Journal of Data-centric Machine Learning Research (2026) Submitted 04/26; Published 04/26

Wake Vision: A Tailored Dataset and Benchmark Suite for TinyML Computer Vision Applications

Colby Banbury colbybanbury@gmail.com
Harvard University
Cambridge, MA 02138, USA

Emil Njor* njor@mmmi.sdu.dk
Technical University of Denmark, Kongens Lyngby, 2800, Denmark
Harvard University, Cambridge, MA 02138, USA

Andrea Mattia Garavagno AndreaMattia.Garavagno@edu.unige.it
Harvard University, Cambridge, MA 02138, USA
Scuola Superiore Sant'Anna, Pisa, 56127, Italy
University of Genoa, Genoa, 16126, Italy

Mark Mazumder markmazumder@g.harvard.edu
Harvard University
Cambridge, MA 02138, USA

Matthew Stewart matthew_stewart@g.harvard.edu
Harvard University
Cambridge, MA 02138, USA

Pete Warden pete@petewarden.com
Moonshine AI

Manjunath Kudlur
Moonshine AI

Nat Jeffries
Moonshine AI

Xenofon Fafoutis xefa@dtu.dk
Technical University of Denmark
Kongens Lyngby, 2800, Denmark

Vijay Janapa Reddi vj@eecs.harvard.edu
Harvard University
Cambridge, MA 02138, USA

Reviewed on OpenReview: *https://openreview.net/forum?id=eb82tUIZRA*

Editor: Zach Xu

*. Now affiliated with The University of Southern Denmark, Vejle, 7100, Denmark

©2026 Banbury, Njor, Garavagno, Mazumder et al.

Abstract

Tiny machine learning (TinyML) co-locates models with sensors on microcontrollers, where small models (which are disproportionately sensitive to label noise) and bespoke binary tasks (which lack standard benchmarks) make general-purpose dataset practices a poor fit. Visual Wake Words (VWW), the prior standard TinyML person detection benchmark, contains roughly 123K images and has an estimated label error rate of 7.8%, which limits its usefulness for production-grade systems. Manual labeling, however, is prohibitively expensive for the scale and diversity of TinyML use cases.

We address this gap with the Wake Vision pipeline, an automated method for generating and curating large-scale binary classification datasets for TinyML. We use *data-centric TinyML* for the dataset construction, curation, and lifecycle methods that produce the large, well-curated datasets these systems require. The pipeline combines label fusion across image-level and bounding-box sources, confidence-, area-, and depiction-aware filtering, label correction on the evaluation splits, and automatic generation of fine-grained benchmark subsets. Applying it to person detection, we release Wake Vision, a dataset of almost 6M images (close to 100× more person images than VWW) with a manually relabeled validation and test set at a 2.2% label error rate.

Models trained on Wake Vision improve test accuracy by up to 6.6% over VWW across MobileNetV2, MCUNet, MicroNets, and ColabNAS architectures, and match or exceed VWW-trained models on 13 of 16 fine-grained subsets covering perceived gender, perceived age, distance, lighting, and depictions. The advantage holds under distribution shift on three out-of-distribution datasets covering driving and overhead-surveillance imagery. We additionally uncover two TinyML-specific insights: small models are more sensitive to label errors than large models, and two-stage training, which pretrains on the noisier large set and fine-tunes on the cleaner small set, is a viable strategy even for tiny, low-capacity models. Beyond person detection, the Wake Vision pipeline applies to the 9.6K trainable classes of Open Images v7; on bird detection it produces a dataset 27× larger than a VWW-style baseline with label error reduced from 6.6% to 0.6%. All artifacts are released under CC-BY 4.0 through TensorFlow Datasets and Hugging Face. To continue improving the dataset over time, we partner with the Edge AI Foundation to host community competitions; the first round contributed a label-correction technique that reduced the Wake Vision (Large) label error rate from 15.2% to 9.8%, at a cost orders of magnitude below the $600,000 implied by manual relabeling at this scale.

Keywords: TinyML, Dataset, Benchmark, Computer Vision, Person Detection

1 Introduction

Tiny machine learning (TinyML) (Banbury et al., 2020; Warden and Situnayake, 2019; Banbury et al., 2021a) co-locates models with sensors on microcontrollers (MCUs), typically with a memory budget of hundreds of kilobytes, nearly four orders of magnitude less than common smartphone models (Banbury et al., 2021b; David et al., 2021). These constraints rule out conventional benchmarks. ImageNet's thousand-class output is infeasible on MCUs (Deng et al., 2009; Chowdhery et al., 2019), and training data must instead come from datasets purpose-built for the binary or low-class-count tasks that fit MCU budgets. Existing TinyML datasets such as Visual Wake Words (VWW) (Chowdhery et al., 2019) and Google Speech Commands (Warden, 2018) target exactly these simpler applications, supporting binary classification and tens of classes, respectively, but their limited scale and label error rates make them unsuitable for training and evaluating production-grade TinyML models.

Table 1: Wake Vision (**bold**) compared against the standard TinyML person detection benchmark and two general image classification datasets sometimes used as person-detection proxies (Chowdhery et al., 2019; Krizhevsky et al., 2009; Everingham et al., 2012). Wake Vision (Large) and Wake Vision (Quality) share the same validation and test sets.

Dataset	Total Images	# of Person Images Train	Validation	Test	Fine-Grained Filtering	Suitable for TinyML
Wake Vision (Quality)	**1,322,574**	**624,115**	**9,291**	**27,881**	✓	✓
Wake Vision (Large)	**5,760,428**	**2,880,214**	–	–	✗	✓
Visual Wake Words	123,287	36,000	3,926	19,107	✗	✓
CIFAR-100	60,000	2,500	–	500	✗	✗
PASCAL VOC 2012	11,530	1,994	2,093	–	✗	✗

The gap is methodological as well as empirical. Building TinyML datasets at the scale required for modern training, and across the diversity of tasks TinyML is being deployed for, requires automated pipelines for creating, curating, and evaluating datasets, not just larger downloads. We use *data-centric TinyML* for dataset construction, curation, and lifecycle methods that target two TinyML-specific constraints: (i) deployment models small enough where label noise materially erodes accuracy and (ii) bespoke binary or low-class-count tasks for which standard benchmarks do not exist, requiring datasets to be generated per-task from broader sources rather than reused, a property the Wake Vision pipeline addresses by targeting any of the 9.6K trainable classes of Open Images v7 (Section 3.5).

