ORBIT: A Real-World Few-Shot Dataset for Teachable Object Recognition

Daniela Massiceti[1] Luisa Zintgraf[2] John Bronskill[3] Lida Theodorou[4]

Matthew Tobias Harris[4] Edward Cutrell[1] Cecily Morrison[1] Katja Hofmann[1] Simone Stumpf[4]

[1]*Microsoft Research* [2]*University of Oxford* [3]*University of Cambridge* [4]*City, University of London*

Abstract

Object recognition has made great advances in the last decade, but predominately still relies on many high-quality training examples per object category. In contrast, learning new objects from only a few examples could enable many impactful applications from robotics to user personalization. Most few-shot learning research, however, has been driven by benchmark datasets that lack the high variation that these applications will face when deployed in the real-world. To close this gap, we present the ORBIT dataset and benchmark, grounded in the real-world application of teachable object recognizers for people who are blind/low-vision. The dataset contains 3,822 videos of 486 objects recorded by people who are blind/low-vision on their mobile phones. The benchmark reflects a realistic, highly challenging recognition problem, providing a rich playground to drive research in robustness to few-shot, high-variation conditions. We set the benchmark's first state-of-the-art and show there is massive scope for further innovation, holding the potential to impact a broad range of real-world vision applications including tools for the blind/low-vision community. We release the dataset at https://doi.org/10.25383/city.14294597 and benchmark code at https://github.com/microsoft/ORBIT-Dataset.

1. Introduction

Object recognition systems have made spectacular advances in recent years [42, 47, 43, 37, 14, 30, 36] however, most systems still rely on training datasets with 100s to 1,000s of high-quality, labeled examples per object category. These demands make training datasets expensive to collect, and limit their use to all but a few application areas.

Few-shot learning aims to reduce these demands by training models to recognize completely novel objects from only a few examples [9, 49, 40, 2, 38, 11, 46]. This will enable recognition systems that can adapt in real-world, dynamic scenarios, from self-driving cars to applications where users provide the training examples themselves. Meta-learning algorithms which "learn to learn" [45, 9, 49, 11] hold particular promise toward this goal with recent advances opening exciting possibilities for light-weight, adaptable recognition.

(a) Frames from clean videos

(b) Frames from clutter videos

Figure 1: High-variation examples in the ORBIT dataset – a facemask, hairbrush, keys, and watering can. Full videos in the supplementary material. Further examples in Figure A.5.

Most few-shot learning research, however, has been driven by datasets that lack the high variation — in number of examples per object and quality of those examples (framing, blur, etc.; see Table 1) — that recognition systems will likely face when deployed in the real-world. Key datasets such as Omniglot [23, 49] and *mini*ImageNet [49], for example, present highly structured benchmark tasks which assume a fixed number of objects and training examples per object. Meta-Dataset [48], another key dataset, poses a more challenging benchmark task of adapting to novel *datasets* given a small (random) number of training examples. Its constituent datasets [23, 17, 39, 26, 32, 50, 6], however, mirror the high-quality images of Omniglot and *mini*ImageNet, leaving robustness to the noisy frames that would be streamed from a real-world system unaddressed. While these datasets have catalyzed research in few-shot learning, state-of-the-art performance is now relatively saturated and leaves reduced scope for algorithmic innovation [16, 4, 33].

To drive further innovation in few-shot learning for real-world impact, there is a strong need for datasets that capture the high variation inherent in real-world applications. We

motivate that both the dataset and benchmark task should be grounded in a potential real-world application to bring real-world recognition challenges to life in their entirety. An application area that neatly encapsulates a few-shot, high-variation scenario are teachable object recognisers (TORs) for people who are blind/low-vision [24, 18]. Here, a user can customize an object recognizer by capturing a small number of (high-variation) training examples of essential objects on their mobile phone. The recognizer is then trained (in deployment) on these examples such that it can recognize the user's objects in novel scenarios. As a result, TORs capture a microcosm of highly challenging and realistic conditions that can be used to drive research in real-world recognition tasks, with the potential to impact a broad range of applications beyond just tools for the blind/low-vision community.

We introduce the ORBIT dataset [31], a collection of videos recorded by people who are blind/low-vision on their mobile phones, and an associated few-shot benchmark grounded in TORs. Both were designed in collaboration with a team of machine learning (ML), human-computer interaction, and accessibility researchers, and will enable the ML community to 1) accelerate research in few-shot, high-variation object recognition, and 2) explore new research directions in few-shot *video* recognition. We intend both as a rich playground to drive research in robustness to challenging, real-world conditions, a step beyond what curated few-shot datasets and structured benchmark tasks can offer, and to ultimately impact a broad range of real-world vision applications. In summary, our contributions are:

1. ORBIT benchmark dataset. The ORBIT benchmark dataset [31] (Section 3) is a collection of 3822 videos of 486 objects recorded by 77 blind/low-vision people on their mobile phones and can be downloaded at https://doi.org/10.25383/city.14294597. Examples are shown in Figures 1 and A.5. Unlike existing datasets [39, 8, 26, 49, 48], ORBIT show objects in a wide range of realistic conditions, including when objects are poorly framed, occluded by hands and other objects, blurred, and in a wide variation of backgrounds, lighting, and object orientations.

2. ORBIT teachable object recognition benchmark. We formulate a few-shot benchmark on the ORBIT dataset (Section 4) that is grounded in TORs for people who are blind/low-vision. Contrasting existing few-shot (and other) works, the benchmark proposes a novel user-centric formulation which measures personalization to individual users. It also incorporates metrics that reflect the potential computational cost of real-world deployment on a mobile device. These and the benchmark's other metrics are specifically designed to drive innovation for realistic settings.

3. State-of-the-art (SOTA) on the ORBIT benchmark. We implement 4 few-shot learning models that cover the main classes of approach in the field, extend them to videos, and establish the first SOTA on the ORBIT benchmark (Section 5). We also perform empirical studies showing that training on existing few-shot learning datasets is *not* sufficient for good performance on the ORBIT benchmark (Table 4) leaving significant scope for algorithmic innovation in few-shot techniques that can handle high-variation data.

