) but results are very similar on LVIS (Retina: 87% vs. FRCNN: 89%) and Objects365 (Retina: 74% vs. FRCNN: 76%).

Taken together we observe the same patterns for single- and two-stage detectors with different backbones and learning rate schedules on two datasets (Objects365 and LVIS) for 1-shot and 5-shot evaluation. This suggests that our conclusions may extend to most object detection models and we can expect to significantly boost performance using the large toolboxes which exist for traditional object detection.

4.7 State-of-the-art on COCO using LVIS

Using the insights from above, we now demonstrate state-of-the-art one-shot detection performance on COCO by training on a large number of categories. We use LVIS and create four splits which leave out all categories that have a correspondence in the respective COCO split. As LVIS is a re-annotation of COCO, this means that we expand the categories in the training set while training on the same set of images. Training with the more diverse LVIS annotations leads to a noticeable performance improvement from 22.8 to 25.0 %AP^{50}, which can be improved even further to 27.4 %AP^{50} by using the stronger ResNeXt-101 backbone, outperforming the previous best model by 5.4 %AP^{50} (Table 4). In

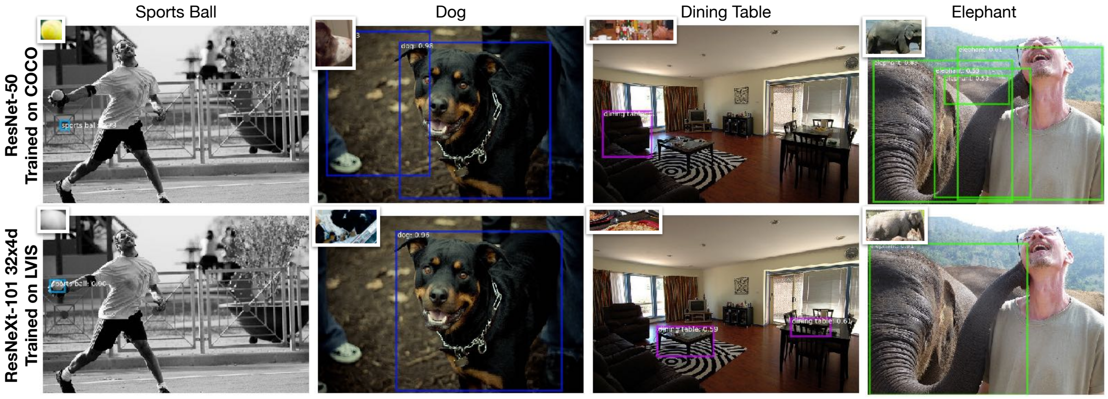


Figure 6: Predictions on COCO tend to be more accurate and cleaner when using a bigger backbone and training on LVIS. Especially on categories with more ambiguous references like sports ball or dining table the LVIS trained model is more precise. Additionally the ResNeXt backbone leads to ”cleaner” results with less false positives.

			1-shot		5-shot	
Model	Backb.	Train Data	Train C.	Held-Out C.	Train C.	Held-Out C.
Siam. Mask R-CNN*	R50	COCO	37.6	16.3	41.3	18.5
CoAE**	R50	COCO	40.9	22.0	-	-
AIT***	R50	COCO	47.5	24.3	-	-
Siam. RetinaNet	R50	COCO	50.6	18.9	55.5	22.1
Siam. Faster R-CNN	R50	COCO	49.7	22.8	54.9	27.6
Siam. Mask R-CNN	R50	COCO	51.9	22.9	57.9	27.8
Siam. Cascade R-CNN	R50	COCO	50.3	22.0	56.2	27.2
Siam. Faster R-CNN	X101 32x4d	COCO	**56.4**	23.5	**61.9**	28.6
Siam. Faster R-CNN	R50	LVIS	36.2	25.0	43.5	31.7
Siam. Faster R-CNN	X101 32x4d	LVIS	42.5	**27.4**	50.3	**34.8**

Table 4: Performance (AP^{50}) on COCO can be improved by training on LVIS. Siamese Mask R-CNN and Siamese Cascade R-CNN are identical to Siamese Faster R-CNN except for an additional mask head or cascaded bbox heads. (*[30], ** [18], *** [5])

relative terms that means going from 45% relative performance to 65%, thus substantially outperforming the previous best method (55% relative performance [18]) both in absolute and relative terms. Visual inspection of the results (Fig. 6) shows cleaner predictions with less false positives especially for difficult reference images.

5 Discussion

It has long been assumed and recently shown [38, 19, 10] that training with more categories improves few-shot learning performance. However the question whether this is due to better overall model performance or better generalization has not been answered so far. Our results show that the underlying mechanism is an improvement in generalization from 45% relative performance on COCO to 89% on LVIS. The effect is consistent for different detectors, backbone architectures and training schedules which suggests that the effect will hold for almost any model. If this trend continues with more categories object detection that generalizes to any object is within reach. This, however, does not mean that one-shot object detection is “solved”. There are at least three important steps to take:

First and foremost the performance of example-based object detectors has to improve significantly.

Our experiments outline a path forward, demonstrating that methods that profit general object detection transfer to novel categories when the generalization gap is closed. Secondly, we have to better understand the mechanisms that lead to the generalization gap. Our results indicate that one of the main reasons is a shortcut [12] - memorizing the training categories. That stronger models also perform better on novel categories with progressive closing of the gap is an indicator that the key issue was indeed overfitting. However more investigation will be required to determine which factors are important. Is it the sheer number of categories or is it their diversity, granularity, frequency? Or is the main factor semantic relationship as results from [38] and [19] suggest? Finally we have to find a way to transfer this success to smaller datasets with less categories. While we achieve a new state-of-the-art on COCO the generalization gap there (69%) is still larger than on LVIS (89%).

5.1 Future datasets should focus on the diversity of categories.

Our findings have important implications for the design of future datasets. For the goal of generalization a broader range of categories is helpful at any dataset size (Fig. 5B: green curve above blue curve at any data fraction), while from a certain point onward more examples per category lead to diminishing returns (Fig. 5B: green curve flattens out). At a time where few-shot and long-tail problems become more important in computer vision this suggest that future data collection and annotation efforts should focus more on a broad set of categories and less on the number of instances for each of those categories.

An open question is, how broad datasets have to be. Despite being a big step forward, training on LVIS still leaves a small generalization gap that widens when using stronger models. In other words: some amount of overfitting on the training categories remains. The good news is that we don't see a saturation (Fig. 5B: dark blue curves still rise at the maximum number of categories) so further increasing the number of categories should reduce the remaining gap.

5.2 The bigger picture

Our insight that applying existing methods on larger and more diverse datasets can lead to unexpected capabilities is mirrored in other areas. This phenomenon has been observed time and again and was termed the “unreasonable effectiveness of data” [15, 42] or the “bitter lesson” [43]. It played a key role in the breakthrough of DNNs thanks to ImageNet [36, 24] as well as recent results on game-play [1] or language modelling [3]. Recently [22] and [34] achieve impressive results demonstrating strong performance at one-shot and zero-shot ImageNet classification. As in our study, simple methods (transfer learning in [22] and unsupervised image captioning in [34]) on large and diverse datasets led to results that are far better than what one would have expected: Achieving ResNet performance with 10 [22] respectively zero [34] annotated samples per class in their case; 89% relative performance on LVIS in our case. We hope that by building on this insight we can soon move from trying to solve few-shot learning towards using few-shot learning to solve other problems.

References

[1] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv:1912.06680*, 2019.

[2] Sujoy Kumar Biswas and Peyman Milanfar. One shot detection with laplacian object and fast matrix cosine similarity. *TPAMI*, 2015.

[3] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv:2005.14165*, 2020.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv:2005.12872*, 2020.

[5] Ding-Jie Chen, He-Yen Hsieh, and Tyng-Luh Liu. Adaptive image transformer for one-shot object detection. In *CVPR*, 2021.

[6] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv:1906.07155*, 2019.

[7] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *ICLR*, 2019.

[8] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. *arXiv:1909.02729*, 2019.

[9] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 2010.

[10] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector. In *CVPR*, 2020.

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML*, 2017.

[12] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *arXiv:2004.07780*, 2020.

[13] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*, 2014.

[14] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019.

[15] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *Intelligent Systems*, 2009.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[18] Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. One-shot object detection with co-attention and co-excitation. In *NeurIPS*, 2019.

[19] Shuqiang Jiang, Yaohui Zhu, Chenlong Liu, Xinhang Song, Xiangyang Li, and Weiqing Min. Dataset bias in few-shot image recognition. *arXiv:2008.07960*, 2020.

[20] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. *arXiv:1812.01866*, 2018.

[21] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese Neural Networks for One-shot Image Recognition. *ICML*, 2015.

[22] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Large scale learning of general visual representations for transfer. *arXiv:1912.11370*, 2019.

[23] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Shahab Kamali, Matteo Malloci, Jordi Pont-Tuset, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://storage.googleapis.com/openimages/web/index.html*, 2017.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[25] Xiang Li, Lin Zhang, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. One-shot object detection without fine-tuning. *arXiv:2005.03819*, 2020.

[26] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *CVPR*, 2017.

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014.

[29] Claudio Michaelis, Matthias Bethge, and Alexander S. Ecker. One-Shot segmentation in clutter. *arXiv:1803.09597*, 2018.

[30] Claudio Michaelis, Ivan Ustyuzhaninov, Matthias Bethge, and Alexander S. Ecker. One-Shot instance segmentation. *arXiv:1811.11507*, 2018.

[31] Akihiro Nakamura and Tatsuya Harada. Revisiting fine-tuning for few-shot learning. *arXiv:1910.00216*, 2019.

[32] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *International Conference on Computer Vision (ICCV)*, 2017.

[33] Anton Osokin, Denis Sumin, and Vasily Lomakin. Os2d: One-stage one-shot object detection by matching anchor features. *arXiv:2003.06800*, 2020.

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv:2103.00020*, 2021.

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

[37] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv:1807.05960*, 2018.

[38] Othman Sbai, Camille Couprie, and Mathieu Aubry. Impact of base dataset design on few-shot image classification. In *ECCV*, 2020.

[39] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-Shot Learning for Semantic Segmentation. *BMVC*, 2017.

[40] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, 2019.

[41] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical Networks for Few-shot Learning. In *NIPS*, 2017.

[42] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017.

[43] Richard Sutton. The bitter lesson. *Incomplete Ideas (blog), March*, 2019.

[44] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NIPS*, 2016.

[45] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. *arXiv:2003.06957*, 2020.

[46] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Meta-learning to detect rare objects. In *ICCV*, 2019.

[47] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[48] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[49] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin. Meta r-cnn: Towards general solver for instance-level low-shot learning. In *ICCV*, 2019.

[50] Tengfei Zhang, Yue Zhang, Xian Sun, Hao Sun, Menglong Yan, Xue Yang, and Kun Fu. Comparison network for one-shot conditional object detection. *arXiv:1904.02317*, 2019.