Wake Vision is the successor to Visual Wake Words (VWW). It preserves the binary person-detection task and evaluation conventions VWW established for the TinyML community, and addresses VWW's limitations in scale, label quality, and licensing through the Wake Vision pipeline, an automated method for generating and curating large-scale binary classification datasets for TinyML. The pipeline fuses multiple label sources, applies confidence-, area-, and depiction-aware filtering, supports manual and automated label correction on the evaluation splits, and generates fine-grained benchmark subsets (Figure 2 in Section 3). Automation is a necessity rather than a convenience. Our manual relabeling of approximately 70K validation and test images cost $7,000 USD, which extrapolates to roughly $600,000 USD for a 6M-image training set, and human labelers introduce inconsistency at scale. The pipeline derives datasets from Open Images v7 (Kuznetsova et al., 2020; Krasin et al., 2017), which lets the resulting datasets be released under a permissive CC-BY 4.0 license.

We demonstrate the pipeline by creating Wake Vision, a binary person detection dataset. Person detection is the canonical vision use case for TinyML (Banbury et al., 2021a), supporting applications from occupancy detection (Piechocki et al., 2022) and smart HVAC/lighting (Zacharia et al., 2022) to an always-on 'wake model' that gates power-intensive sensors and larger models (Heitz et al., 2008). As shown in Table 1, Wake Vision provides almost 6M images, close to 100× more person images than Visual Wake Words (VWW) (Chowdhery et al., 2019), and improves test accuracy by up to 6.6%. Alongside the dataset, we develop a suite of five fine-grained benchmarks (perceived gender, perceived age, distance, lighting, and depictions; Figure 1, Section 3.3), on which Wake Vision-trained

models match or exceed VWW-trained models on 13 of 16 subsets, exposing failure modes that an aggregate accuracy metric obscures.

Wake Vision exposes two TinyML-specific findings (Section 4.1, Section 4.7). First, smaller models lose more accuracy from training-set label noise than larger models do. Under our label-noise injection protocol, the smallest model loses 1.3% accuracy when the training error rate rises from 6.8% to 15%, versus 0.5% for the largest. Second, two-stage training that pretrains on Wake Vision (Large) and fine-tunes on Wake Vision (Quality) reaches 85.72% test accuracy, versus 84.89% for Quality alone and 80.80% for Large alone, combining the scale of one set and the quality of the other.

Our work makes four contributions:

- **The Wake Vision pipeline**, an automated method for building bespoke large-scale binary classification datasets for TinyML, combining multi-source label fusion, confidence-, area-, and depiction-aware filtering, label correction, and fine-grained benchmark synthesis (Sections 3.1–3.3). Beyond person detection, we apply the pipeline to two additional Open Images categories chosen because they also exist in MS-COCO, allowing direct comparison to the VWW-derived methodology: it produces a car detection dataset 12× larger than the VWW-style baseline (label error 6.4% → 4.8%) and a bird detection dataset 27× larger (6.6% → 0.6%, both estimated on 500-image samples), letting TinyML practitioners generate task-specific datasets without further large-scale labeling effort (Section 3.5, Section 4.5).
- **Wake Vision and a community-driven improvement loop.** A person detection dataset of almost 6M images (close to 100× more person images than VWW), with a manually relabeled validation and test set at a 2.2% label error rate (versus 7.8% in VWW), released under CC-BY 4.0 through TensorFlow Datasets and Hugging Face. We treat the dataset as a maintained artifact through recurring competitions hosted by the Edge AI Foundation; the first round contributed a label-correction technique that reduced the Wake Vision (Large) label error rate from 15.2% to 9.8%, a single-round improvement to a property that conventionally requires manual relabeling at a cost we estimate at $600,000 for a 6M-image set (Sections 3.2, 4.8).
- **A fine-grained benchmark suite** that evaluates TinyML model robustness across five dimensions: perceived gender, perceived age, distance, lighting, and depictions. Wake Vision-trained models match or exceed VWW-trained models on 13 of 16 fine-grained subsets (strict wins on 11, ties on 2 lighting subsets), with the largest gains on depicted persons, older individuals, and dark lighting (Section 3.3, Section 4.4).
- **Empirical findings for data-centric TinyML**, including that two-stage training (Wake Vision Large pretraining followed by Wake Vision Quality fine-tuning) achieves the best accuracy across MobileNetV2, MCUNet, MicroNets, and ColabNAS architectures, that small models are disproportionately sensitive to label noise, and that the accuracy advantage of Wake Vision-trained models persists under distribution shift on driving and overhead-surveillance imagery (Section 4.1, Section 4.7, Section 4.6).

Our dataset, benchmarks, code, and models are at `https://wakevision.ai`.

2 Background and Related Work

We organize related work into three categories:

(a) Older Person (b) Near Person (c) Bright Image (d) Female Person (e) Depicted Person

Figure 1: One sample from each of the five fine-grained benchmark dimensions in Wake Vision: perceived age, distance, lighting, perceived gender, and depictions. Full subset compositions appear in Figure 5.

Generating Binary Classification Datasets for TinyML Automatically generated datasets can significantly lower costs associated with deploying TinyML systems. Prior to our work, the primary method for generating binary classification datasets for TinyML applications was through the code used to create Visual Wake Words (VWW) (Chowdhery et al., 2019), which can be applied to other MS-COCO (Lin et al., 2014) classes. While we offer the same capability for Open Images, the Wake Vision pipeline extends beyond dataset generation, providing customizable curation properties that include label correction, fine-grained benchmark suite generation, and community-driven enhancements through competitions, addressing the entire dataset lifecycle.