Code for loading the dataset, computing benchmark metrics, and running the baselines is available at https://github.com/microsoft/ORBIT-Dataset.

2. Related Work

Few-shot learning datasets. Omniglot [23, 49], *mini*ImageNet [49], and Meta-Dataset [48] have driven recent progress in few-shot learning. Impressive gains have been achieved on Omniglot and *mini*ImageNet [49, 16, 4, 33], however results are now largely saturated and highly depend on the selected feature embedding. Meta-Dataset, a dataset of 10 datasets, formulates a more challenging task where whole *datasets* are held-out, but these datasets contain simple and clean images, such as clipart drawings of characters/symbols [23, 49, 17], and ImageNet-like images [26, 39, 32, 50, 6] showing objects in uniform lighting, orientations, and camera viewpoints. The ORBIT dataset and benchmark presents a more challenging few-shot task with high-variation examples captured in real-world scenarios.

High-variation datasets. Datasets captured by users in real-world settings are naturally high-variation [1, 12, 7, 21, 27, 18, 41, 13], but none collected thus far explicitly target few-shot object recognition. *ObjectNet* [1] is a test-only dataset of challenging images (e.g. unusual orientations/backgrounds) for "many-shot" classification. *Something-Something* [12] and *EPIC-Kitchens* [7] are video datasets collected by users with mobile and head-mounted cameras, respectively, but are focused on action recognition based on many examples and "action captions". *Core50* [27] is a video dataset captured on mobile phones for a continual learning recognition task. In contrast to ORBIT, the videos are high quality (captured by sighted people, with well-lit centered objects). Other high-variation datasets include those collected by people who are blind/low-vision [18, 41, 13] (see *IncluSet* for a repository of accessibility datasets [19]) however, most are not appropriate for few-shot learning. *TeGO* [18] contains mobile phone images of 19 objects taken by only 2 users (1 sighted, 1 blind) in 2 environments (1 uniform background, 1 cluttered scene). It validates the TOR use-case, but is too small to deliver a robust, deployable system. *VizWiz* [13], although larger scale (31,173 mobile phone images contributed by 11,045 blind/low-vision users) targets image captioning and question-answering tasks, and is not annotated with object labels. The ORBIT dataset and benchmark is motivated by the lack of datasets that have the scale and structure required for few-shot, high-variation real-world applications, and adds to the growing repository of datasets for accessibility.

		Omniglot [23]	*mini*ImageNet [49]	Meta-Dataset [48]	TEgO [24]	**ORBIT Benchmark**
	Data type	Image	Image	Image	Image	Video frames
	# classes	1623	100	4934	19	486
	# samples/class	20	600	6-340,029	180-487	33-3,600
	# total samples	32,460	60,000	52,764,077	11,930	2,687,934
	Goal	Image classification	Image classification	Image classification	Image classification	Frame classification
	Task	Fixed shot/way	Fixed shot/way	Random shot/way	Fixed shot/way	Random shot/way
	Source	Turk	Web	Web	Mobile phone	Mobile phone
	Data collectors	Sighted (20)	Sighted	Sighted	Sighted (1) Blind (1)	Blind (67)
High-variation features	Unbalanced classes	✗	✗	✓	✗	✓
	Lighting variation	✗	✓	✓	✗	✓
	Background variation	✗	✓	✓	✓*	✓
	Viewpoint variation	✗	✗	✗	✓	✓
	Ill-framed objects	✗	✗	✗	✓	✓
	Blur	✗	✗	✗	✗	✓

Table 1: Comparison of few-shot learning datasets. Note, the ORBIT benchmark dataset is a subset of all videos contributed by collectors (see Appendix B). *Collected in 2 controlled environments – 1 uniform background, 1 cluttered space.

3. ORBIT Benchmark Dataset

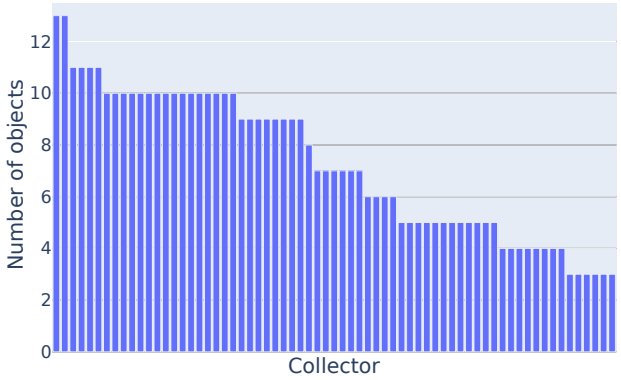
Our goal is to drive research in recognition tasks under few-shot, high-variation conditions so that deployed few-shot systems are robust to such conditions. Toward this goal, we focus on a real-world application that serves as a microcosm of a few-shot, high-variation setting — TORs for people who are blind/low-vision – and engage the blind/low-vision community in collecting a large-scale dataset.

The collection took place in two phases, and collectors recorded and submitted all videos (completely anonymously) via an accessible iOS app (see Appendix A.2). The collection protocol was designed and validated through extensive user studies [44] and led to the key decision to capture *videos* rather than images of objects. This was based on the hypothesis that a video increases a blind collector's chances of capturing frames that contained the object while reducing the time/effort cost to the collector, compared to multiple attempts at a single image. The study was approved by the City, University of London Research Ethics Committee. The full data collection protocol is described in Appendix A.1 and a datasheet [10] for the dataset is included in Appendix E.