[51] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017.

A Appendix

A.1 Additional few-shot results

We provide five-shot results for the experiments in Fig. 5 in Fig. A.1.

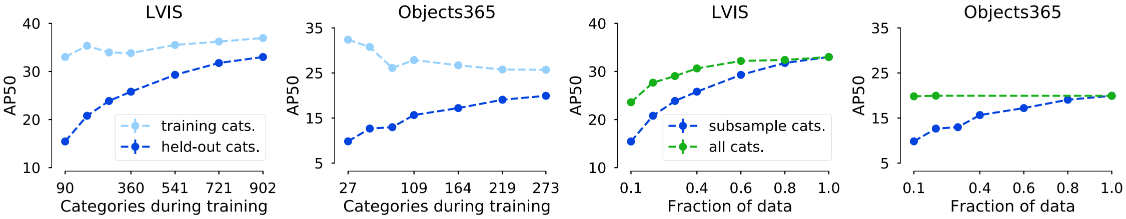


Figure A.1: Performing the experiments in Fig. 5 with five reference images (five-shot) leads to no qualitative difference.

Shortcut Learning in Deep Neural Networks

Robert Geirhos[1,2,*,§], Jörn-Henrik Jacobsen[3,*], Claudio Michaelis[1,2,*],
Richard Zemel[†,3], Wieland Brendel[†,1], Matthias Bethge[†,1] & Felix A. Wichmann[†,1]

[1] *University of Tübingen, Germany*
[2] *International Max Planck Research School for Intelligent Systems, Germany*
[3] *University of Toronto, Vector Institute, Canada*
[*] *Joint first / † joint senior authors*
[§] *To whom correspondence should be addressed:* `robert.geirhos@wichmannlab.org`

Abstract

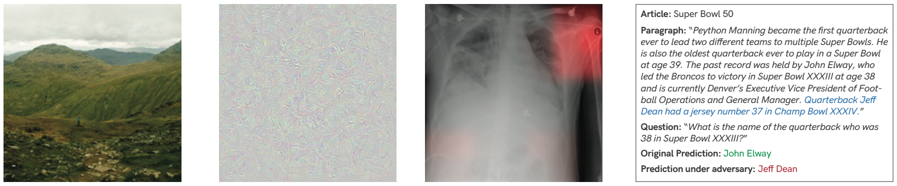
Deep learning has triggered the current rise of artificial intelligence and is the workhorse of today's machine intelligence. Numerous success stories have rapidly spread all over science, industry and society, but its limitations have only recently come into focus. In this perspective we seek to distil how many of deep learning's problem can be seen as different symptoms of the same underlying problem: *shortcut learning*. Shortcuts are decision rules that perform well on standard benchmarks but fail to transfer to more challenging testing conditions, such as real-world scenarios. Related issues are known in Comparative Psychology, Education and Linguistics, suggesting that shortcut learning may be a common characteristic of learning systems, biological and artificial alike. Based on these observations, we develop a set of recommendations for model interpretation and benchmarking, highlighting recent advances in machine learning to improve robustness and transferability from the lab to real-world applications.

1 Introduction

If science was a journey, then its destination would be the discovery of simple explanations to complex phenomena. There was a time when the existence of tides, the planet's orbit around the sun, and the observation that "things fall down" were all largely considered to be independent phenomena—until 1687, when Isaac Newton formulated his law of gravitation that provided an elegantly simple explanation to all of these (and many more). Physics has made tremendous progress over the last few centuries, but the thriving field of deep learning is still very much at the beginning of its journey—often lacking a detailed understanding of the underlying principles.

For some time, the tremendous success of deep learning has perhaps overshadowed the need to thoroughly understand the behaviour of Deep Neural Networks (DNNs). In an ever-increasing pace, DNNs were reported as having achieved human-level object classification performance [1], beating world-class human Go, Poker, and Starcraft players [2, 3],

This is the preprint version of an article that has been published by Nature Machine Intelligence (https://doi.org/10.1038/s42256-020-00257-z).



Article: Super Bowl 50

Paragraph: "*Peython Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had a jersey number 37 in Champ Bowl XXXIV.*"

Question: "*What is the name of the quarterback who was 38 in Super Bowl XXXIII?*"

Original Prediction: John Elway

Prediction under adversary: Jeff Dean

Task for DNN	Caption image	Recognise object	Recognise pneumonia	Answer question
Problem	Describes green hillside as grazing sheep	Hallucinates teapot if certain patterns are present	Fails on scans from new hospitals	Changes answer if irrelevant information is added
Shortcut	Uses background to recognise primary object	Uses features irrecognisable to humans	Looks at hospital token, not lung	Only looks at last sentence and ignores context

Figure 1. Deep neural networks often solve problems by taking shortcuts instead of learning the intended solution, leading to a lack of generalisation and unintuitive failures. This pattern can be observed in many real-world applications.

detecting cancer from X-ray scans [4], translating text across languages [5], helping combat climate change [6], and accelerating the pace of scientific progress itself [7]. Because of these successes, deep learning has gained a strong influence on our lives and society. At the same time, however, researchers are unsatisfied about the lack of a deeper understanding of the underlying principles and limitations. Different from the past, tackling this lack of understanding is not a purely scientific endeavour anymore but has become an urgent necessity due to the growing societal impact of machine learning applications. If we are to trust algorithms with our lives by being driven in an autonomous vehicle, if our job applications are to be evaluated by neural networks, if our cancer screening results are to be assessed with the help of deep learning—then we indeed need to understand thoroughly: When does deep learning work? When does it fail, and why?

In terms of understanding the limitations of deep learning, we are currently observing a large number of failure cases, some of which are visualised in Figure 1. DNNs achieve super-human performance recognising objects, but even small invisible changes [8] or a different background context [9, 10] can completely derail predictions. DNNs can generate a plausible caption for an image, but—worryingly—they can do so without ever looking at that image [11]. DNNs can accurately recognise faces, but they show high error rates for faces from minority groups [12]. DNNs can predict hiring decisions on the basis of résumés, but the algorithm's decisions are biased towards selecting men [13].

How can this discrepancy between super-human performance on one hand and astonishing failures on the other hand be reconciled? One central observation is that many failure cases are not independent phenomena, but are instead connected in the sense that DNNs follow unintended "shortcut" strategies. While superficially successful, these strategies typically fail under slightly different circumstances. For instance, a DNN may appear to classify cows perfectly well—but fails when tested on pictures where cows appear outside the typical grass landscape, revealing "grass" as an unintended (shortcut) predictor for "cow" [9]. Likewise, a language model may appear to have learned to reason—but drops to chance performance when superficial correlations are removed from the dataset [14]. Worse yet, a machine classifier successfully detected pneumonia from X-ray scans of a number of hospitals, but its performance was surprisingly low for scans from novel hospitals: The model had unexpectedly learned to identify particular hospital systems with near-perfect accuracy (e.g. by detecting a hospital-specific metal token on the scan, see Figure 1). Together with the hospital's pneumonia prevalence rate it was able to achieve a reasonably good prediction—without learning much about pneumonia at all [15].

At a principal level, shortcut learning is not a novel phenomenon. The field of machine learning with its strong mathematical underpinnings has long aspired to develop a formal understanding of shortcut learning which has led to a variety of mathematical concepts and an increasing amount of work under different terms such as *learning under covariate shift* [16], *anti-causal learning* [17], *dataset bias* [18], the *tank legend* [19] and the *Clever Hans effect* [20]. This perspective aims to present a unifying view of the various phenomena that can be collectively termed shortcuts, to describe common themes underlying them, and lay out the approaches that are being taken to address them both in theory and in practice.

The structure of this perspective is as follows. Starting from an intuitive level, we introduce shortcut learning across biological neural networks (Section 2) and then approach a more systematic level by introducing a taxonomy (Section 3) and by investigating the origins of shortcuts (Section 4). In Section 5, we highlight how these characteristics affect different areas of deep learning (Computer Vision, Natural Language Processing, Agent-based Learning, Fairness). The remainder of this perspective identifies actionable strategies towards diagnosing and understanding shortcut learning (Section 6) as well as current research directions attempting to overcome shortcut learning (Section 7). Overall, our selection of examples is biased towards Computer Vision since this is one of the areas where deep learning has had its biggest successes, and an area where examples are particularly easy to visualise. We hope that this perspective facilitates the awareness for shortcut learning and motivates new research to tackle this fundamental challenge we currently face in machine learning.

2 Shortcut learning in biological neural networks

Shortcut learning typically reveals itself by a strong discrepancy between intended and actual learning strategy, causing an unexpected failure. Interestingly, machine learning is not alone with this issue: From the way students learn to the unintended strategies rats use in behavioural experiments—variants of shortcut learning are also common for biological neural networks. We here point out two examples of unintended learning strategies by natural systems in the hope that this may provide an interesting frame of reference for thinking about shortcut learning within and beyond artificial systems.

2.1 Shortcut learning in Comparative Psychology: unintended cue learning

Rats learned to navigate a complex maze apparently based on subtle colour differences—very surprising given that the rat retina has only rudimentary machinery to support at best somewhat crude colour vision. Intensive investigation into this curious finding revealed that the rats had tricked the researchers: They did not use their visual system at all in the experiment and instead simply discriminated the colours by the odour of the colour paint used on the walls of the maze. Once smell was controlled for, the remarkable colour discrimination ability disappeared ...[1]

Animals are no strangers to finding simple, unintended solutions that fail unexpectedly: They are prone to *unintended cue learning*, as shortcut learning is called in Comparative

[1] Nicholas Rawlins, personal communication with F.A.W. some time in the early 1990s, confirmed via email on 12.11.2019.

Psychology and the Behavioural Neurosciences. When discovering cases of unintended cue learning, one typically has to acknowledge that there was a crucial difference between performance in a given experimental paradigm (e.g. rewarding rats to identify different colours) and the investigated mental ability one is actually interested in (e.g. visual colour discrimination). In analogy to machine learning, we have a striking discrepancy between intended and actual learning outcome.

2.2 Shortcut learning in Education: surface learning

Alice loves history. Always has, probably always will. At this very moment, however, she is cursing the subject: After spending weeks immersing herself in the world of Hannibal and his exploits in the Roman Empire, she is now faced with a number of exam questions that are (in her opinion) to equal parts dull and difficult. "How many elephants did Hannibal employ in his army—19, 34 or 40?" ... Alice notices that Bob, sitting in front of her, seems to be doing very well. Bob of all people, who had just boasted how he had learned the whole book chapter by rote last night ...