Person Detection in TinyML Applications Person detection has emerged as the canonical computer vision use case for TinyML systems (Banbury et al., 2020, 2021a). In the context of resource-constrained computing, TinyML systems demand tasks that carefully balance computational feasibility with real-world utility. Person detection represents an optimal compromise. It is sufficiently lightweight to be implemented on embedded microcontrollers while maintaining enough discriminative power for practical applications. In particular, person detection serves as a critical activation mechanism in resource-constrained systems. These lightweight models support always-on deployment on low-powered MCUs to selectively "wake" power-intensive sensors, processors, and larger, more capable ML models upon detecting human presence (Heitz et al., 2008). This architectural pattern enables substantial energy savings while preserving privacy through on-device processing, and supports diverse applications ranging from occupancy detection (Piechocki et al., 2022), smart HVAC systems (Zacharia et al., 2022), or poacher detection (Doull et al., 2021).

Existing Person Detection Datasets In the TinyML computer vision domain, the Visual Wake Words (VWW) dataset (Chowdhery et al., 2019; David et al., 2021; Lin et al., 2020; Banbury et al., 2021b,a) has established itself as the de facto standard. Before Wake Vision, it was the only open-source dataset specifically designed for person detection with direct commercial applicability. However, VWW faces significant limitations: its small size and indirect access requirements (requiring regeneration from MS-COCO (Lin et al., 2014)) restrict its accessibility and utility.

Although other datasets contain person labels within general image classification collections, such as Cifar-100 (Krizhevsky et al., 2009) and PASCAL Visual Object Classes (Everingham et al., 2012), these present significant drawbacks for TinyML applications. Specifically, their inadequate representation of the open "no-person" class can lead to poor per-

ceived performance of TinyML models (Chowdhery et al., 2019). Our Wake Vision dataset addresses these limitations by providing nearly two orders of magnitude more images than any existing public, permissively licensed dataset (Table 1). Furthermore, it distinguishes itself as the only person detection dataset offering a fine-grained benchmark suite and official distribution through popular services like TensorFlow Datasets (TFDS) and Hugging Face Datasets.

3 Person Detection Dataset and Benchmark Generation

We apply the Wake Vision pipeline shown in Figure 2 to person detection, producing a *production-grade* dataset two orders of magnitude larger than prior TinyML person detection datasets (Table 2). We use *production-grade* to mean a dataset whose scale, manually relabeled evaluation splits, permissive license, and distribution through standard ML frameworks make it suitable for training and evaluating models that will be deployed in real systems, rather than only used as a research benchmark. The dataset's size, quality, and detailed metadata open avenues of TinyML research that smaller benchmarks cannot, and Figure 1 illustrates example images.

More specifically, the pipeline implements six data-centric operations (Figure 2): (1) *multi-source label fusion* that reconciles image-level and bounding-box annotations from Open Images into a single binary task (Section 3.1); (2) *confidence-, area-, and depiction-aware filtering* that removes uncertain, distant-subject, and depicted-only samples to match deployment conditions (Section 3.1); (3) *automated label correction* via Confident Learning on the evaluation splits as a low-cost first pass (Appendix K); (4) *manual relabeling* of the validation and test sets through a crowd-sourced platform, prioritized to where label quality has the largest effect on reported model performance (Section 3.2); (5) *fine-grained benchmark synthesis* that uses the per-image metadata produced by steps (1)–(4) to derive evaluation subsets along axes that an aggregate accuracy metric obscures (Section 3.3); and (6) *community-driven continuous improvement* through recurrent competitions whose winning contributions are integrated into subsequent dataset releases (Section 4.8). Sections 3.1–4.8 apply these operations to person detection, and Section 3.5 demonstrates that the same pipeline targets the 9.6K trainable classes of Open Images v7, with a comparison to Visual Wake Words (VWW).

3.1 Label Generation

A large person detection dataset is indispensable for TinyML research. However, manually labeling millions of images would be prohibitively expensive for a nascent field like TinyML. Therefore, we use the Wake Vision pipeline to bootstrap Wake Vision using existing large-scale data efforts.

The base label in Wake Vision is a binary person/non-person label. The first step in the Wake Vision pipeline is to derive these labels from the Open Images dataset (Kuznetsova et al., 2020; Krasin et al., 2017), which contains both image-level and bounding box labels of 9.6K trainable classes and 600 objects, respectively. Image-level labels are unlocalized, describing objects present in an image, whereas bounding box labels localize an object by four coordinates. The bounding box label classes are hierarchically structured, allowing one class to be a subcategory or a part of another class. For example a "Woman" is a

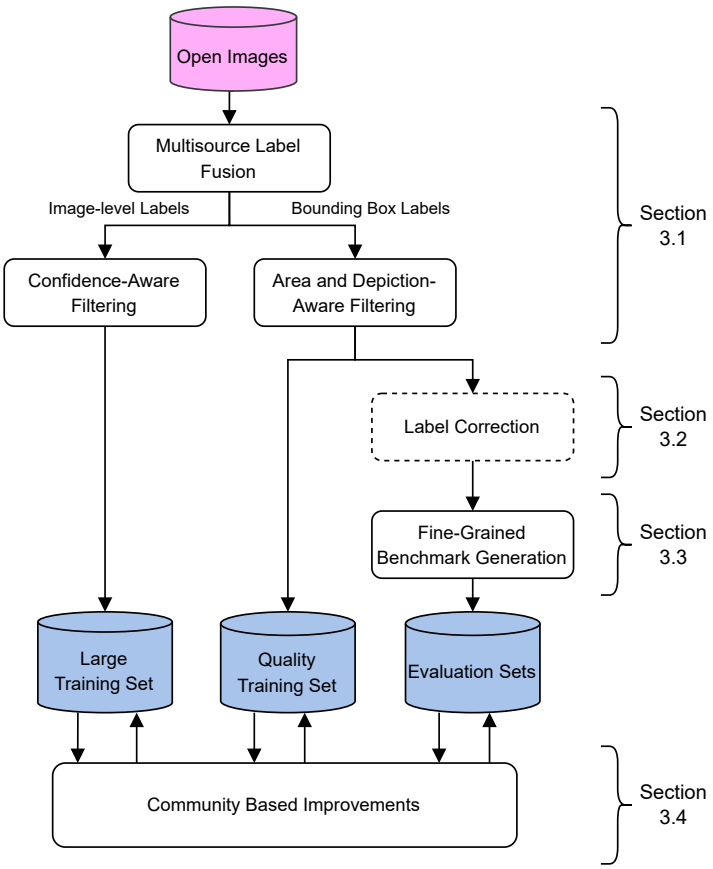


Figure 2: The Wake Vision dataset generation pipeline. Open Images image-level and bounding-box annotations are fused, filtered for confidence, person bounding-box area, and depiction status, optionally relabeled (manually or automatically), and split into training, validation, and test sets together with fine-grained benchmark subsets. Stages are detailed in Sections 3.1 to 3.3; dotted lines indicate optional steps.

subcategory of a "Person," and a "Human Hand" is a part of "Person." A comprehensive list of Open Images labels related to Wake Vision can be found in Appendix I.