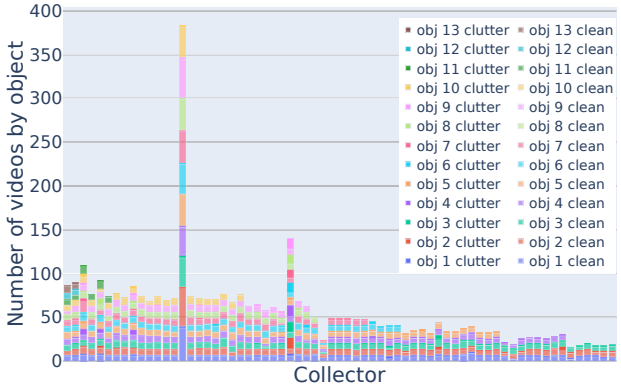
We summarize the benchmark dataset in Table 2 and describe it in detail below (see Appendix B for dataset preparation, and Appendix C for example clips). The benchmark dataset is used to run the benchmark described in Section 4.

Number of collectors. Globally, 77 collectors contributed to the ORBIT benchmark dataset. Collectors who contributed only 1 object were merged to enforce a minimum of 3 objects per user such that the per-user classification task was a minimum of 3-way, resulting in an effective 67 users.

Numbers of videos and objects. Collectors contributed a total of 486 objects and 3,822 videos (2,687,934 frames, 83GB). 2,996 videos showed the object in isolation, referred to as *clean* videos, while 826 showed the object in a realistic, multi-object scene, referred to as *clutter* videos. We collected both types to match what a TOR will encounter in the real-world (see Section 4.2.2). Each collector contributed on average 7.3 (±2.8) objects, with 5.8 (±3.9) clean videos and 1.8 (±1.1) clutter videos per object. Figure 2 shows the number of objects (2a) and number of videos per collector (2b). We discuss the impact of the 2 collectors who contributed more videos than the average collector in Appendix B.3.



(a) Number of objects per collector.



(b) Number of videos (stacked by object) per collector.

Figure 2: Number of objects and videos across 67 collectors.

Types of objects. Collectors provided object labels for each video contributed. Objects covered course-grained categories (e.g. remote, keys, wallet) as well as fine-grained

	Collectors	Objects	Videos	Videos per object			Frames per video		
				mean/std	25/75th perc.	min/max	mean/std	25/75th perc.	min/max
Total	67	486	3822	7.9/4.8	7.0/7.0	3.0/46.0	703.3/414.1	396.2/899.0	33.0/3600.0
Clean			2996	6.2/4.6	5.0/6.0	2.0/44.0	771.3/420.6	525.8/900.0	33.0/3600.0
Clutter			826	1.7/1.5	1.0/2.0	1.0/13.0	456.7/272.9	248.5/599.0	40.0/3596.0
Per-collector	1	7.3/2.8	57.0/47.4	7.5/4.0	6.6/7.4	3.4/38.4	728.8/208.8	609.4/808.2	213.1/1614.3
Clean			44.7/44.0	5.8/3.9	4.8/6.0	2.4/36.5	809.9/244.7	664.7/898.5	219.3/1872.6
Clutter			12.3/10.8	1.8/1.1	1.0/2.0	1.0/9.9	728.8/208.8	609.4/808.2	213.1/1614.3

Table 2: ORBIT benchmark dataset.

categories (e.g. Apple TV remote, Virgin remote, Samsung TV remote control). For summarization purposes, we clustered the objects based on object similarity and observe a long-tailed distribution (see Figure A.7b). The largest clusters contained different types of remotes/controls, keys, wallets/purses, guidecanes, doors, airpods, headphones, mobile phones, watches, sunglasses and Braille readers. More than half of the clusters contained just 1 object. The clustering algorithm and cluster contents are included in Appendix D.

Bounding box annotations. Since the clutter videos could contain multiple objects, we provide bounding box annotations around the target object in all clutter videos (available in the code repository). We use these to compute the proportion of time the target object spends in- versus out-of-frame per video, and show this in Figure A.6 averaged over all clutter videos per collector. On average, the target object is in-frame for ∼95% of any given clutter video.

Video lengths. Video lengths depended on the recording technique required for each video type (see Appendix A.1). On average, clean videos were 25.7s (∼771 frames at 30 FPS), and clutter videos were 15.2s (∼457 frames at 30 FPS).

Unfiltered ORBIT dataset. Some collectors did not meet the minimum requirements to be included in the benchmark dataset (e.g. an object did not have both clean and clutter videos). The benchmark dataset was therefore extracted from a larger set of 4733 videos (3,161,718 frames, 97GB) of 588 objects contributed by 97 collectors. We summarize the unfiltered dataset in Appendix A.3.

4. Teachable Object Recognition Benchmark

The ORBIT dataset can be used to explore a wide set of real-world recognition tasks from continual learning [27, 28] to video segmentation [25, 34, 29]. In this paper, we focus on few-shot object recognition from high-variation examples and present a realistic and challenging few-shot benchmark grounded in TORs for people who are blind/low-vision.

In Section 4.1, we describe how a TOR works, mapping it to a few-shot learning problem, before presenting the benchmark's evaluation protocol and metrics in Section 4.2.

4.1. Teachable Object Recognition

We define a TOR as a generic recognizer that can be customized to a user's personal objects using a small number of training examples – in our case, videos – which the user has captured themselves. The 3 steps to realizing a TOR are:

(1) **Train.** A recognition model is trained on a large dataset of objects where each object has only a few examples. The model can be optimized to either i) directly recognize a set of objects [46, 5] or ii) learn *how* to recognize a set of objects (i.e. meta-learn) [9, 40, 49, 38]. This happens before deploying the model in the real world.

(2) **Personalize.** A real-world user captures a few examples of a set of their personal objects. The deployed model is trained on this user's objects using just these examples.

(3) **Recognize.** The user employs their now-personalized recognizer to identify their personal objects in novel (test) scenarios. As the user points their recognizer at a scene, it delivers frame-by-frame predictions.

4.1.1 TORs as a few-shot learning problem

The **(1) train** step of a TOR can be mapped to the 'meta-training' phase typically used in few-shot learning set-ups. The **(2) personalize** and **(3) recognize** steps can be mapped to 'meta-testing' (see Figure 3). With this view, we now formalize the teachable object recognition task, drawing on nomenclature from the few-shot literature [9, 40, 38, 11].