In educational research, Bob's reproductive learning strategy would be considered *surface learning*, an approach that relies on narrow testing conditions where simple discriminative generalisation strategies can be highly successful. This fulfils the characteristics of shortcut learning by giving the appearance of good performance but failing immediately under more general test settings. Worryingly, surface learning helps rather than hurts test performance on typical multiple-choice exams [21]: Bob is likely to receive a good grade, and judging from grades alone Bob would appear to be a much better student than Alice in spite of her focus on understanding. Thus, in analogy to machine learning we again have a striking discrepancy between intended and actual learning outcome.

3 Shortcuts defined: a taxonomy of decision rules

With examples of biological shortcut learning in mind (examples which we will return to in Section 6), what does shortcut learning in artificial neural networks look like? Figure 2 shows a simple classification problem that a neural network is trained on (distinguishing a star from a moon).[2] When testing the model on similar data (blue) the network does very well—or so it may seem. Very much like the smart rats that tricked the experimenter, the network uses a shortcut to solve the classification problem by relying on the location of stars and moons. When location is controlled for, network performance deteriorates to random guessing (red). In this case (as is typical for object recognition), classification based on object shape would have been the intended solution, even though the difference between intended and shortcut solution is not something a neural network can possibly infer from the training data.

On a general level, any neural network (or machine learning algorithm) implements a decision rule which defines a relationship between input and output—in this example assigning a category to every input image. Shortcuts, the focus of this article, are one particular group of decision rules. In order to distinguish them from other decision rules, we here introduce a taxonomy of decision rules (visualised in Figure 3), starting from a very general rule and subsequently adding more constraints until we approach the intended solution.

[2] Code is available from `https://github.com/rgeirhos/shortcut-perspective`.

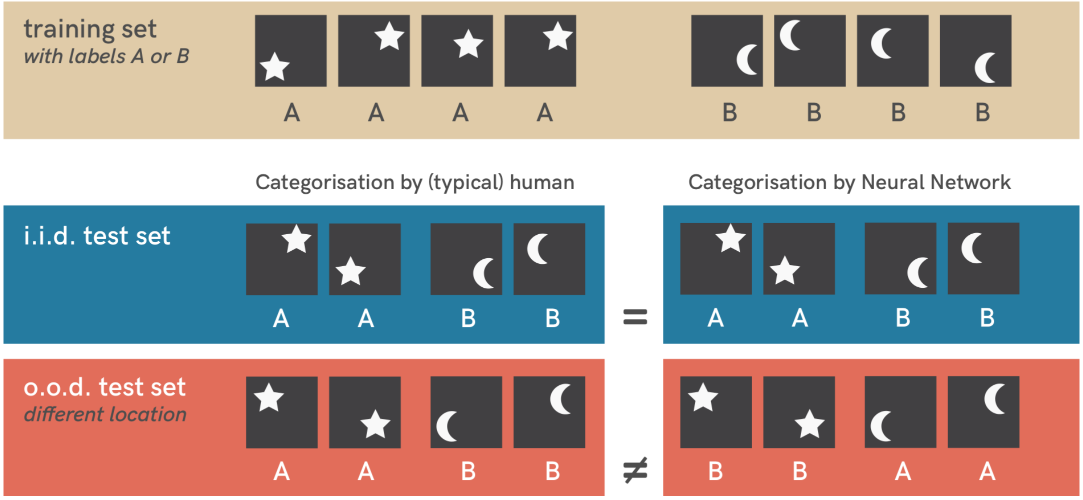


Figure 2. Toy example of shortcut learning in neural networks. When trained on a simple dataset of stars and moons (top row), a standard neural network (three layers, fully connected) can easily categorise novel similar exemplars (mathematically termed i.i.d. test set, defined later in Section 3). However, testing it on a slightly different dataset (o.o.d. test set, bottom row) reveals a shortcut strategy: The network has learned to associate object location with a category. During training, stars were always shown in the top right or bottom left of an image; moons in the top left or bottom right. This pattern is still present in samples from the i.i.d. test set (middle row) but not in o.o.d. test images (bottom row), exposing the shortcut.

(1) all possible decision rules, including non-solutions
Imagine a model that tries to solve the problem of separating stars and moons by predicting "star" every time it detects a white pixel in the image. This model uses an *uninformative feature* (the grey area in Figure 3) and does not reach good performance on the data it was trained on, since it implements a poor decision rule (both moon and star images contain white pixels). Typically, interesting problems have an abundant amount of non-solutions.

(2) training solutions, including overfitting solutions
In machine learning it is common practice to split the available data randomly into a training and a test set. The training set is used to guide the model in its selection of a (hopefully useful) decision rule, and the test set is used to check whether the model achieves good performance on similar data it has not seen before. Mathematically, the notion of similarity between training and test set commonly referred to in machine learning is the assumption that the samples in both sets are drawn from the same distribution. This is the case if both the data generation mechanism and the sampling mechanism are identical. In practice this is achieved by randomising the split between training and test set. The test set is then called independent and identically distributed (i.i.d.) with regard to the training set. In order to achieve high average performance on the test set, a model needs to learn a function that is approximately correct within a subset of the input domain which covers most of the probability of the distribution. If a function is learned that yields the correct output on the training images but not on the i.i.d. test images, the learning machine uses *overfitting features* (the blue area in Figure 3).

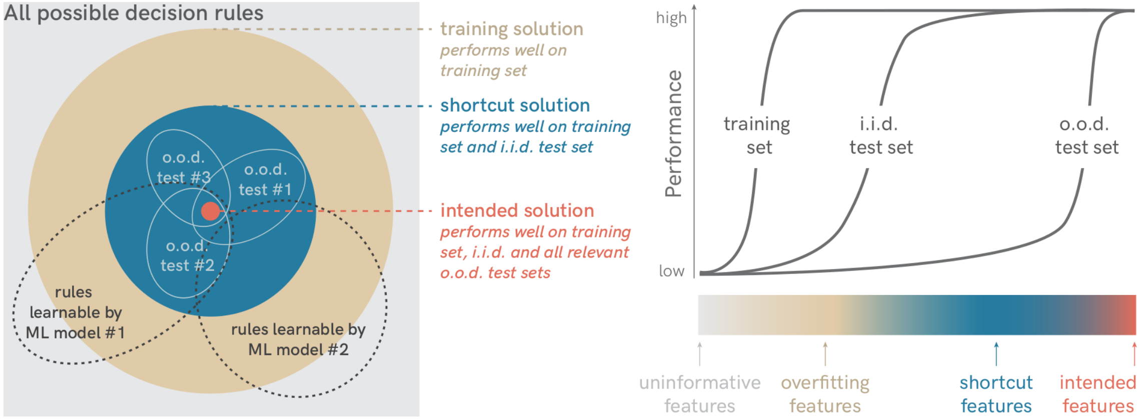

All possible decision rules
training solution
performs well on training set
shortcut solution
performs well on training set and i.i.d. test set
intended solution
performs well on training set, i.i.d. and all relevant o.o.d. test sets
o.o.d. test #3
o.o.d. test #1
o.o.d. test #2
rules learnable by ML model #1
rules learnable by ML model #2
high
Performance
low
training set
i.i.d. test set
o.o.d. test set
uninformative features
overfitting features
shortcut features
intended features

Figure 3. Taxonomy of decision rules. Among the set of all possible rules, only some solve the training data. Among the solutions that solve the training data, only some generalise to an i.i.d. test set. Among those solutions, shortcuts fail to generalise to different data (o.o.d. test sets), but the intended solution does generalise.

(3) i.i.d. test solutions, including shortcuts

Decision rules that solve both the training and i.i.d. test set typically score high on standard benchmark leaderboards. However, even the simple toy example can be solved through at least three different decision rules: (a) by shape, (b) by counting the number of white pixels (moons are smaller than stars) or (c) by location (which was correlated with object category in the training and i.i.d. test sets). As long as tests are performed only on i.i.d. data, it is impossible to distinguish between these. However, one can instead test models on datasets that are systematically different from the i.i.d. training and test data (also called *out-of-distribution* or *o.o.d.* data). For example, an o.o.d. test set with randomised object size will instantly invalidate a rule that counts white pixels. Which decision rule is the *intended solution* is clearly in the eye of the beholder, but humans often have clear expectations. In our toy example, humans typically classify by shape. A standard fully connected neural network[3] trained on this dataset, however, learns a location-based rule (see Figure 2). In this case, the network has used a *shortcut feature* (the blue area in Figure 3): a feature that helps to perform well on i.i.d. test data but fails in o.o.d. generalisation tests.

(4) intended solution

Decision rules that use the *intended features* (the red area in Figure 3) work well not only on an i.i.d. test set but also perform as intended on o.o.d. tests, where shortcut solutions fail. In the toy example, a decision rule based on object shape (the intended feature) would generalise to objects at a different location or with a different size. Humans typically have a strong intuition for what the intended solution should be capable of. Yet, for complex problems, intended solutions are mostly impossible to formalise, so machine learning is needed to estimate these solutions from examples. Therefore the choice of examples, among other aspects, influence how closely the intended solution can be approximated.

[3] A convolutional (rather than fully connected) network would be prevented from taking this shortcut by design.

4 Shortcuts: where do they come from?

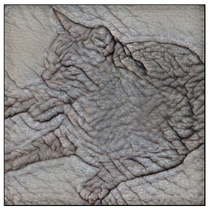
Following this taxonomy, shortcuts are decision rules that perform well on i.i.d. test data but fail on o.o.d. tests, revealing a mismatch between intended and learned solution. It is clear that shortcut learning is to be avoided, but where do shortcuts come from, and what are the defining real-world characteristics of shortcuts that one needs to look out for when assessing a model or task through the lens of shortcut learning? There are two different aspects that one needs to take into account. First, shortcut opportunities (or shortcut features) in the data: possibilities for solving a problem differently than intended (Section 4.1). Second, feature combination: how different features are combined to form a decision rule (Section 4.2). Together, these aspects determine how a model generalises (Section 4.3).

4.1 Dataset: shortcut opportunities

What makes a cow a cow? To DNNs, a familiar background can be as important for recognition as the object itself, and sometimes even more important: A cow at an unexpected location (such as a beach rather than grassland) is not classified correctly [9]. Conversely, a lush hilly landscape without any animal at all might be labelled as a "herd of grazing sheep" by a DNN [22].

This example highlights how a systematic relationship between object and background or context can easily create a shortcut opportunity. If cows happen to be on grassland for most of the training data, detecting grass instead of cows becomes a successful strategy for solving a classification problem in an unintended way; and indeed many models base their predictions on context [23, 24, 25, 26, 9, 27, 10]. Many shortcut opportunities are a consequence of natural relationships, since grazing cows are typically surrounded by grassland rather than water. These so-called *dataset biases* have long been known to be problematic for machine learning algorithms [18]. Humans, too, are influenced by contextual biases (as evident from faster reaction times when objects appear in the expected context), but their predictions are much less affected when context is missing [28, 29, 30, 31]. In addition to shortcut opportunities that are fairly easy to recognise, deep learning has led to the discovery of much more subtle shortcut features, including high-frequency patterns that are almost invisible to the human eye [32, 33]. Whether easy to recognise or hard to detect, it is becoming more and more evident that shortcut opportunities are by no means disappearing when the size of a dataset is simply scaled up by some orders of magnitude (in the hope that this is sufficient to sample the diverse world that we live in [34]). Systematic biases are still present even in "Big Data" with large volume and variety, and consequently even large real-world datasets usually contain numerous shortcut opportunities. Overall, it is quite clear that data alone rarely constrains a model sufficiently, and that data cannot replace making assumptions [35]. The totality of all assumptions that a model incorporates (such as, e.g., the choice of architecture) is called the *inductive bias* of a model and will be discussed in more detail in Section 6.3.