Users can adapt the Wake Vision pipeline to meet the needs of specific use cases by utilizing the numerous configuration options available. For example, a configuration option exists to change whether artistic depictions of humans are labeled as non-persons or excluded from the dataset. We refer to Appendix G for details about these configuration options.

We observe that bounding box labels are generally more accurate and provide more information for data filtering while being less numerous. The Open Images training set has ~9 million images with image-level person labels, but only ~1.7 million images with a person bounding box. Consequently, this presents a trade-off between more data (image-level

Table 2: Number of images in the Wake Vision dataset

	Person Images	Non-Person Images	Excluded
Wake Vision (Large) Training Dataset	2,880,214	2,880,214	2,176,551
Wake Vision (Quality) Training Dataset	624,115	624,115	6,688,749
Validation Dataset	9,291	9,291	17,824
Test Dataset	27,881	27,881	53,543

labels) or higher-quality labels (bounding box labels). In response, we provide two Wake Vision training sets: Wake Vision (Large), labeled via Open Images image-level labels, and Wake Vision (Quality), a smaller set labeled via Open Images bounding boxes (Table 1). Since the Open Images validation and test sets are fully labeled with higher-quality bounding box labels, we derive the Wake Vision validation and test sets from the bounding box labels in these Open Images splits.

Wake Vision (Large) Training Set Image-level labels in Open Images include a confidence property, which represents the certainty that a label is correct. This confidence ranges from 0 to 10 for machine-generated labels and is strictly either 0 or 10 for human-verified labels. See Section 3.2 and Appendix G for more information. The Wake Vision pipeline can be configured to remove low-confidence machine-labeled person images, which we do for Wake Vision by only including labels with a minimum confidence of 7.

Wake Vision (Quality) Training Set Bounding box labels in Open Images, in contrast to image-level labels, are all verified and localized by humans, minimizing false positive labels. Bounding box labels can be used to calculate an approximation of the area of an image occupied by a person. We use area as a proxy for the distance of a person to the camera, and by default, exclude far-away persons, i.e., those that take up less than 5% of the image (see Figure 4a for an example). Finally, bounding box attributes include an artistic depiction flag (an example is provided in Figure 4b). The Wake Vision pipeline excludes depictions from class members by default; that is, a depiction of a person is not considered a person. See Figure 3 for an illustration of the bounding box filtering process for person detection.

3.2 Label Correction

Label errors are a challenge in computer vision that limit progress and obscure true performance (Northcutt et al., 2021b; Beyer et al., 2020). This is especially prevalent in large datasets that use complex labeling pipelines to scale while managing costs. Recognizing the importance of label quality, we take measures to estimate and improve the accuracy of Wake Vision's labels.

As any labels generated by the Wake Vision pipeline are derived from Open Images v7, errors in these labels are inherited by our datasets. All image-level labels in the Open Images dataset originate from machine-generated candidate labels. Many of these labels are later verified by human annotators, and a subset of them are given bounding boxes and additional annotations.

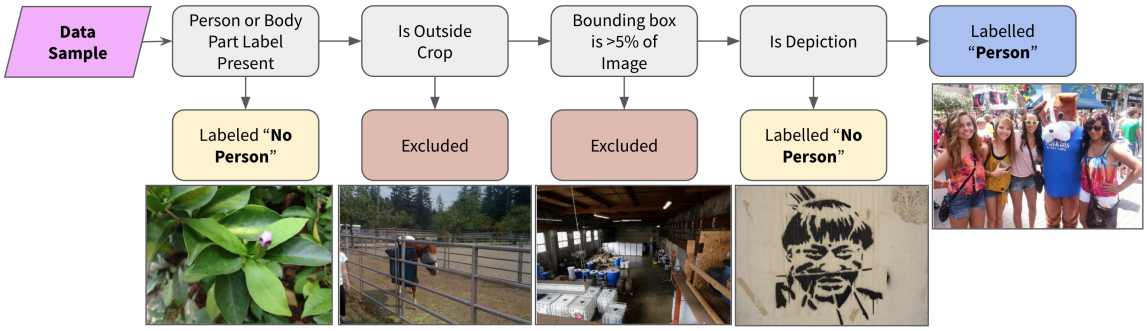


Figure 3: A flowchart of the bounding-box filtering process of an image for person detection.

While the machine-generated label phase aims to identify objects efficiently, any instances missed during this initial step are unlikely to be captured in downstream phases. Consequently, the Open Images dataset may contain numerous false negative labels, referring to images where an object is present but lacks the corresponding label. Therefore, additional measures are necessary to identify and correct such labeling omissions. Given the size of datasets such as Wake Vision, manually correcting label errors across the entire dataset becomes an arduous undertaking.

For our Wake Vision dataset, we therefore prioritize labeling the validation and test sets so that the dataset can be used to accurately evaluate a model's performance. For this task, we use the Scale Rapid tool from Scale AI (`https://scale.com/`) to relabel the Wake Vision validation and test sets. Details about the relabeling process, including instructions provided and the total cost, can be found in Appendix J. With the ground truth established in the validation and test sets, Wake Vision becomes an ideal foundation for research on automated data cleaning techniques (Northcutt et al., 2021a). We perform an initial exploration of these methods in Appendix K.

We report our estimated label error rate after label corrections versus VWW in Table 3. These estimates are all based on a random subset of 500 samples from each dataset, manually evaluated by the authors. The Wake Vision validation and test set estimated error rates are considerably lower than that of the VWW dataset top-level error rate.