We construct a set of train users $\mathcal{K}^{\text{train}}$ and test users $\mathcal{K}^{\text{test}}$ ($\mathcal{K}^{\text{train}} \cap \mathcal{K}^{\text{test}} = \varnothing$) akin to the train and test object classes used in few-shot learning. A user κ has a set of personal objects $\mathcal{P}^\kappa$ that they want a recognizer to identify, setting up a $|\mathcal{P}^\kappa|$-way classification problem. To this end, the user captures a few videos of each object, together called the user's "context" set $\mathcal{C}^\kappa = \{(\bar{\boldsymbol{v}}, p)_i\}_{i=1}^N$, where $\bar{\boldsymbol{v}}$ is a context video, $p \in \mathcal{P}^\kappa$ is its object label, and N is the total number of the user's context videos. The goal is to use $\mathcal{C}^\kappa$ to learn a recognition model f_{θ^κ} that can identify the user's objects, where θ^κ are the model parameters specific to user κ.

Once personalized, the user can point their recognizer at novel "target" scenarios to receive per-frame predictions:

$$y_f^* = \underset{y_f \in \mathcal{P}^\kappa}{\arg\max}\ f_{\theta^\kappa}(v_f) \qquad v_f \in \boldsymbol{v} \qquad (\boldsymbol{v}, p) \in \mathcal{T}^\kappa \quad (1)$$

where v_f is a target frame, $\boldsymbol{v}$ is a target video, $\mathcal{T}^\kappa$ is all the user's target videos, and $y_f \in \mathcal{P}^\kappa$ is the frame-level label.[1]

[1] Note, $y_f = p$ where $p \in \mathcal{P}^\kappa$ is the video-level object label

Following the typical paradigm, during meta-training (i.e. the **train** step), multiple tasks are sampled per user $\kappa \in \mathcal{K}^{\text{train}}$ where a task is a random sub-sample of the user's $\mathcal{C}^\kappa$ and $\mathcal{T}^\kappa$ (see Appendix G.2). The recognition model can be trained on these tasks using an episodic [9, 40, 49, 38] or non-episodic approach [5, 46, 22]. We formalize both in the context of TORs in Appendix F. Then, at meta-testing, one task is sampled per test user $\kappa \in \mathcal{K}^{\text{test}}$ containing *all* the user's context and target videos. For each test user, the recognizer is personalized using all their context videos $\mathcal{C}^\kappa$ (i.e. the **personalize** step), and then evaluated on each of the user's target videos in $\mathcal{T}^\kappa$ (i.e. the **recognize** step). In the following section, we discuss this evaluation protocol.

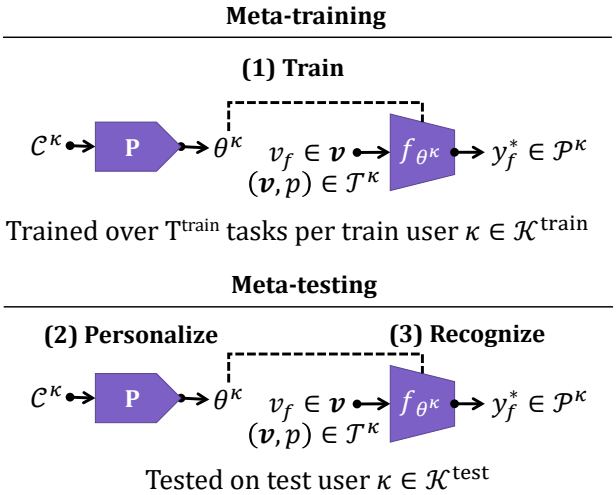


Figure 3: Teachable object recognizers cast as a few-shot learning problem. P is the personalization method, for example, several gradient steps using a optimization-based approach, or parameter generation using a model-based approach (see Section 5.1).

4.2. Evaluation protocol

ORBIT's evaluation protocol is designed to reflect how well a TOR will work in the hands of a real-world user — both in terms of performance and computational cost to personalize. To achieve this, we test (and train) in a user-centric way where tasks are sampled *per-user* (that is, only from a given user's objects and its associated context/target videos). This contrasts existing few-shot (and other) benchmarks, and offers powerful insights into how well a meta-trained TOR can personalize to a single user.

4.2.1 Train/validation/test users

The user-centric formulation in Section 4.1.1 calls for a disjoint set of train users $\mathcal{K}^{\text{train}}$ and test users $\mathcal{K}^{\text{test}}$. We therefore separate the 67 ORBIT collectors into 44 train users and 17 test users, with the remaining 6 marked as validation users $\mathcal{K}^{\text{val}}$. To ensure the test case is sufficiently challenging, we enforce that test (and validation) users have a minimum of 5 objects (see further details in Appendix B.3). The total number of objects in the splits are 278/50/158, respectively. We report statistics for each set of train/validation/test users in Appendix C, mirroring those over all users in Section 3.

4.2.2 Evaluation modes

We establish 2 evaluation modes:

Clean video evaluation (CLE-VE). We construct a test user's context set $\mathcal{C}^\kappa$ from their clean videos, and target set $\mathcal{T}^\kappa$ from a *held-out* set of their clean videos. This mode serves as a simple check that the user's clean videos can be used to recognize the user's objects in novel 'simple' scenarios when the object is in isolation.

Clutter video evaluation (CLU-VE). We construct a test user's context set $\mathcal{C}^\kappa$ from their clean videos, and target set $\mathcal{T}^\kappa$ from their clutter videos. This mode matches the real-world usage of a TOR where a user captures clean videos to register objects, and needs to identify those objects in complex, cluttered environments. We consider CLU-VE to be ORBIT's primary evaluation mode since it most closely matches how a TOR will be used in the real-world.