4.2 Decision rule: shortcuts from discriminative learning



What makes a cat a cat? To standard DNNs, the example image on the left clearly shows an elephant, not a cat. Object textures and other local structures in images are highly useful for object classification in standard datasets [36], and DNNs strongly rely on texture cues for object classification, largely ignoring global object shape [37, 38].

In many cases, relying on object textures can be sufficient to solve an object categorisation task. Obviously, however, texture is only one of many attributes that define an object. Discriminative learning differs from generative modeling by picking any feature that is sufficient to reliably discriminate on a given dataset but the learning machine has no notion of how realistic examples typically look like and how the features used for discrimination are combined with other features that define an object. In our example, using textures for object classification becomes problematic if other intended attributes (like shape) are ignored entirely. This exemplifies the importance of feature combination: the definition of an object relies on a (potentially highly non-linear) combination of information from different sources or attributes that influence a decision rule.[4] In the example of the cat with elephant texture above, a shape-agnostic decision rule that merely relies on texture properties clearly fails to capture the task of object recognition as it is understood for human vision. While the model uses an important attribute (texture) it tends to equate it with the definition of the object missing out other important attributes such as shape. Of course, being aligned with the human decision rule does not always conform to our intention. In medical or safety-critical applications, for instance, we may instead seek an improvement over human performance.

Inferring human-interpretable object attributes like shape or texture from an image requires specific nonlinear computations. In typical end-to-end discriminative learning, this again may be prone to shortcut learning. Standard DNNs do not impose any human-interpretability requirements on intermediate image representations and thus might be severely biased to the extraction of overly simplistic features which only generalise under the specific design of the particular dataset used but easily fail otherwise. Discriminative feature learning goes as far that some decision rules only depend on a single predictive pixel [39, 40, 41] while all other evidence is ignored.[5] In principle, ignoring some evidence can be beneficial. In object recognition, for example, we want the decision rule to be invariant to an object shift. However, undesirable invariance (sometimes called *excessive invariance*) is harmful and modern machine learning models can be invariant to almost all features that humans would rely on when classifying an image [41].

[4] In Cognitive Science, this process is called *cue combination*.

[5] In models of animal learning, the *blocking effect* is a related phenomenon. Once a predictive cue/feature (say, a light flash) has been associated with an outcome (e.g. food), animals sometimes fail to associate a new, equally predictive cues with the same outcome [42, 43, 44].

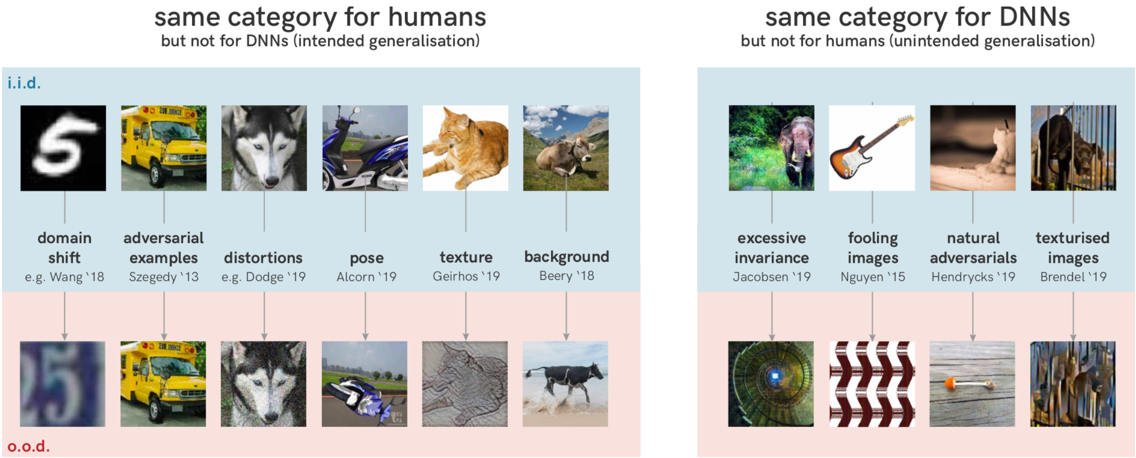
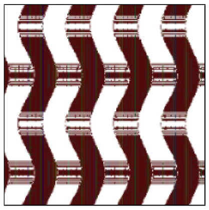


Figure 4. Both human and machine vision generalise, but they generalise very differently. Left: image pairs that belong to the same category for humans, but not for DNNs. Right: images pairs assigned to the same category by a variety of DNNs, but not by humans.

4.3 Generalisation: how shortcuts can be revealed



What makes a guitar a guitar? When tested on this pattern never seen before, standard DNNs predict "guitar" with high certainty [45]. Exposed by the generalisation test, it seems that DNNs learned to detect certain patterns (curved guitar body? strings?) instead of guitars: a successful strategy on training and i.i.d. test data that leads to unintended generalisation on o.o.d. data.
This exemplifies the inherent link between shortcut learning and generalisation. By itself, generalisation is not a part of shortcut learning—but more often than not, shortcut learning is discovered through cases of unintended generalisation, revealing a mismatch between human-intended and model-learned solution. Interestingly, DNNs do not suffer from a general lack of o.o.d. generalisation (Figure 4) [45, 36, 46, 41]. DNNs recognise guitars even if only some abstract pattern is left—however, this remarkable generalisation performance is undesired, at least in this case. In fact, the set of images that DNNs classify as "guitar" with high certainty is incredibly big. To humans only some of these look like guitars, others like patterns (interpretable or abstract) and many more resemble white noise or even look like airplanes, cats or food [8, 45, 41]. Figure 4 on the right, for example, highlights a variety of image pairs that have hardly anything in common for humans but belong to the same category for DNNs. Conversely, to the human eye an image's category is not altered by innocuous *distribution shifts* like rotating objects or adding a bit of noise, but if these changes interact with the shortcut features that DNNs are sensitive to, they completely derail neural network predictions [8, 47, 9, 48, 49, 50, 38]. This highlights that generalisation failures are neither a failure to learn nor a failure to generalise at all, but instead a failure to generalise in the intended direction—generalisation and robustness can be considered the flip side of shortcut learning. Using a certain set of features creates insensitivity towards other features. Only if the selected features are still present after a distribution shift, a model generalises o.o.d.

5 Shortcut learning across deep learning

Taken together, we have seen how shortcuts are based on dataset shortcut opportunities and discriminative feature learing that result in a failure to generalise as intended. We will now turn to specific application areas, and discover how this general pattern appears across Computer Vision, Natural Language Processing, Agent-based (Reinforcement) Learning and Fairness / algorithmic decision-making. While shortcut learning is certainly not limited to these areas, they might be the most prominent ones where the problem has been observed.

Computer Vision To humans, for example, a photograph of a car still shows the same car even when the image is slightly transformed. To DNNs, in contrast, innocuous transformations can completely change predictions. This has been reported in various cases such as shifting the image by a few pixels [47], rotating the object [49], adding a bit of random noise or blur [51, 50, 52, 53] or (as discussed earlier) by changing background [9] or texture while keeping the shape intact [38] (see Figure 4 for examples). Some key problems in Computer Vision are linked to shortcut learning. For example, transferring model performance across datasets (*domain transfer*) is challenging because models often use domain-specific shortcut features, and shortcuts limit the usefulness of unsupervised representations [54]. Furthermore, *adversarial examples* are particularly tiny changes to an input image that completely derail model predictions [8] (an example is shown in Figure 4). Invisible to the human eye, those changes modify highly predictive patterns that DNNs use to classify objects [33]. In this sense, adversarial examples—one of the most severe failure cases of neural networks—can at least partly be interpreted as a consequence of shortcut learning.

Natural Language Processing The widely used language model BERT has been found to rely on superficial cue words. For instance, it learned that within a dataset of natural language arguments, detecting the presence of "not" was sufficient to perform above chance in finding the correct line of argumentation. This strategy turned out to be very useful for drawing a conclusion without understanding the content of a sentence [14]. Natural Language Processing suffers from very similar problems as Computer Vision and other fields. Shortcut learning starts from various dataset biases such as annotation artefacts [55, 56, 57, 58]. Feature combination crucially depends on shortcut features like word length [59, 60, 14, 61], and consequently leads to a severe lack of robustness such as an inability to generalise to more challenging test conditions [62, 63, 64, 65]. Attempts like incorporating a certain degree of unsupervised training as employed in prominent language models like BERT [5] and GPT-2 [66] did not resolve the problem of shortcut learning [14].

Agent-based (Reinforcement) Learning Instead of learning how to play Tetris, an algorithm simply learned to pause the game to evade losing [67]. Systems of Agent-based Learning are usually trained using Reinforcement Learning and related approaches such as evolutionary algorithms. In both cases, designing a good reward function is crucial, since a reward function measures how close a system is to solving the problem. However, they all too often contain unexpected shortcuts that allow for so-called *reward hacking* [68]. The existence of loopholes exploited by machines that follow the letter (and not the spirit) of the reward function highlight how difficult it is to design a shortcut-free reward function [69]. Reinforcement Learning is also a widely used method in Robotics, where there is a commonly observed *generalisation* or *reality gap* between simulated training environment and real-world use case. This can be thought of as a consequence of narrow shortcut learning by adapting to specific details of the simulation. Introducing additional variation in colour, size, texture, lighting, etc. helps a lot in closing this gap [70, 71].

Fairness & algorithmic decision-making Tasked to predict strong candidates on the basis of their résumés, a hiring tool developed by Amazon was found to be biased towards preferring men. The model, trained on previous human decisions, found gender to be such a strong predictor that even removing applicant names would not help: The model always found a way around, for instance by inferring gender from all-woman college names [13]. This exemplifies how some—but not all—problems of (un)fair algorithmic decision-making are linked to shortcut learning: Once a predictive feature is found by a model, even if it is just an artifact of the dataset, the model's decision rule may depend entirely on the shortcut feature. When human biases are not only replicated, but worsened by a machine, this is referred to as *bias amplification* [72]. Other shortcut strategies include focusing on the majority group in a dataset while accepting high error rates for underrepresented groups [12, 73], which can amplify existing societal disparities and even create new ones over time [74]. In the dynamical setting a related problem is called *disparity amplification* [74], where sequential feedback loops may amplify a model's reliance on a majority group. It should be emphasised, however, that fairness is an active research area of machine learning closely related to invariance learning that might be useful to quantify and overcome biases of both machine and human decision making.