3.3 Fine-grained Benchmark Suite

While Wake Vision provides a substantial improvement in scale and quality over existing datasets, we recognize that overall test set performance alone may not capture the nuanced challenges of real-world deployment for TinyML applications. Internet-sourced images typically exhibit a distinct bias towards well-composed photographs that are well-lit, properly framed, and feature clearly visible subjects. This stands in stark contrast to the challenging conditions where TinyML person detection systems are typically deployed, such as varying lighting conditions, extreme viewing distances, or partial occlusions (Plumerai, 2021).

To bridge this gap between benchmark performance and practical utility, we augment Wake Vision with a comprehensive fine-grained benchmark suite. This suite evaluates model robustness across challenging real-world scenarios where traditional accuracy metrics might mask significant failure modes. The benchmarks assess performance across critical

(a) An example of a person far away

(b) An example of a depicted person

Figure 4: Examples of challenging outlier images.

dimensions, including lighting conditions, subject distance, and demographic attributes, enabling developers to identify potential biases or limitations during the design phase rather than after deployment. The suite comprises five fine-grained benchmark sets, three of which (Distance, Lighting, and Depictions) are applicable to any dataset generated by the Wake Vision pipeline, while the other two (Perceived Gender and Perceived Age) are specific to Wake Vision; Figure 5 shows examples from each.

Each of the five fine-grained benchmarks has been picked based on a combination of its relevance to TinyML use cases and the availability of requisite metadata to generate the sets. Each benchmark set is a subset of the respective validation or test set filtered based on the criteria under test. The benchmark determines whether a model is sufficiently accurate in the planned deployment setting. For example, a model designer may make different design choices for a use case where the subject is close to the camera and well-lit compared to the inverse setting. Appendix D contains a case study that shows how fine-grained evaluation sets can be used to design robust models.

Perceived Gender and Age Underrepresented sub-groups typically constitute a small portion of a generic test set; therefore, top-line metrics often obscure a bias in a model until it is deployed (Hooker et al., 2019). This benchmark, generated from the Open Images More Inclusive Annotations for People (MIAP) extended fairness labels (Schumann et al., 2021), evaluates a model separately on demographics that are underrepresented in the underlying dataset distribution, identifying bias. These labels are based on perceived gender and age representation and are not necessarily representative.

Distance This tests how the distance of people in images impacts model performance. If a person detection system is intended to recognize subjects at great distances, the system's performance on the faraway dataset will be more informative than its performance on the top-level test set. We created three datasets based on the percentage of the image the subject bounding box covers. The three sets are near (>60%), at a medium distance (10-60%), and far away (<10%).

Lighting The lighting data sets test the performance of Machine Learning (ML) systems in different lighting conditions. In scenarios like security monitoring, outdoor robotics, or augmented reality applications, models must be robust to varying lighting conditions, including low-light environments. We create three fine-grained datasets of this type for dark,

Table 3: Label error rate of VWW and Wake Vision after label correction. The Wake Vision Train (Quality) and Train (Large) sets have different error rates due to the different label sources. The Wake Vision Validation and Test sets have a lower error rate due to manual relabeling.

Dataset	Wake Vision Set	Label Error
Visual Wake Words	Train, Val, & Test	7.8%
Wake Vision	Train (Large)	15.2%
Wake Vision	Train (Quality)	6.8%
Wake Vision	Val & Test	2.2%

normal, and bright lighting conditions, respectively. We quantify lighting conditions by the average pixel values of images in greyscale, a simple but effective method for distinguishing lighting conditions (Zhang et al., 2017). We define low as an average pixel value less than 85, normal as between 85 and 170, and bright lighting conditions as greater than 170.

Depictions A particularly challenging task for a person detection model is to correctly reject depictions of people. In many use cases, a person detection model should not trigger on a depiction. For example, a room occupancy detector could incorrectly identify a painting on the wall as a person. This benchmark measures a model's accuracy on three related sets of non-person samples: depictions of people, depictions that are not of people, and images that do not contain a depiction of any kind. Depictions of people can range from photo-realistic to crude stick figures.

3.4 Community Accessibility and Involvement

Wake Vision and its fine-grained benchmark suite are available through TensorFlow Datasets (TensorFlow Datasets) and HuggingFace Datasets (Lhoest et al., 2021) to enable easy access and use by the community. The images and labels are rehosted to ensure the dataset will not shrink due to dead links. Removal of images can be requested at the email address reported in Appendix P. The rehosted labels are generated according to our default dataset configuration, further described in Appendix G. Most datasets remain static after the initial release. To continuously improve Wake Vision, we partner with the Edge AI Foundation (EDGE AI FOUNDATION, c) to gather community involvement through recurrent competitions (Section 4.8). It is our vision that the iterative enhancements brought by the community are integrated into the dataset, leading to a continuous improvement cycle.

3.5 Generating Binary Image Classification Datasets for TinyML

Beyond person detection, many other TinyML applications can benefit from large-scale binary classification datasets. For example, bird detection enables smart nest systems (Debauche et al., 2020) and automated feeding (Gerhardson and Min, 2022), while vehicle detection enables automated parking space occupancy monitoring (Bura et al., 2018) and blind spot monitoring (Shen and Yan, 2018). In particular, Open Images v7 (Kuznetsova et al.,

Figure 5: Images from each fine-grained benchmark dataset.

2020; Krasin et al., 2017) contains 9.6K categories and 600 boxable objects, such as birds and vehicles, significantly exceeding the 80 object categories available in MS COCO (Lin et al., 2014). In order to provide users with a quick way to bootstrap a dataset and model for an arbitrary TinyML vision task, the Wake Vision pipeline targets any of the classes in Open Images. We provide a comparison between the Wake Vision pipeline and VWW in Section 4.5.

4 Results

We present a comprehensive evaluation of the Wake Vision dataset and pipeline through the following analyses. First, we compare our two training sets to derive insights into best practices when training. Second, we demonstrate that Wake Vision is an effective drop-in replacement for Visual Wake Words (VWW) by conducting cross-evaluation studies using identical MobileNetV2 models. Third, we validate Wake Vision's benefits across diverse model architectures used in the TinyML community, from MCUNet (Lin et al., 2020) to ColabNAS (Garavagno et al., 2024) families. Fourth, we present fine-grained benchmark results across demographics, environmental conditions, and visual contexts to assess model

Table 4: Accuracy on test set of a MobileNetV2-0.25 model trained on image-level and bounding box labels, with and without KD. A MobileNetV2-1.0 model trained on the bounding box set is the teacher.