4.2.3 Evaluation metrics

For a test user $\kappa \in \mathcal{K}^{\text{test}}$, we evaluate their personalized recognizer f_{θ^κ} on each of their target videos. We denote a target video of object $p \in \mathcal{P}^\kappa$ as $\boldsymbol{v} = [v_1, \ldots, v_F]$, and its frame predictions as $\boldsymbol{y}^* = [y_1^*, \ldots, y_F^*]$, where F is the number of frames and $y_f^* \in \mathcal{P}^\kappa$. We further denote y_{mode}^* as the video's most frequent frame prediction. For a given target video, we compute its:

Frame accuracy: the number of correct frame predictions, by the total number of frames in the video.

Frames-to-recognition (FTR): the number of frames (w.r.t. the first frame v_1) before a correct prediction is made, by the total number of frames in the video.

Video accuracy: 1, if the video-level prediction equals the video-level object label, $y_{\text{mode}}^* = p$, otherwise 0.

We compute these metrics for each target video in all tasks for all users in $\mathcal{K}^{\text{test}}$. We report the average and 95% confidence interval of each metric over this flattened set of videos, denoted $\mathcal{T}^{\text{all}}$ (see equations in Table 3). We also compute a further 2 computational cost metrics:

MACS to personalize: number of Multiply-Accumulate operations (MACS) to compute a test user's personalized parameters θ^κ using their context videos $\mathcal{C}^\kappa$, reported as the average over all tasks pooled across test users.

Number of parameters: total parameters in recognizer.

We flag frame accuracy as ORBIT's primary metric because it most closely matches how a TOR will ultimately be used. The remaining metrics are complementary: FTR captures how long a user would have to point their recognizer at a

FRAME ACCURACY (↑)	FRAMES-TO-RECOGNITION (↓)	VIDEO ACCURACY (↑)
$\frac{1}{\lvert\mathcal{T}^{\text{all}}\rvert}\sum_{(\boldsymbol{v},p)\in\mathcal{T}^{\text{all}}}\frac{\sum_{f=1}^{\lvert\boldsymbol{v}\rvert}\mathbb{1}[y_f^*=p]}{\lvert\boldsymbol{v}\rvert}$	$\frac{1}{\lvert\mathcal{T}^{\text{all}}\rvert}\sum_{(\boldsymbol{v},p)\in\mathcal{T}^{\text{all}}}\frac{\arg\min_{v_f\in\boldsymbol{v}} y_f^*=p}{\lvert\boldsymbol{v}\rvert}$	$\frac{1}{\lvert\mathcal{T}^{\text{all}}\rvert}\sum_{(\boldsymbol{v},p)\in\mathcal{T}^{\text{all}}}\mathbb{1}\left[y^*_{\text{mode}}=p\right]\quad y^*_{\text{mode}}=\arg\max_{p\in\mathcal{P}^\kappa}\sum_{f=1}^{\lvert\boldsymbol{v}\rvert}\mathbb{1}[y_f^*=p]$

Table 3: ORBIT evaluation metrics. Symbols ↑ / ↓ indicate up / down is better, respectively. $\mathcal{T}^{\text{all}}$ is the set of all target videos pooled across all tasks for all test users in $\mathcal{K}^{\text{test}}$.

scene before it identified the target object (with fewer frames being better) while video accuracy summarizes the predictions over a whole video. MACS to personalize provides an indication whether personalization could happen directly on a user's device or a cloud-based service is required, each impacting how quickly a recognizer could be personalized. The number of parameters indicates the storage and memory requirements of the model on a device, and if cloud-based, the bandwidth required to download the personalized model. It is also useful to normalize performance by model capacity.

5. Experimental analyses and results

5.1. Baselines & training set-up

Baselines. There are 3 main classes of few-shot learning approaches. In *metric-based* approaches, a per-class embedding is computed using the (labeled) examples in the context set, and a target example is classified based on its distance to each [40, 49]. In *optimization-based* approaches, the model takes many [51, 46, 5] or few [9, 52, 2] gradient steps on the context examples, and the updated model then classifies the target examples. Finally, in *amortization-based* approaches, the model uses the context examples to directly generate the parameters of the classifier which is then used to classify a target example [38, 11].

We establish baselines on the ORBIT dataset across these 3 classes. Within the episodic approaches, we choose Prototypical Nets [40] for the metric family, MAML [9] for the optimization family, and CNAPs [38] for the amortization family. We also implement a non-episodic fine-tuning baseline following [46, 5] who show that it can rival more complex methods. This selection of models offers good coverage over those that are competitive on current few-shot learning image classification benchmarks. For all implementation details of these baselines see Appendix G.1.

Video representation. In Section 4.1.1, tasks are constructed from the context and target videos of a given user's objects. We sample clips from each video and represent each clip as an average over its (learned) frame-level features. For memory reasons, we do not sample all clips from a video. Instead, during meta-training, we randomly sample S^{train} non-overlapping clips, each of L contiguous frames, from both context and target videos. Each clip is averaged and treated as an 'element' in the context/target set, akin to an image in typical few-shot image classification. During meta-testing, however, following Section 4.2 and Eq. (1), we must evaluate a test user's personalized recognizer on *every* frame in *all* of their target videos. We, therefore, sample *all* overlapping clips in a target video, where a clip is an L-sized buffer of each frame plus its short history. Ideally, this should also be done for context videos, however, due to memory reasons, we sample S^{test} non-overlapping L-sized clips from each context video, similar to meta-training. In our baseline implementations, $S^{\text{train}} = 4$, $S^{\text{test}} = 8$, and $L = 8$ (for further details see Appendices G.2 and G.3).

How frames are sampled during training/testing, and how videos are represented is flexible. The evaluation protocol's only strict requirement is that a model outputs a prediction for every frame from every target video for every test user.