6 Diagnosing and understanding shortcut learning

Shortcut learning currently occurs across deep learning, causing machines to fail unexpectedly. Many individual elements of shortcut learning have been identified long ago by parts of the machine learning community and some have already seen substantial progress, but currently a variety of approaches are explored without a commonly accepted strategy. We here outline three actionable steps towards diagnosing and analysing shortcut learning.

6.1 Interpreting results carefully

Distinguishing datasets and underlying abilities Shortcut learning is most deceptive when gone unnoticed. The most popular benchmarks in machine learning still rely on i.i.d. testing which drags attention away from the need to verify how closely this test performance measures the *underlying ability* one is actually interested in. For example, the ImageNet dataset [75] was intended to measure the ability "object recognition", but DNNs seem to rely mostly on "counting texture patches" [36]. Likewise, instead of performing "natural language inference", some language models perform well on datasets by simply detecting correlated key words [56]. Whenever there is a discrepancy between the simplicity with which a dataset (e.g. ImageNet, SQuAD) can be solved and the complexity evoked by the high-level description of the underlying ability (e.g. object recognition, scene understanding, argument comprehension), it is important to bear in mind that a dataset is useful only for as long as it is a good proxy for the ability one is actually interested in [56, 76]. We would hardly be intrigued by reproducing human-defined labels on datasets per se (a lookup table would do just as well in this case)—it is the underlying generalisation ability that we truly intend to measure, and ultimately improve upon.

Morgan's Canon for machine learning Recall the cautionary tale of rats sniffing rather than seeing colour, described in Section 2.1. Animals often trick experimenters by solving an experimental paradigm (i.e., dataset) in an unintended way without using the underlying ability one is actually interested in. This highlights how incredibly difficult it can be for humans to imagine solving a tough challenge in any other way than the human way: Surely, at Marr's implementational level [77] there may be differences between rat and human colour discrimination. But at the algorithmic level there is often a tacit assumption that human-like performance implies human-like strategy (or algorithm) [78]. This *same strategy assumption* is paralleled by deep learning: Surely, DNN units are different from biological neurons—but if DNNs successfully recognise objects, it seems natural to assume that they are using object shape like humans do [37, 36, 38].

Comparative Psychology with its long history of comparing mental abilities across species has coined a term for the fallacy to confuse human-centered interpretations of an observed behaviour and the actual behaviour at hand (which often has a much simpler explanation): *anthropomorphism*, "the tendency of humans to attribute human-like psychological characteristics to nonhumans on the basis of insufficient empirical evidence" [79, p. 5]. As a reaction to the widespread occurrence of this fallacy, psychologist Lloyd Morgan developed a conservative guideline for interpreting non-human behaviour as early as 1903. It later became known as Morgan's Canon: "In no case is an animal activity to be interpreted in terms of higher psychological processes if it can be fairly interpreted in terms of processes which stand lower on the scale of psychological evolution and development" [80, p. 59]. Picking up on a simple correlation, for example, would be considered a process that stands low on this psychological scale whereas "understanding a scene" would be considered much higher. It has been argued that Morgan's Canon can and should be applied to interpreting machine learning results [79], and we consider it to be especially relevant in the context of shortcut learning. Accordingly, it might be worth acquiring the habit to confront machine learning models with a "Morgan's Canon for machine learning"[6]: *Never attribute to high-level abilities that which can be adequately explained by shortcut learning.*

Testing (surprisingly) strong baselines In order to find out whether a result may also be explained by shortcut learning, it can be helpful to test whether a baseline model exceeds expectations even though it does not use intended features. Examples include using nearest neighbours for scene completion and estimating geolocation [81, 82], object recognition with local features only [36], reasoning based on single cue words [59, 14] or answering questions about a movie without ever showing the movie to a model [83]. Importantly, this is not meant to imply that DNNs cannot acquire high-level abilities. They certainly do have the potential to solve complex challenges and serve as scientific models for prediction, explanation and exploration [84]—however, we must not confuse performance on a *dataset* with the acquisition of an *underlying ability*.

6.2 Detecting shortcuts: towards o.o.d. generalisation tests

Making o.o.d. generalisation tests a standard practice Currently, measuring model performance by assessing validation performance on an i.i.d. test set is at the very heart of the vast majority of machine learning benchmarks. Unfortunately, in real-world settings

[6] Our formulation is adapted from Hanlon's razor, "Never attribute to malice that which can be adequately explained by stupidity".

the i.i.d. assumption is rarely justified; in fact, this assumption has been called “the big lie in machine learning” [85]. While any metric is typically only an approximation of what we truly intend to measure, the i.i.d. performance metric may not be a good approximation as it can often be misleading, giving a false sense of security. In Section 2.2 we described how Bob gets a good grade on a multiple-choice exam through rote learning. Bob’s reproductive approach gives the superficial appearance of excellent performance, but it would not generalise to a more challenging test. Worse yet, as long as Bob continues to receive good grades through surface learning, he is unlikely to change his learning strategy.

Within the field of Education, what is the best practice to avoid surface learning? It has been argued that changing the type of examination from multiple-choice tests to essay questions discourages surface learning, and indeed surface approaches typically fail on these kinds of exams [21]. Essay questions, on the other hand, encourage so-called *deep* or *transformational* learning strategies [86, 87], like Alice’s focus on understanding. This in turn enables transferring the learned content to *novel* problems and consequently achieves a much better overlap between the educational objectives of the teacher and what the students actually learn [88]. We can easily see the connection to machine learning—transferring knowledge to novel problems corresponds to testing generalisation beyond the narrowly learned setting [89, 90, 91]. If model performance is assessed only on i.i.d. test data, then we are unable to discover whether the model is actually acquiring the ability we think it is, since exploiting shortcuts often leads to deceptively good results on standard metrics [92]. We, among many others [93, 78, 94, 95, 96], have explored a variety of o.o.d. tests and we hope it will be possible to identify a sufficiently simple and effective test procedure that could replace i.i.d. testing as a new standard method for benchmarking machine learning models in the future.

Designing good o.o.d. tests While a distribution shift (between i.i.d. and o.o.d. data) has a clear mathematical definition, it can be hard to detect in practice [101, 102]. In these cases, training a classifier to distinguish samples in dataset A from samples in dataset A’ can reveal a distribution shift. We believe that good o.o.d. tests should fullfill at least the following three conditions: First, per definition there needs to be a *clear distribution shift*, a shift that may or may not be distinguishable by humans. Second, it should have a *well-defined intended solution*. Training on natural images while testing on white noise would technically constitute an o.o.d. test but lacks a solution. Third, a good o.o.d. test is a test where the majority of *current models struggle*. Typically, the space of all conceivable o.o.d. tests includes numerous uninteresting tests. Thus given limited time and resources, one might want to focus on challenging test cases. As models evolve, generalisation benchmarks will need to evolve as well, which is exemplified by the Winograd Schema Challenge [103]. Initially designed to overcome shortcut opportunities caused by the open-ended nature of the Turing test, this common-sense reasoning benchmark was scrutinised after modern language models started to perform suspiciously well—and it indeed contained more shortcut opportunities than originally envisioned [104], highlighting the need for revised tests. Fortunately, stronger generalisation tests are beginning to gain traction across deep learning. While o.o.d. tests will likely need to evolve alongside the models they aim to evaluate, a few current encouraging examples are listed in Box I. In summary, rigorous generalisation benchmarks are crucial when distinguishing between the intended and a shortcut solution, and it would be extremely useful if a strong generally applicable testing procedure will emerge from this range of approaches.

Box I. EXAMPLES OF INTERESTING O.O.D. BENCHMARKS

We here list a few selected, encouraging examples of o.o.d. benchmarks.

Adversarial attacks can be seen as testing on model-specific worst-case o.o.d. data, which makes it an interesting diagnostic tool. If a successful adversarial attack [8] can change model predictions without changing semantic content, this is an indication that something akin to shortcut learning may be occurring [33, 97].

ARCT with removed shortcuts is a language argument comprehension dataset that follows the idea of removing known shortcut opportunities from the data itself in order to create harder test cases [14].

Cue conflict stimuli like images with conflicting texture and shape information pitch features/cues against each other, such as an intended against an unintended cue [38]. This approach can easily be compared to human responses.

ImageNet-A is a collection of natural images that several state-of-the-art models consistently classify wrongly. It thus benchmarks models on worst-case natural images [46].

ImageNet-C applies 15 different image corruptions to standard test images, an approach we find appealing for its variety and usability [52].

ObjectNet introduces the idea of scientific controls into o.o.d. benchmarking, allowing to disentangle the influence of background, rotation and viewpoint [98].

PACS and other domain generalisation datasets require extrapolation beyond i.i.d. data per design by testing on a domain different from training data (e.g. cartoon images) [99].

Shift-MNIST / biased CelebA / unfair dSprites are controlled toy datasets that introduce correlations in the training data (e.g. class-predictive pixels or image quality) and record the accuracy drop on clean test data as a way of finding out how prone a given architecture and loss function are to picking up on shortcuts [39, 40, 100, 41].

6.3 Shortcuts: why are they learned?

The "Principle of Least Effort" Why are machines so prone to learning shortcuts, detecting grass instead of cows [9] or a metal token instead of pneumonia [15]? Exploiting those shortcuts seems much *easier* for DNNs than learning the intended solution. But what determines whether a solution is easy to learn? In Linguistics, a related phenomenon is called the "Principle of Least Effort" [119], the observation that language speakers generally try to minimise the amount of effort involved in communication. For example, the use of "plane" is becoming more common than "airplane", and in pronouncing "cupboard", "p" and "b" are merged into a single sound [120, 121]. Interestingly, whether a language change makes it easier for the speaker doesn't always simply depend on objective measures like word length. On the contrary, this process is shaped by a variety of different factors, including the anatomy (architecture) of the human speech organs and previous language experience (training data).

Box II. SHORTCUT LEARNING & INDUCTIVE BIASES

The four components listed below determine the *inductive bias* of a model and dataset: the set of assumptions that influence which solutions are learnable, and how readily they can be learned. Although in theory DNNs can approximate any function (given potentially infinite capacity) [105], their inductive bias plays an important role for the types of patterns they prefer to learn given finite capacity and data.

- **Structure: architecture.** Convolutions make it harder for a model to use location—a prior [106] that is so powerful for natural images that even untrained networks can be used for tasks like image inpainting and denoising [107]. In Natural Language Processing, transformer architectures [108] use *attention layers* to understand the context by modelling relationships between words. In most cases, however, it is hard to understand the implicit priors in a DNN and even standard elements like ReLU activations can lead to unexpected effects like unwarranted confidence [109].