Training set	Label Source	Training Set Size	Accuracy	Distilled Accuracy
Wake Vision (Large)	Image-Level	5,760,428	80.80±0.18%	85.00±0.20%
Wake Vision (Quality)	Bounding Box	1,248,230	84.89±0.11%	85.17±0.18%
Wake Vision (Combined)	Both	5,760,428	85.72±0.04%	-

robustness in real-world scenarios. Fifth, we show that the Wake Vision pipeline produces useful binary classification datasets beyond person detection, demonstrating it on bird and car detection. Sixth, we show models trained on Wake Vision maintain their higher performance compared to those trained on VWW when evaluated on several Out-of-Distribution (OOD) datasets, such as driving and overhead surveillance imagery. Seventh, we evaluate the impact of dataset quality and size on low-capacity model training to inform future data-centric efforts. Eighth, we present the results of the first Wake Vision Challenge and the community-driven improvements it has yielded for the Wake Vision dataset. We close with lessons that generalize beyond person detection for data-centric TinyML. Through these analyses, we demonstrate that Wake Vision not only improves upon VWW's performance but also enables a more thorough evaluation of TinyML computer vision models.

4.1 Training Set Evaluation

To evaluate the performance of the two Wake Vision training sets, we train identical models on each set. We use MobileNetV2 (Sandler et al., 2018) models with a width modifier of 0.25 for 200,000 steps on 224x224x3 images using AdamW (Loshchilov and Hutter, 2017), a learning rate of 2e-3 with cosine decay and a weight decay of 4e-6. After training, we compare the performance of each model on the common test set.

The result of this comparison is shown in Table 4. The Wake Vision (Quality) training set outperforms the Wake Vision (Large) training set by 4.09% test accuracy, indicating that label quality is more important than quantity in this evaluation setting. The gap between the two is reduced to just 0.17% when training exclusively on soft labels from a teacher model (MobileNetV2-1.0) trained on bounding box labels. We further study how training set size and quality impact accuracy across model sizes, and observe that smaller models are more sensitive to errors in the training set, dropping accuracy by a delta of 0.8% compared to larger models (Section 4.7).

Using the two training sets in unison to train a model where Wake Vision (Large) acts as a pre-training set, and Wake Vision (Quality) acts as a fine-tuning set achieves our best performance at a mean top-level test accuracy of 85.72%. The two training sets together provide a foundation for further research on data-centric TinyML (Mazumder et al., 2023; Njor et al., 2023).

Table 5: Accuracy on the Wake Vision and VWW test sets by models trained on the VWW, Wake Vision (Quality), and Wake Vision (Combined) training sets.

		Train		
		VWW	Wake Vision (Quality)	Wake Vision (Combined)
Test	VWW	88.33±0.29%	**88.59±0.17%**	**89.34±0.02%**
	Wake Vision	83.79±0.23%	**84.89±0.11%**	**85.72±0.04%**

4.2 Wake Vision & VWW Cross Evaluation

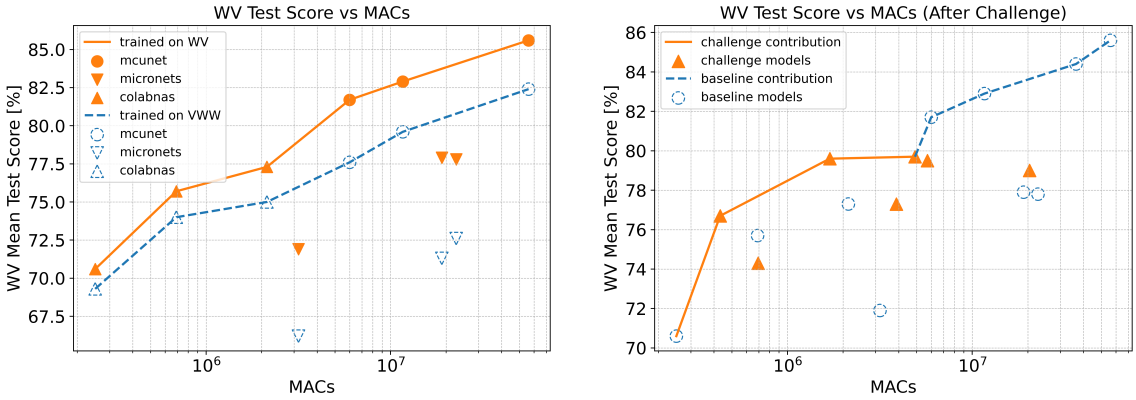
A key challenge in advancing TinyML research is ensuring that improvements can be readily adopted by the community without requiring significant modifications to existing workflows or architectures. While novel datasets can offer better quality or scale, their practical impact is limited if they require substantial changes to model architectures, training pipelines, or deployment processes. We therefore designed Wake Vision to serve as a drop-in replacement for VWW, and validate this drop-in capability through a cross-evaluation study. We train two identical MobileNetV2 (Sandler et al., 2018) models using the same training recipe from Section 4.1. One model uses VWW's training set, and the other uses Wake Vision's Quality training set. After training these identical models using the respective datasets' training sets with the same recipe for an equal number of steps, we evaluate their performance on the corresponding test sets.

Table 5 shows the result of the cross-evaluation. The Wake Vision trained model outperforms the VWW trained model with a 0.26% improvement on VWW's own test set, which indicates that Wake Vision is a direct improvement over VWW, not simply a domain shift. We also achieve a 1.1% improvement over VWW on Wake Vision's Test set, indicating that the new test set is more challenging. Table 5 also shows the performance of the Wake Vision (Combined) model described in Section 3.1. This model achieves 0.75% improvement over the Wake Vision (Quality) trained model on the VWW test set and 0.83% improvement on the Wake Vision test set. The Wake Vision (Combined) model is trained for longer than the VWW and Wake Vision (Quality) models but shows the value of using the training sets in unison even on the VWW test set.