Number of tasks per test user. Because context videos are sub-sampled during meta-testing, a test user's task contains a random set, rather than *all*, context clips. To account for potential variation, therefore, we sample 5 tasks per test user, and pool all their target videos into $\mathcal{T}^{\text{all}}$ for evaluation. If memory was not a constraint, following Section 4.1.1, we would sample one task per test user which contained *all* context and *all* target clips.

5.2. Analyses

Baseline comparison. Performance is largely consistent across the baseline models in both CLE-VE and CLU-VE modes (see Table 4). In CLE-VE, all methods are equivalent in frame accuracy, FTR and video accuracy, except for ProtoNets and CNAPs which trail slightly in frame accuracy. Comparing this to CLU-VE, we see overall performance drops of 10-15 percentage points. Here, models are overall equivalent on frame and video accuracy, however ProtoNets and FineTuner lead in FTR. Further, absolute CLU-VE scores are in the low 50s. Looking at the best possible bounds (computed using the bounding box annotations, see Figure A.6c) suggests that there is ample scope for improvement and motivates the need for approaches that can handle distribution shifts from clean (context) to real-world, cluttered scenes (target), and are robust to high-variation data more generally.

In computational cost, ProtoNets has the lowest cost to personalize requiring only a single forward pass of a user's context videos, while FineTuner has the highest, requiring 50 gradient steps. This, along with the total number of parameters (which are similar across models), suggests that ProtoNets and CNAPs would be better suited to deployment on a mobile device.

	Clean Video Evaluation (CLE-VE)				Clutter Video Evaluation (CLU-VE)					
MODEL	FRAME ACC	FTR	VIDEO ACC	MACS TO PERSONALIZE	FRAME ACC	FTR	VIDEO ACC	MACS TO PERSONALIZE	METHOD TO PERSONALIZE	# PARAMS
Best possible	-	-	-	-	**95.31 (1.37)**	**0.00 (0.00)**	**100.00 (0.00)**	-	-	-
ProtoNets [40]	65.16 (1.96)	**7.55 (1.35)**	**81.88 (2.51)**	**2.82×10^{12}**	**50.34 (1.74)**	**14.93 (1.52)**	**59.93 (2.48)**	**3.53×10^{12}**	**1 forward pass**	**11.17M**
CNAPs [38]	66.15 (2.08)	**8.40 (1.40)**	**79.56 (2.63)**	3.09×10^{12}	**51.47 (1.81)**	17.87 (1.69)	**59.53 (2.48)**	3.87×10^{12}	**1 forward pass**	12.75M
MAML [9]	**70.58 (2.10)**	**8.62 (1.56)**	**80.88 (2.56)**	84.63×10^{12}	**51.67 (1.88)**	20.95 (1.84)	57.87 (2.50)	105.99×10^{12}	15 gradient steps	**11.17M**
FineTuner [46]	**69.47 (2.16)**	**7.82 (1.54)**	**79.67 (2.62)**	282.09×10^{12}	**53.73 (1.80)**	**14.44 (1.50)**	**63.07 (2.44)**	353.30×10^{12}	50 gradient steps	**11.17M**

Table 4: Baselines on the ORBIT Dataset. Results are reported as the average (95% confidence interval) over all target videos pooled from 85 test tasks (5 tasks per test user, 17 test users). Best possible scores are computed using bounding box annotations which are available for the clutter videos (see Appendix C and Figure A.6).

MODEL	FRAME ACC	FTR	VIDEO ACC
ProtoNets [40]	**58.98 (2.23)**	**11.55 (1.79)**	**69.17 (3.01)**
CNAPs [38]	51.86 (2.49)	20.81 (2.33)	60.77 (3.18)
MAML [9]	42.55 (2.67)	37.28 (2.99)	46.96 (3.25)
FineTuner [46]	**61.01 (2.24)**	**11.53 (1.82)**	**72.60 (2.91)**

Table 5: CLE-VE performance when meta-training on Meta-Dataset and meta-testing on ORBIT (for CLU-VE see Table A.3). Even on clean videos, models perform poorly compared to when meta-training on ORBIT (Table 4) suggesting that existing few-shot datasets may be insufficient for real-world adaptation.

Meta-training on other few-shot learning datasets. A meta-trained model should, in principle, have the ability to learn *any* new object (from any dataset) with only a few examples. We investigate this by meta-training the baseline models on Meta-Dataset [48] using its standard task sampling protocol and then testing them on the ORBIT dataset (i.e. personalizing to test users with no training). We adapt the meta-trained models to videos by taking the average over frame features in clips sampled from context and target videos (see Section 5.1). In Table 5, we see that even on the easier, clean videos (CLE-VE), performance is notably lower than the corresponding baselines in Table 4 (for CLU-VE see Table A.3). MAML and CNAPs perform particularly poorly while ProtoNets and FineTuner fare slightly better, however, are still 6-8 percentage points below their above counterparts in frame accuracy. This suggests that even though much progress has been made on existing few-shot benchmarks, they are not representative of real-world conditions and models trained on them may struggle to learn new objects when only high-variation examples are available.

Per-user performance. In addition to averaging over $\mathcal{T}^{\text{all}}$, the benchmark's user-centric paradigm allows us to average *per-user* (i.e. over just their target videos). This is useful because it provides a measure of how well a meta-trained TOR would personalize to an individual real-world user. In Figure 4 however, we show that ProtoNets' personalization is not consistent across users, for some going as low as 25% in frame accuracy (for other metrics/models see Figure A.10). A TOR should be able adapt to *any* real-world user, thus future work should not only aim to boost performance on the metrics but also reduce variance across test users.

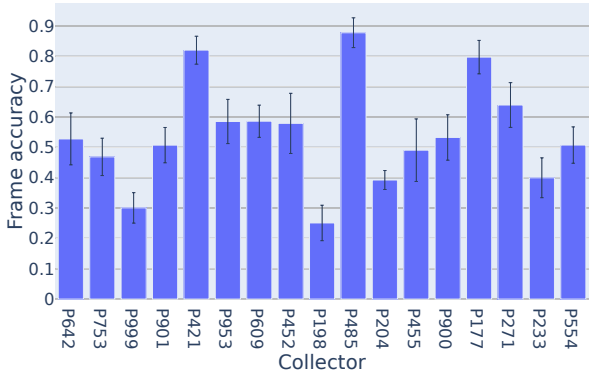


Figure 4: CLU-VE frame accuracy varies widely across test users (error bars are 95% confidence intervals) with ProtoNets [40]. For other metrics and models see Figure A.10.