- **Experience: training data.** As discussed in Section 4.1, shortcut opportunities are present in most data and rarely disappear by adding more data [32, 69, 56, 38, 33]. Modifying the training data to block specific shortcuts has been demonstrated to work for reducing adversarial vulnerability [110] and texture bias [38].

- **Goal: loss function.** The most commonly used loss function for classification, *cross-entropy*, encourages DNNs to stop learning once a simple predictor is found; a modification can force neural networks to use all available information [41]. Regularisation terms that use additional information about the training data have been used to disentangle intended features from shortcut features [39, 111].

- **Learning: optimisation.** Stochastic gradient descent and its variants bias DNNs towards learning simple functions [112, 113, 114, 115]. The learning rate influences which patterns networks focus on: Large learning rates lead to learning simple patterns that are shared across examples, while small learning rates facilitate complex pattern learning and memorisation [116, 117]. The complex interactions between training method and architecture are poorly understood so far; strong claims can only be made for simple cases [118].

Understanding the influence of inductive biases In a similar vein, whether a solution is easy to learn for machines does not simply depend on the data but on all of the four components of a machine learning algorithm: architecture, training data, loss function, and optimisation. Often, the training process starts with feeding training data to the model with a fixed architecture and randomly initialised parameters. When the model's prediction is compared to ground truth, the loss function measures the prediction's quality. This supervision signal is used by an optimiser for adapting the model's internal parameters such that the model makes a better prediction the next time. Taken together, these four components (which determine the *inductive bias* of a model) influence how certain solutions are much easier to learn than others, and thus ultimately determine whether a shortcut is learned instead of the intended solution [122]. Box II provides an overview of the connections between shortcut learning and inductive biases.

7 Beyond shortcut learning

A lack of out-of-distribution generalisation can be observed all across machine learning. Consequently, a significant fraction of machine learning research is concerned with overcoming shortcut learning, albeit not necessarily as a concerted effort. Here we highlight connections between different research areas. Note that an exhaustive list would be out of the scope for this work. Instead, we cover a diverse set of approaches we find promising, each providing a unique perspective on learning beyond shortcut learning.

Domain-specific prior knowledge Avoiding reliance on unintended cues can be achieved by designing architectures and data-augmentation strategies that discourage learning shortcut features. If the orientation of an object does not matter for its category, either data-augmentation or hard-coded rotation invariance [123] can be applied. This strategy can be applied to almost any well-understood transformation of the inputs and finds its probably most general form in auto-augment as an augmentation strategy [124]. Extreme data-augmentation strategies are also the core ingredient of the most successful semi-supervised [125] and self-supervised learning approaches to date [126, 127].

Adversarial examples and robustness Adversarial attacks are a powerful analysis tool for worst-case generalisation [8]. Adversarial examples can be understood as counterfactual explanations, since they are the smallest change to an input that produces a certain output. Achieving counterfactual explanations of predictions aligned with human intention makes the ultimate goals of adversarial robustness tightly coupled to causality research in machine learning [128]. Adversarially robust models are somewhat more aligned with humans and show promising generalisation abilities [129, 130]. While adversarial attacks test model performance on model-dependent worst-case noise, a related line of research focuses on model-independent noise like image corruptions [51, 52].

Domain adaptation, -generalisation and -randomisation These areas are explicitly concerned with out-of-distribution generalisation. Usually, multiple distributions are observed during training time and the model is supposed to generalise to a new distribution at test time. Under certain assumptions the intended (or even causal) solution can be learned from multiple domains and environments [131, 39, 111]. In robotics, domain randomisation (setting certain simulation parameters randomly during training) is a very successful approach for learning policies that generalise to similar situations in the real-world [70].

Fairness Fairness research aims at making machine decisions “fair” according to a certain definition [132]. Individual fairness aims at treating similar individuals similarly while group fairness aims at treating subgroups no different than the rest of the population [133, 134]. Fairness is closely linked to generalisation and causality [135]. Sensitive group membership can be viewed as a domain indicator: Just like machine decisions should not typically be influenced by changing the domain of the data, they also should not be biased against minority groups.

Meta-learning Meta-learning seeks to learn how to learn. An intermediate goal is to learn representations that can adapt quickly to new conditions [136, 137, 138]. This ability is connected to the identification of causal graphs [139] since learning causal features allows for small changes when changing environments.

Generative modelling and disentanglement Learning to generate the observed data forces a neural network to model every variation in the training data. By itself, however, this does not necessarily lead to representations useful for downstream tasks [140], let alone out-of-distribution generalisation. Research on disentanglement addresses this shortcoming by learning generative models with well-structured latent representations [141]. The goal is to recover the true generating factors of the data distribution from observations [142] by identifying independent causal mechanisms [128].

8 Conclusion

"The road reaches every place, the short cut only one"
— James Richardson [143]

Science aims for understanding. While deep learning as an engineering discipline has seen tremendous progress over the last few years, deep learning as a scientific discipline is still lagging behind in terms of understanding the principles and limitations that govern how machines learn to extract patterns from data. A deeper understanding of how to overcome shortcut learning is of relevance beyond the current application domains of machine learning and there might be interesting future opportunities for cross-fertilisation with other disciplines such as Economics (designing management incentives that do not jeopardise long-term success by rewarding unintended "shortcut" behaviour) or Law (creating laws without "loophole" shortcut opportunities). Until the problem is solved, however, we offer the following four recommendations:

(1) Connecting the dots: shortcut learning is ubiquitous

Shortcut learning appears to be a ubiquitous characteristic of learning systems, biological and artificial alike. Many of deep learning's problems are connected through shortcut learning—models exploit dataset shortcut opportunities, select only a few predictive features instead of taking all evidence into account, and consequently suffer from unexpected generalisation failures. "Connecting the dots" between affected areas is likely to facilitate progress, and making progress can generate highly valuable impact across various applications domains.

(2) Interpreting results carefully

Discovering a shortcut often reveals the existence of an easy solution to a seemingly complex dataset. We argue that we will need to exercise great care before attributing high-level abilities like "object recognition" or "language understanding" to machines, since there is often a much simpler explanation.

(3) Testing o.o.d. generalisation

Assessing model performance on i.i.d. test data (as the majority of current benchmarks do) is insufficient to distinguish between intended and unintended (shortcut) solutions. Consequently, o.o.d. generalisation tests will need to become the rule rather than the exception.

(4) Understanding what makes a solution easy to learn

DNNs always learn the easiest possible solution to a problem, but understanding which solutions are easy (and thus likely to be learned) requires disentangling the influence of structure (architecture), experience (training data), goal (loss function) and learning (optimisation), as well as a thorough understanding of the interactions between these factors.

Shortcut learning is one of the key roadblocks towards fair, robust, deployable and trustworthy machine learning. While overcoming shortcut learning in its entirety may potentially be impossible, any progress towards mitigating it will lead to a better alignment between learned and intended solutions. This holds the promise that machines behave much more reliably in our complex and ever-changing world, even in situations far away from their training experience. Furthermore, machine decisions would become more transparent, enabling us to detect and remove biases more easily. Currently, the research on shortcut learning is still fragmented into various communities. With this perspective we hope to fuel discussions across these different communities and to initiate a movement that pushes for a new standard paradigm of generalisation that is able to replace the current i.i.d. tests.

Acknowledgement

The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting R.G. and C.M.; the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for supporting C.M. via grant EC 479/1-1; the Collaborative Research Center (Projektnummer 276693517—SFB 1233: Robust Vision) for supporting M.B. and F.A.W.; the German Federal Ministry of Education and Research through the Tübingen AI Center (FKZ 01IS18039A) for supporting W.B. and M.B.; as well as the Natural Sciences and Engineering Research Council of Canada and the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DoI/IBC) contract number D16PC00003 for supporting J.J.

The authors would like to thank Judy Borowski, Max Burg, Santiago Cadena, Alexander S. Ecker, Lisa Eisenberg, Roland Fleming, Ingo Fründ, Samuel Greiner, Florian Grießer, Shaiyan Keshvari, Ruth Kessler, David Klindt, Matthias Kümmerer, Benjamin Mitzkus, Hendrikje Nienborg, Jonas Rauber, Evgenia Rusak, Steffen Schneider, Lukas Schott, Tino Sering, Yash Sharma, Matthias Tangemann, Roland Zimmermann and Tom Wallis for helpful discussions.

Author contributions

The project was initiated by R.G. and C.M. and led by R.G. with support from C.M. and J.J.; M.B. and W.B. reshaped the initial thrust of the perspective and together with R.Z. supervised the machine learning components. The toy experiment was conducted by J.J. with input from R.G. and C.M. Most figures were designed by R.G. and W.B. with input from all other authors. Figure 2 (left) was conceived by M.B. The first draft was written by R.G., J.J. and C.M. with input from F.A.W. All authors contributed to the final version and provided critical revisions from different perspectives.

References

[1] He, K., Zhang, X., Ren, S. & Sun, J. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, 1026–1034 (2015).

[2] Silver, D. *et al.* Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484 (2016).

[3] Moravčík, M. *et al.* Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science* **356**, 508–513 (2017).

[4] Rajpurkar, P. *et al.* Chexnet: Radiologist-level pneumonia detection on chest X-rays with deep learning. *arXiv:1711.05225* (2017).

[5] Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805* (2018).

[6] Rolnick, D. *et al.* Tackling climate change with machine learning. *arXiv preprint arXiv:1906.05433* (2019).

[7] Reichstein, M. *et al.* Deep learning and process understanding for data-driven earth system science. *Nature* **566**, 195 (2019).

[8] Szegedy, C. *et al.* Intriguing properties of neural networks. *arXiv:1312.6199* (2013).

[9] Beery, S., Van Horn, G. & Perona, P. Recognition in terra incognita. In *Proceedings of the European Conference on Computer Vision*, 456–473 (2018).

[10] Rosenfeld, A., Zemel, R. & Tsotsos, J. K. The elephant in the room. *arXiv preprint arXiv:1808.03305* (2018).

[11] Heuer, H., Monz, C. & Smeulders, A. W. Generating captions without looking beyond objects. *arXiv preprint arXiv:1610.03708* (2016).

[12] Buolamwini, J. & Gebru, T. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*, 77–91 (2018).

[13] Dastin, J. Amazon scraps secret AI recruiting tool that showed bias against women. `https://reut.rs/2Od9fPr` (2018).

[14] Niven, T. & Kao, H.-Y. Probing neural network comprehension of natural language arguments. *arXiv preprint arXiv:1907.07355* (2019).

[15] Zech, J. R. *et al.* Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLoS Medicine* **15**, e1002683 (2018).

This study highlights the importance of testing model generalisation in the medical context.

[16] Bickel, S., Brückner, M. & Scheffer, T. Discriminative learning under covariate shift. *Journal of Machine Learning Research* **10**, 2137–2155 (2009).

[17] Schölkopf, B. *et al.* On causal and anticausal learning. In *International Conference on Machine Learning*, 1255–1262 ([Sl: sn], 2012).