4.3 Evaluation Across TinyML Model Architectures

To show that the benefits of using Wake Vision over VWW are not limited to MobileNetV2 architectures, we also train several other models on both training sets, Wake Vision (Quality) and VWW, and compare their performance. For this experiment, we use a collection of models commonly used in the TinyML community, such as models from the MCUNet (Lin et al., 2020) and MicroNets (Banbury et al., 2021b) families. Furthermore, we include models from ColabNAS (Garavagno et al., 2024) to provide evaluation results on even smaller models.

Each model is trained on each dataset using the Adam optimizer with a learning rate of 1e-3 for 243,800 steps at a batch size of 512, equivalent to 100 epochs on the Wake Vision (Quality) training set. In addition, we apply random horizontal flips, rotations and use 8-bit quantization-aware training. We use the checkpoint with the best validation score for testing. A desktop computer featuring a 13th Gen Intel® Core™ i9-13900K × 32, 32 GB



(a) Initial Pareto frontier for Wake Vision vs VWW

(b) Advanced Pareto frontier by challenge submissions

Figure 6: The Wake Vision Challenge submissions advanced our initial Pareto frontiers.

of RAM, and an NVIDIA GeForce RTX 4080 with 16 GB of VRAM has been employed to train the TinyML models.

We cross-evaluate each trained model's performance on the test sets of both VWW and Wake Vision. Results on the Wake Vision test set are presented in Figure 6a, while further details and the results for the VWW test set can be found in Appendix A. For the majority of models, the one trained on the Wake Vision (Quality) set achieves a higher accuracy on both the VWW and Wake Vision test set, which supports our claim that Wake Vision is a direct improvement over VWW. At the extreme, for one MicroNets (Banbury et al., 2021b) model (middle right of Figure 6a), switching the training set to Wake Vision increased the test accuracy by as much as 6.6%. The only instance where Wake Vision isn't a complete improvement over VWW is in Figure 11 in Appendix A, where small models perform better when trained on the same dataset they are tested on (i.e., trained on VWW and tested on VWW), likely due to ultra-low-capacity models failing to generalize past the slight domain shift between the VWW and Wake Vision test sets. Accuracy and performance metrics, such as latency or model size, are in constant tension in TinyML models, so any boost in accuracy achieved through Wake Vision can also be traded off for more efficient architectures (in some cases 3× fewer MACs) while preserving test accuracy.

4.4 Fine-Grained Benchmark Evaluation

Aggregate metrics like accuracy or F1-score provide high-level insight but can mask critical failure modes and biases that emerge in real-world scenarios. In response, we include fine-grained benchmarks in Wake Vision (Section 3.3) across demographics, environmental conditions, and visual contexts. This is particularly important for person detection models deployed in real-world applications, where failures on certain subgroups or conditions could adversely affect fairness and safety.

In Table 6, we compare the models from Section 4.2, trained on Wake Vision and VWW, across various scenarios to understand their relative strengths and limitations. The Wake Vision model exhibits superior robustness across the challenging settings exercised by our benchmarking suite. For instance, on the "Depictions" benchmark, which evaluates

Table 6: Wake Vision Fine-Grained Benchmark Suite. We report the samples in each set and the average F1 score across three Wake Vision models and three VWW models on each benchmark.

		Gender			Age				Distance			Lighting			Depictions		
		Female	Male	Unknown	Young	Middle	Older	Unknown	Near	Medium	Far	Dark	Normal	Bright	Person	Non-Person	No Depiction
Size	Val	684	1310	1612	275	2133	90	1299	5457	2213	398	3255	14315	1012	356	352	8583
	Test	2157	3918	4940	884	6595	276	3837	16333	6876	1140	9420	43010	3332	978	1101	25802
F1	Wake Vision	**0.93**	**0.91**	0.77	**0.94**	**0.91**	**0.94**	0.71	**0.91**	**0.85**	0.59	**0.85**	**0.82**	**0.82**	**0.71**	**0.86**	**0.87**
	VWW	0.89	0.88	**0.78**	0.90	0.88	0.89	**0.74**	0.89	0.84	**0.67**	0.81	**0.82**	**0.82**	0.66	0.82	0.85

performance on images containing persons, non-person objects, or no depictions, the Wake Vision model achieves an F1 score of 0.71 for person depictions, outperforming the VWW model's 0.66. The Wake Vision model also showcases improved performance on the "Age" benchmark, with F1 scores of 0.94, 0.91, and 0.94 for young, middle, and older individuals, surpassing the VWW model's scores of 0.90, 0.88, and 0.89, respectively. This highlights the dataset's effectiveness in enhancing model robustness for detecting people across various age groups, particularly the elderly demographic. In addition, the Wake Vision model demonstrates resilience to challenging lighting conditions, achieving F1 scores of 0.85 in dark lighting scenarios, compared to the VWW's 0.81.

Notably, a VWW model only significantly outperforms a Wake Vision model on the far distance benchmark at a score of 0.67 and 0.59, respectively. During dataset generation, VWW filters out persons smaller than 0.5% of the image, versus 5% for Wake Vision. Therefore, VWW contains images with significantly smaller persons. We did experiment with adopting a default of 0.5% for Wake Vision, however this disproportionately decreased the performance of models in other settings.

Across the five fine-grained axes, Wake Vision-trained models match or exceed VWW-trained models on 13 of 16 subsets and show the largest gains where TinyML deployment conditions diverge most from typical training imagery: depicted persons, older individuals, and dark lighting.

4.5 Generalized Binary Image Classification Datasets

We demonstrate that the Wake Vision pipeline can be used for constructing binary image classification datasets for any category of interest among the 9.6K trainable classes or 600 boxable objects present in Open Images v7 (Kuznetsova et al., 2020; Krasin et al., 2017). Datasets constructed by the Wake Vision pipeline are significantly larger than those produced by the prior VWW methodology applied to the same class. In particular, we compare the Wake Vision pipeline against VWW on *birds* and *cars*, as these two categories are shared across both Open Images and the 80 object categories available in MS COCO. As noted in Section 3.5, bird detection has applications in smart nests (Debauche et al., 2020) and automated feeders (Gerhardson and Min, 2022), while car detection applies to parking monitoring (Bura et al., 2018) and blind-spot systems (Shen and Yan, 2018). The car detection dataset generated via the Wake Vision pipeline contains 12× more images

Table 7: Sample Counts of Car and Bird datasets.