Train task composition. Finally, we investigate the impact of the number of context videos per object (Figure 5), and the number of objects per user (Figure 6) sampled in train tasks on CLU-VE frame accuracy. In the first case, we expect that with more context videos per object, the more diversity the model will see during meta-training, and hence generalize better at meta-testing to novel (target) videos. To test this hypothesis, we fix a quota of 96 frames per object in each train task and sample these frames from increasing numbers of context videos. Frame accuracy increases with more context videos, but overall plateaus between 4-6 context videos per object. Looking at the number of objects sampled per user next, we cap all train user's objects at $\{2, 4, 6, 8\}$, respectively, when meta-training. We then meta-test in two ways: 1) we keep the caps in place on the test users, and 2) we remove the caps. For 1), we see reducing accuracy for increasing numbers of objects, as is expected – classifying between 8 objects is harder than classifying between 2. For 2), we see a significant drop in accuracy relative to 1) suggesting that meta-training with fewer objects than would be encountered at meta-testing is detrimental. This is an important real-world consideration since it is likely that over months/years, a user will accumulate many more objects than is currently present per user in the ORBIT dataset. Overall, however, training with a cap of 6 or more objects yields

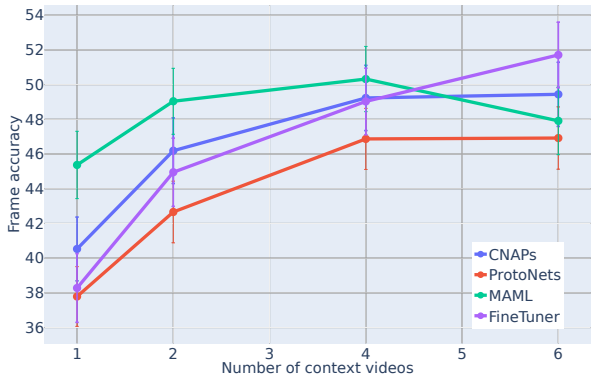


Figure 5: Meta-training with more context videos per object leads to better CLU-VE performance. Frames are sampled from an increasing number of clean videos per object using the number of clips per video (S^{train}) to keep the total number of context frames fixed per train task.

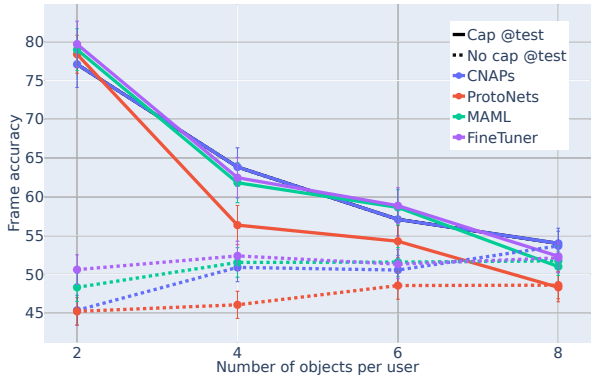


Figure 6: Meta-training and -testing with more objects per user poses a harder recognition problem (solid line), however, meta-training with fewer objects than encountered at meta-testing (dashed line) shows only a small CLU-VE performance drop compared to Table 4, suggesting that models may be able to adapt to more objects in the real-world.

roughly equivalent performance to that reported in Table 4 where no caps are imposed during training. Since ORBIT test users have up to 12 objects (see Figure A.3c), our results suggest that a minimum of half the number of ultimate objects for a test user may be sufficient for meta-training. We repeat these analyses for the other metrics in Figures A.8 and A.9, and include the corresponding tables in Tables A.5 and A.6. We also investigate the impact of the number of tasks sampled per train user, included in Appendix H.

6. Discussion

We present the ORBIT dataset and benchmark, both grounded in the few-shot application of TORs for people who are blind/low-vision. Our baseline performance and further analyses demonstrate, however, that current few-shot approaches struggle on realistic, high-variation data. This gap offers opportunities for new and exciting research, from making models robust to high-variation video data to quantifying the uncertainty in model predictions. More than just pushing the state-of-the-art in existing lines of thought, the ORBIT dataset opens up new types of challenges that derive from systems that will support human-AI partnership. We close by discussing three of these unique characteristics.

ORBIT's user-centric formulation provides an opportunity to measure how well the ultimate system will work in the hands of real-world users. This contrasts most few-shot (and other) benchmarks which retain no notion of the end-user. Our results show that the baselines do not perform consistently across users. In the real-world, the heterogeneity of users, their objects, videoing techniques and devices will make this even more challenging. It will therefore be important for models to quantify, explain and ultimately minimize variation across users, particularly as models are deployed in a wider variety of scenarios outside the high-income countries in which the dataset was collected.

Directly involving users in collecting a dataset intended to drive ML research comes with challenges: user-based datasets are harder to scale than web-scraped datasets [8, 26, 48] and users need an understanding of the potential system in order to contribute useful data. Building the system first would address these challenges, but it cannot be done without algorithmic innovation (which itself requires the dataset). The ORBIT dataset is a starting point and can be used to build the first generation of TORs, which can be deployed and themselves be used to collect more real-world data to drive a cycle of innovation between dataset and application.

Finally, grounding in a real-world application encourages innovation in new directions to meet the real-world conditions of deployment. This could range from new models that are lightweight enough to be personalized directly on a user's phone to new research problems like handling the scenario when none of a user's objects are in the frame.