[18] Torralba, A. & Efros, A. A. Unbiased look at dataset bias. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2011).

This study provides a comprehensive overview of dataset biases in computer vision.

[19] Branwen, G. The neural net tank urban legend. `https://www.gwern.net/Tanks` (2011).

[20] Pfungst, O. *Clever Hans: (the horse of Mr. Von Osten.) a contribution to experimental animal and human psychology* (Holt, Rinehart and Winston, 1911).

[21] Scouller, K. The influence of assessment method on students' learning approaches: Multiple choice question examination versus assignment essay. *Higher Education* **35**, 453–472 (1998).

[22] Shane, J. Do neural nets dream of electric sheep? (2018). URL `https://aiweirdness.com/post/171451900302/do-neural-nets-dream-of-electric-sheep`.

[23] Wichmann, F. A., Drewes, J., Rosas, P. & Gegenfurtner, K. R. Animal detection in natural scenes: Critical features revisited. *Journal of Vision* **10**, 6–6 (2010).

[24] Ribeiro, M. T., Singh, S. & Guestrin, C. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144 (ACM, 2016).

[25] Zhu, Z., Xie, L. & Yuille, A. L. Object recognition with and without objects. *arXiv preprint arXiv:1611.06596* (2016).

[26] Wang, J. *et al.* Visual concepts and compositional voting. *arXiv preprint arXiv:1711.04451* (2017).

[27] Dawson, M., Zisserman, A. & Nellåker, C. From same photo: Cheating on visual kinship challenges. In *Asian Conference on Computer Vision*, 654–668 (Springer, 2018).

[28] Biederman, I. *On the semantics of a glance at a scene* (Hillsdale, NJ: Erlbaum, 1981).

[29] Biederman, I., Mezzanotte, R. J. & Rabinowitz, J. C. Scene perception: Detecting and judging objects undergoing relational violations. *Cognitive Psychology* **14**, 143–177 (1982).

[30] Oliva, A. & Torralba, A. The role of context in object recognition. *Trends in Cognitive Sciences* **11**, 520–527 (2007).

[31] Castelhano, M. S. & Heaven, C. Scene context influences without scene gist: Eye movements guided by spatial associations in visual search. *Psychonomic Bulletin & Review* **18**, 890–896 (2011).

[32] Jo, J. & Bengio, Y. Measuring the tendency of CNNs to learn surface statistical regularities. *arXiv preprint arXiv:1711.11561* (2017).

[33] Ilyas, A. *et al.* Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175* (2019).

This study shows how learning imperceptible predictive features leads to adversarial vulnerability.

[34] Halevy, A., Norvig, P. & Pereira, F. The unreasonable effectiveness of data. *Intelligent Systems* (2009).

[35] Wolpert, D. H. & Macready, W. G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* **1**, 67–82 (1997).

[36] Brendel, W. & Bethge, M. Approximating CNNs with bag-of-local-features models works surprisingly well on ImageNet. In *International Conference on Learning Representations* (2019).

[37] Baker, N., Lu, H., Erlikhman, G. & Kellman, P. J. Deep convolutional networks do not classify based on global object shape. *PLoS Computational Biology* **14**, e1006613 (2018).

[38] Geirhos, R. *et al.* ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations* (2019).

This article shows how shortcut feature combination strategies are linked to distortion robustness.

[39] Heinze-Deml, C. & Meinshausen, N. Conditional variance penalties and domain shift robustness. *arXiv:1710.11469* (2017).

[40] Malhotra, G. & Bowers, J. What a difference a pixel makes: An empirical examination of features used by CNNs for categorisation. In *International Conference on Learning Representations* (2019).

[41] Jacobsen, J.-H., Behrmann, J., Zemel, R. & Bethge, M. Excessive invariance causes adversarial vulnerability. In *International Conference on Learning Representations* (2019).

[42] Kamin, L. J. Predictability, surprise, attention, and conditioning. *Punishment and aversive behavior* 279–96 (1969).

[43] Dickinson, A. *Contemporary animal learning theory*, vol. 1 (CUP Archive, 1980).

[44] Bouton, M. E. *Learning and behavior: A contemporary synthesis.* (Sinauer Associates, 2007).

[45] Nguyen, A., Yosinski, J. & Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 427–436 (IEEE, 2015).

[46] Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J. & Song, D. Natural adversarial examples. arXiv preprint arXiv:1907.07174 (2019).

[47] Azulay, A. & Weiss, Y. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv:1805.12177* (2018).

[48] Wang, M. & Deng, W. Deep visual domain adaptation: A survey. *Neurocomputing* **312**, 135–153 (2018).

[49] Alcorn, M. A. *et al.* Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019).

[50] Dodge, S. & Karam, L. Human and DNN classification performance on images with quality distortions: A comparative study. *ACM Transactions on Applied Perception (TAP)* **16**, 7 (2019).

[51] Geirhos, R. *et al.* Generalisation in humans and deep neural networks. In *Advances in Neural Information Processing Systems* (2018).

[52] Hendrycks, D. & Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations* (2019).

[53] Michaelis, C. *et al.* Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484* (2019).

[54] Minderer, M., Bachem, O., Houlsby, N. & Tschannen, M. Automatic shortcut removal for self-supervised representation learning. *arXiv preprint arXiv:2002.08822* (2020).

[55] Goyal, Y., Khot, T., Summers-Stay, D., Batra, D. & Parikh, D. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6904–6913 (2017).

[56] Gururangan, S. *et al.* Annotation artifacts in Natural Language Inference data. *arXiv preprint arXiv:1803.02324* (2018).

This article highlights how Natural Language Inference models learn heuristics that exploit superficial cues.

[57] Kaushik, D. & Lipton, Z. C. How much reading does reading comprehension require? A critical investigation of popular benchmarks. *arXiv preprint arXiv:1808.04926* (2018).

[58] Geva, M., Goldberg, Y. & Berant, J. Are we modeling the task or the annotator? An investigation of annotator bias in natural language understanding datasets. *arXiv preprint arXiv:1908.07898* (2019).

[59] Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R. & Van Durme, B. Hypothesis only baselines in Natural Language Inference. *arXiv preprint arXiv:1805.01042* (2018).

[60] Kavumba, P. *et al.* When choosing plausible alternatives, Clever Hans can be clever. *arXiv preprint arXiv:1911.00225* (2019).

[61] McCoy, R. T., Pavlick, E. & Linzen, T. Right for the wrong reasons: Diagnosing syntactic heuristics in Natural Language Inference. *arXiv preprint arXiv:1902.01007* (2019).

[62] Agrawal, A., Batra, D. & Parikh, D. Analyzing the behavior of visual question answering models. *arXiv preprint arXiv:1606.07356* (2016).

[63] Belinkov, Y. & Bisk, Y. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173* (2017).

[64] Jia, R. & Liang, P. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328* (2017).

[65] Glockner, M., Shwartz, V. & Goldberg, Y. Breaking NLI systems with sentences that require simple lexical inferences. *arXiv preprint arXiv:1805.02266* (2018).

[66] Radford, A. *et al.* Language models are unsupervised multitask learners. *OpenAI Blog* **1** (2019).

[67] Murphy VII, T. The first level of Super Mario Bros. is easy with lexicographic orderings and time travel. *The Association for Computational Heresy (SIGBOVIK) 2013* 112 (2013).

[68] Amodei, D. *et al.* Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565* (2016).

[69] Lehman, J. *et al.* The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities. *arXiv preprint arXiv:1803.03453* (2018).

This paper provides a comprehensive collection of anecdotes about shortcut learning / reward hacking in Reinforcement Learning and beyond.

[70] Tobin, J. *et al.* Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 23–30 (IEEE, 2017).

[71] Akkaya, I. *et al.* Solving Rubik's Cube with a robot hand. *arXiv:1910.07113* (2019).

[72] Zhao, J., Wang, T., Yatskar, M., Ordonez, V. & Chang, K.-W. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *arXiv preprint arXiv:1707.09457* (2017).

This study shows how algorithms amplify social biases to boost performance.

[73] Rich, A. S. & Gureckis, T. M. Lessons for artificial intelligence from the study of natural stupidity. *Nature Machine Intelligence* **1**, 174 (2019).

[74] Hashimoto, T. B., Srivastava, M., Namkoong, H. & Liang, P. Fairness without demographics in repeated loss minimization. *arXiv preprint arXiv:1806.08010* (2018).

[75] Russakovsky, O. *et al.* ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* **115**, 211–252 (2015).

[76] Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A. & Choi, Y. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830* (2019).

[77] Marr, D. *Vision: A computational investigation into the human representation and processing of visual information* (W.H. Freeman and Company, San Francisco, 1982).

[78] Borowski, J. *et al.* The notorious difficulty of comparing human and machine perception. In *NeurIPS Shared Visual Representations in Human and Machine Intelligence Workshop* (2019).

The case studies presented in this article highlight the difficulty of interpreting machine behaviour in the presence of shortcut learning.

[79] Buckner, C. The Comparative Psychology of Artificial Intelligences (2019). URL `http://philsci-archive.pitt.edu/16034/`.

This opinionated article points out important caveats when comparing human to machine intelligence.

[80] Morgan, C. L. Introduction to Comparative Psychology. (rev. ed.). *New York: Scribner* (1903).

[81] Hays, J. & Efros, A. A. Scene completion using millions of photographs. *ACM Transactions on Graphics (TOG)* **26**, 4 (2007).

[82] Hays, J. & Efros, A. A. IM2GPS: estimating geographic information from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–8 (IEEE, 2008).

[83] Jasani, B., Girdhar, R. & Ramanan, D. Are we asking the right questions in MovieQA? In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 0–0 (2019).

[84] Cichy, R. M. & Kaiser, D. Deep neural networks as scientific models. *Trends in Cognitive Sciences* (2019).

[85] Ghahramani, Z. Panel of workshop on advances in Approximate Bayesian Inference (AABI) 2017 (2017). URL `https://www.youtube.com/watch?v=x1UByHT60mQ&feature=youtu.be&t=37m44s`.

[86] Marton, F. & Säaljö, R. On qualitative differences in learning—II Outcome as a function of the learner's conception of the task. *British Journal of Educational Psychology* **46**, 115–127 (1976).

[87] Biggs, J. Individual differences in study processes and the quality of learning outcomes. *Higher Education* **8**, 381–394 (1979).

[88] Chin, C. & Brown, D. E. Learning in science: A comparison of deep and surface approaches. *Journal of Research in Science Teaching* **37**, 109–138 (2000).

This article from the field of Education reflects upon ways to achieve a better overlap between educational objectives and the way students learn.

[89] Marcus, G. F. Rethinking eliminative connectionism. *Cognitive Psychology* **37**, 243–282 (1998).

[90] Kilbertus, N., Parascandolo, G. & Schölkopf, B. Generalization in anti-causal learning. *arXiv preprint arXiv:1812.00524* (2018).