In conclusion, the ORBIT dataset and benchmark aims to shape the next generation of recognition tools for the blind/low-vision community starting with TORs, and to improve the robustness of vision systems across a broad range of other applications.

Acknowledgments

The ORBIT Dataset is funded by Microsoft AI for Accessibility. LZ is supported by the 2017 MSR PhD Scholarship Program and 2020 MSR EMEA PhD Award. JB is supported by the EPSRC Prosperity Partnership EP/T005386/1. We thank VICTA, RNC, RNIB, CNIB, Humanware, Tekvision School for the Blind, BlindSA, NFB, and AbilityNet. Finally, we thank Emily Madsen for help with the video validation, and all the ORBIT collectors for their time and contributions.

References

[1] 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. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2019. 2

[2] Luca Bertinetto, João F. Henriques, Philip H.S. Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019. 1, 6

[3] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics (TACL)*, 5:135–146, 2017. 15

[4] Da Chen, Yuefeng Chen, Yuhong Li, Feng Mao, Yuan He, and Hui Xue. Self-supervised learning for few-shot image classification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021. 1, 2

[5] Yinbo Chen, Xiaolong Wang, Zhuang Liu, Huijuan Xu, and Trevor Darrell. A new meta-baseline for few-shot learning. *arXiv preprint arXiv:2003.04390*, 2020. 4, 5, 6, 29

[6] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1, 2

[7] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The EPIC-KITCHENS dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 2, 8

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017. 1, 4, 5, 6, 7, 28, 29, 31, 32, 33, 34

[10] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *arXiv preprint arXiv:1803.09010*, 2018. 3, 24

[11] Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard Turner. Meta-learning probabilistic inference for prediction. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019. 1, 4, 6

[12] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzyńska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017. 2

[13] Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. VizWiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017. 1

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 29

[16] Yuqing Hu, Vincent Gripon, and Stéphane Pateux. Leveraging the feature distribution in transfer-based few-shot learning. *arXiv preprint arXiv:2006.03806*, 2020. 1, 2

[17] Jonas Jongejan, Henry Rowley, Takashi Kawashima, Jongmin Kim, and Nick Fox-Gieg. The Quick, Draw! – A.I. experiment, 2016. 1, 2

[18] Hernisa Kacorri. Teachable machines for accessibility. In *Proceedings of the International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS)*, 2017. 2

[19] Hernisa Kacorri, Utkarsh Dwivedi, Sravya Amancherla, Mayanka Jha, and Riya Chanduka. IncluSet: A data surfacing repository for accessibility datasets. In *Proceedings of the International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS)*, 2020. 2

[20] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015. 29, 30

[21] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Sara Beery, et al. WILDS: A Benchmark of in-the-Wild Distribution Shifts. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021. 2

[22] Steinar Laenen and Luca Bertinetto. On episodes, prototypical networks, and few-shot learning. *arXiv preprint arXiv:2012.09831*, 2020. 5

[23] Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci)*, 2011. 1, 2, 3

[24] Kyungjun Lee and Hernisa Kacorri. Hands holding clues for object recognition in teachable machines. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 2019. 2, 3

[25] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. Key-segments for video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2011. 4

[26] 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

Proceedings of the European Conference on Computer Vision (ECCV), 2014. 1, 2, 8

[27] Vincenzo Lomonaco and Davide Maltoni. Core50: A new dataset and benchmark for continuous object recognition. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2017. 2, 4

[28] Vincenzo Lomonaco, Davide Maltoni, and Lorenzo Pellegrini. Rehearsal-free continual learning over small non-IID batches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshop*, 2020. 4

[29] Xiankai Lu, Wenguan Wang, Jianbing Shen, Yu-Wing Tai, David J Crandall, and Steven CH Hoi. Learning video object segmentation from unlabeled videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 4

[30] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 1

[31] Daniela Massiceti, Lida Theodorou, Luisa Zintgraf, Matthew Tobias Harris, Simone Stumpf, Cecily Morrison, Edward Cutrell, and Katja Hofmann. ORBIT: A real-world few-shot dataset for teachable object recognition collected from people who are blind or low vision. URL https://doi.org/10.25383/city.14294597. City, University of London, Apr 2021. 2, 13

[32] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics & Image Processing (ICVGIP)*, 2008. 1, 2

[33] Eunbyung Park and Junier B Oliva. Meta-curvature. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2019. 1, 2

[34] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4

[35] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 29

[36] Joseph Redmon and Ali Farhadi. YOLO9000: Better, faster, stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2015. 1

[38] James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E. Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2019. 1, 4, 5, 6, 7, 28, 29, 31, 32, 33, 34

[39] 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. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1, 2, 29

[40] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 1, 4, 5, 6, 7, 28, 29, 31, 32, 33, 34

[41] Joan Sosa-García and Francesca Odone. "Hands on" visual recognition for visually impaired users. *ACM Transactions on Accessible Computing (TACCESS)*, 10(3):1–30, 2017. 2

[42] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-V4, Inception-ResNet and the impact of residual connections on learning. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, 2017. 1

[43] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019. 1

[44] Lida Theodorou, Daniela Massiceti, Luisa Zintgraf, Simone Stumpf, Cecily Morrison, Edward Cutrell, Matthew Tobias Harris, and Katja Hofmann. Disability-first Dataset Creation: Lessons from Constructing a Dataset for Teachable Object Recognition with Blind and Low Vision Data Collectors. In *Proceedings of the International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS)*, 2021. 3, 12

[45] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012. 1

[46] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 4, 5, 6, 7, 29, 31, 32, 33, 34

[47] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy: FixEfficientNet. *arXiv preprint arXiv:2003.08237*, 2020. 1

[48] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020. 1, 2, 3, 7, 8, 29

[49] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2016. 1, 2, 3, 4, 5, 6, 28

[50] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical Report CNS-TR-2011-001, 2011. 1, 2

[51] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks?

In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2014. 6, 29

[52] Luisa M. Zintgraf, Kyriacos Shiarlis, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019. 6, 28