[91] Marcus, G. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631* (2018).

[92] Lapuschkin, S. *et al.* Unmasking Clever Hans predictors and assessing what machines really learn. *Nature Communications* **10**, 1096 (2019).

This study highlights how shortcut learning can lead to deceptively good results on standard metrics.

[93] Lake, B. M., Ullman, T. D., Tenenbaum, J. B. & Gershman, S. J. Building machines that learn and think like people. arXiv preprint arXiv:1604.00289 (2016).

[94] Chollet, F. The measure of intelligence. *arXiv preprint arXiv:1911.01547* (2019).

[95] Crosby, M., Beyret, B. & Halina, M. The Animal-AI Olympics. *Nature Machine Intelligence* **1**, 257–257 (2019).

[96] Juliani, A. *et al.* Obstacle tower: A generalization challenge in vision, control, and planning. *arXiv preprint arXiv:1902.01378* (2019).

[97] Engstrom, L. *et al.* A discussion of 'adversarial examples are not bugs, they are features'. *Distill* (2019).

[98] Barbu, A. *et al.* ObjectNet: a large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Advances in Neural Information Processing Systems*, 9448–9458 (2019).

[99] Li, D., Yang, Y., Song, Y.-Z. & Hospedales, T. M. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 5542–5550 (2017).

[100] Creager, E. *et al.* Flexibly fair representation learning by disentanglement. *arXiv preprint arXiv:1906.02589* (2019).

[101] Recht, B., Roelofs, R., Schmidt, L. & Shankar, V. Do CIFAR-10 classifiers generalize to CIFAR-10? *arXiv preprint arXiv:1806.00451* (2018).

[102] Recht, B., Roelofs, R., Schmidt, L. & Shankar, V. Do ImageNet classifiers generalize to ImageNet? *arXiv preprint arXiv:1902.10811* (2019).

[103] Levesque, H., Davis, E. & Morgenstern, L. The Winograd Schema Challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning* (2012).

[104] Trichelair, P., Emami, A., Trischler, A., Suleman, K. & Cheung, J. C. K. How reasonable are common-sense reasoning tasks: A case-study on the Winograd Schema Challenge and SWAG. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3373–3378 (2019).

[105] Hornik, K., Stinchcombe, M. & White, H. Multilayer feedforward networks are universal approximators. *Neural Networks* **2**, 359–366 (1989).

[106] d'Ascoli, S., Sagun, L., Bruna, J. & Biroli, G. Finding the needle in the haystack with convolutions: On the benefits of architectural bias. *arXiv preprint arXiv:1906.06766* (2019).

[107] Ulyanov, D., Vedaldi, A. & Lempitsky, V. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9446–9454 (2018).

[108] Vaswani, A. *et al.* Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008 (2017).

[109] Hein, M., Andriushchenko, M. & Bitterwolf, J. Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 41–50 (2019).

[110] Madry, A., Makelov, A., Schmidt, L., Tsipras, D. & Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations* (2018).

[111] Arjovsky, M., Bottou, L., Gulrajani, I. & Lopez-Paz, D. Invariant risk minimization. *arXiv preprint arXiv:1907.02893* (2019).

[112] Wu, L., Zhu, Z. & E, W. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239* (2017).

[113] De Palma, G., Kiani, B. T. & Lloyd, S. Deep neural networks are biased towards simple functions. *arXiv preprint arXiv:1812.10156* (2018).

[114] Valle-Perez, G., Camargo, C. Q. & Louis, A. A. Deep learning generalizes because the parameter-function map is biased towards simple functions. In *International Conference on Learning Representations* (2019).

[115] Sun, K. & Nielsen, F. Lightlike neuromanifolds, Occam's Razor and deep learning. *arXiv preprint arXiv:1905.11027* (2019).

[116] Arpit, D. *et al.* A closer look at memorization in deep networks. In *International Conference on Machine Learning* (2017).

[117] Li, Y., Wei, C. & Ma, T. Towards explaining the regularization effect of initial large learning rate in training neural networks. *arXiv preprint arXiv:1907.04595* (2019).

[118] Bartlett, P. L., Long, P. M., Lugosi, G. & Tsigler, A. Benign overfitting in linear regression. *arXiv preprint arXiv:1906.11300* (2019).

[119] Zipf, G. K. *Human Behavior and the Principle of Least Effort* (Addison-Wesley press, 1949).

[120] Ohala, J. J. The phonetics and phonology of aspects of assimilation. *Papers in Laboratory Phonology* **1**, 258–275 (1990).

[121] Vicentini, A. The economy principle in language. *Notes and Observations from early modern English grammars. Mots, Palabras, Words* **3**, 37–57 (2003).

[122] Sinz, F. H., Pitkow, X., Reimer, J., Bethge, M. & Tolias, A. S. Engineering a less artificial intelligence. *Neuron* **103**, 967–979 (2019).

[123] Cohen, T. & Welling, M. Group equivariant convolutional networks. In *International Conference on Machine Learning*, 2990–2999 (2016).

[124] Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V. & Le, Q. V. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 113–123 (2019).

[125] Berthelot, D. *et al.* Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249* (2019).

[126] Hjelm, R. D. *et al.* Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670* (2018).

[127] Oord, A. v. d., Li, Y. & Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[128] Schölkopf, B. Causality for machine learning. *arXiv preprint arXiv:1911.10500* (2019).

[129] Schott, L., Rauber, J., Bethge, M. & Brendel, W. Towards the first adversarially robust neural network model on MNIST. In *International Conference on Learning Representations* (2019).

[130] Engstrom, L. *et al.* Learning perceptually-aligned representations via adversarial robustness. *arXiv:1906.00945* (2019).

[131] Peters, J., Bühlmann, P. & Meinshausen, N. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **78**, 947–1012 (2016).

[132] Dwork, C., Hardt, M., Pitassi, T., Reingold, O. & Zemel, R. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, 214–226 (2012).

[133] Zemel, R., Wu, Y., Swersky, K., Pitassi, T. & Dwork, C. Learning fair representations. In *International Conference on Machine Learning*, 325–333 (2013).

[134] Hardt, M., Price, E. & Srebro, N. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, 3315–3323 (2016).

[135] Kusner, M. J., Loftus, J., Russell, C. & Silva, R. Counterfactual fairness. In *Advances in Neural Information Processing Systems*, 4066–4076 (2017).

[136] Schmidhuber, J. Evolutionary principles in self-referential learning. *On learning how to learn: The meta-meta-... hook.) Diploma thesis, Institut f. Informatik, Tech. Univ. Munich* **1**, 2 (1987).

[137] Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D. & Lillicrap, T. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning*, 1842–1850 (2016).

[138] Finn, C., Abbeel, P. & Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning* (2017).

[139] Bengio, Y. *et al.* A meta-transfer objective for learning to disentangle causal mechanisms. *arXiv preprint arXiv:1901.10912* (2019).

[140] Fetaya, E., Jacobsen, J.-H., Grathwohl, W. & Zemel, R. Understanding the limitations of conditional generative models. In *International Conference on Learning Representations* (2020).

[141] Higgins, I. *et al.* Beta-VAE: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations* (2017).

[142] Hyvärinen, A. & Oja, E. Independent component analysis: algorithms and applications. *Neural Networks* **13**, 411–430 (2000).

[143] Richardson, J. *Vectors: aphorisms & ten-second essays* (Ausable Press, 2001).

Appendix

A Toy example: method details

The code to reproduce our toy example (Figure 2) is available from `https://github.com/rgeirhos/shortcut-perspective`. Two easily distinguishable shapes (star and moon) were placed on a 200×200 dimensional 2D canvas. The training set is constructed out of 4000 images, where 2000 contain a star shape and 2000 a moon shape. The star shape is randomly placed in the top right and bottom left quarters of the canvas, whereas the moon shape is randomly placed in the top left and bottom right quarters of the canvas. At test time the setup is nearly identical, 1000 images with a star and 1000 images with a moon are presented. However, this time the position of star and moon shapes are randomised over the full canvas, i.e. in test images stars and moons can appear at any location.

We train two classifiers on this dataset: a fully connected network as well as a convolutional network. The classifiers are trained for five epochs with a batch size of 100 on the training set and evaluated on the test set. The training objective is standard crossentropy loss and the optimizer is Adam with a learning rate of 0.00001, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 1e-08$. The fully connected network was a three-layer ReLU MLP (multilayer perceptron) with 1024 units in each layer and two output units corresponding to the two target classes. It reaches 100% accuracy at training time and approximately chance-level accuracy at test time (51.0%). The convolutional network had three convolutional layers with 128 channels, a stride of 2 and filter size of 5×5 interleaved with ReLU nonlinearities, followed by a global average pooling and a linear layer mapping the 128 outputs to the logits. It reaches 100% accuracy on train and test set.

B Image rights & attribution

Figure 1 consists of four images from different sources. The first image from the left was taken from `https://aiweirdness.com/post/171451900302/do-neural-nets-dream-of-electric-sheep` with permission of the author. The second image from the left was generated by ourselves. The third image from the left is from ref. [15]. It was released under the CC BY 4.0 license as stated here: `https://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.1002683` and adapted by us from Figure 2B of the corresponding publication. The image on the right is Figure 1 from ref. [64]. It was released under CC BY 4.0 license as stated here: `https://www.aclweb.org/anthology/D17-1215/`(at the bottom) and retrieved by us from .

The image from Section 4.1 was adapted from Figure 1 of ref. [9] with permission from the authors (image cropped from original figure by us). The image from Section 4.2 was adapted from Figure 1 of ref. [38] with permission from the authors (image cropped from original figure by us). The image from Section 4.3 was adapted from Figure 1 of ref. [45] with permission from the authors (image cropped from original figure by us).

Figure 4 consists of a number of images from different sources. The first author of the corresponding publication is mentioned in the figure for identification. The images from ref. [8] were released under the CC BY 3.0 license as stated here: `https://arxiv.org/abs/1312.6199` and adapted by us from Figure 5a of the corresponding publication (images cropped from original figure by us). The images from ref. [50] were adapted from Figure 1 of the corresponding paper with permission from the authors (images cropped from original figure by us). The images from ref. [49] were adapted from Figure 1 of the corresponding paper with permission from the authors (images cropped from original figure by us). The images from ref. [38] were adapted from Figure 1 of the corresponding paper with permission from the authors (images cropped from original figure by us). The images from ref. [41] were adapted from Figure 1 of the corresponding paper with permission from the authors (images cropped from original figure by us). The images from ref. [36] were adapted from Figure 5 of the corresponding paper with permission from the authors (images cropped from original figure by us). The images from ref. [9] were adapted from Figure 1 of the corresponding paper with permission from the authors (images cropped from original figure by us). The images from ref. [45] were adapted from Figure 1 and Figure 2 of the corresponding paper with permission from the authors (images cropped from original figures by us).