Methodology	# of Images	Label Error Rate
Car		
Wake Vision (Ours)	157,668	4.8%
VWW (Chowdhery et al., 2019)	12,902	6.4%
Bird		
Wake Vision (Ours)	31,642	0.6%
VWW (Chowdhery et al., 2019)	1,148	6.6%

than VWW, with a decrease in label error rate from 6.4% to 4.8%. Similarly, the bird detection dataset provides 27× more images with a decrease from 6.6% to 0.6% in label error rate.

Table 7 compares the number of images as well as the label error rate for the car and bird datasets, when generated using both our methodology and the VWW one. All datasets are balanced, i.e., have the same number of images for the target and background class. Label error rates have been manually computed on a subset of 500 randomly selected images for each generated dataset. Examples of positive class images from the Wake Vision test splits are shown in Figure 7.

4.6 Accuracy under Data Distribution Shifts

In order to assess the generalizability of Wake Vision to other data distributions, we simulate data distribution shifts (Beery et al., 2018; Taori et al., 2020) by running inference on a selection of three new datasets using the models already trained on Wake Vision and VWW from Section 4.3 and computing Pareto curves on these new datasets. In particular, we seek to emulate downstream applications of person detection (such as CCTV surveillance) when using Wake Vision for training data. The upstream data sources of Open Images for Wake Vision and MS-COCO for VWW exhibit image characteristics common in handheld photographs, often with center-biased locations of nearby subjects that are large in frame and with minimal barrel distortion. We wish to evaluate how Wake Vision performs under data distribution shifts, such as with surveillance imagery which is often taken from an overhead perspective in challenging lighting conditions and in the presence of fisheye lens distortion.

We select the TUD-MotionPairs dataset (Wojek et al., 2009) (2,185 positives and 385 negatives) containing imagery from a vehicle's driving perspective to emulate pedestrian or cyclist detection, and two Kaggle datasets containing surveillance imagery from distant and overhead perspectives: Surveillance Images for Person Detection (Martins, 2025) (5,265 positives and 1,338 negatives) and Human Detection Dataset (Werner, 2025) (559 positives and 362 negatives). Randomly selected examples from each dataset are shown in Figure 8. Due to their unusual perspective, lighting, and lens properties, these three datasets could be considered out-of-distribution (OOD) with respect to most images in Open Images and MS-COCO used when training the models in Section 4.3. We evaluate these pretrained

(a) Wake Vision "Car" samples

(b) Wake Vision "Bird" samples

Figure 7: Positive class examples from the test splits of our Wake Vision generated datasets for "Car" and "Bird" objects, demonstrating our ability to easily prototype large-scale TinyML vision datasets (Section 3.5).

models on our three selected OOD datasets to compare the performance of Wake Vision-trained models against VWW-trained models under data distribution shift at inference time. Figure 9 presents our results. Models trained on Wake Vision generally outperform models trained on VWW under these distribution shifts. This suggests that Wake Vision's larger scale and wider diversity of image characteristics improve generalization, though the size of the advantage varies with the target distribution.

4.7 Impact of Dataset Quality & Size in TinyML

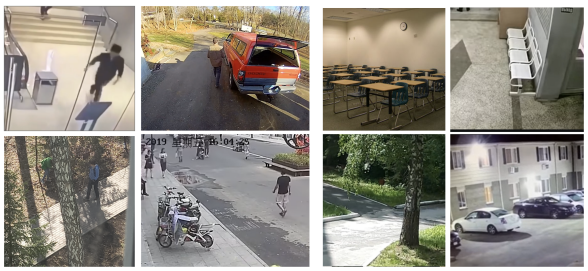
Understanding the relationship between dataset quality and size is important for training efficient models in real-world scenarios. While gathering large amounts of data has become easier, ensuring high-quality labeling remains costly and time-consuming. This creates an important trade-off: is it better to have a smaller dataset with high-quality labels or a larger dataset with more label noise? The answer likely depends on the capacity of the model and the specific task.

We investigate the impact of dataset quality and size on Resnet (He et al., 2016) style models of varying capacity. We measure quality by the approximate rate of label errors. The Wake Vision (Quality) training set has an estimated error rate of approximately 6.8% (Table 3). We simulate higher error rate datasets (15% and 30%) by flipping the binary label with a certain probability. Appendix C gives the derivation for the flip probability. We also sweep the dataset size by taking a slice of Wake Vision (Quality). Each model is trained for 50,000 steps on 224x224x3 images using AdamW (Loshchilov and Hutter, 2017), a learning rate of 0.001 with a cosine schedule and a weight decay of 0.004.

Figure 10 shows that smaller models (leftmost figure) are more sensitive to higher error rates than large models, similar to prior findings that model capacity can help memorize

(a) TUD-MotionPairs

(b) Kaggle1



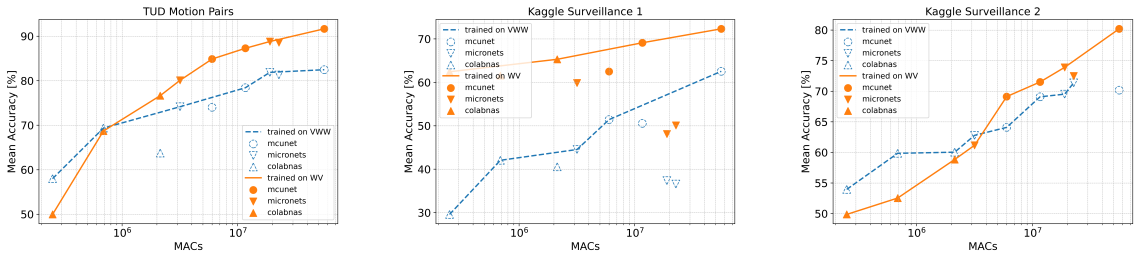
(c) Kaggle2

Figure 8: Four positive and four negative examples drawn randomly from each of our three Out-of-Distribution datasets.

and ignore errors and outliers (Zhang et al., 2021; Feldman, 2020; Goyal et al., 2024). The lowest accuracy drop in the smallest model is 1.3%, going from the base train (quality) error rate of approximately 6.8% to 15% compared to only a 0.5% accuracy loss for the largest model. In contrast, large models benefit more from big datasets and start to overfit on the smaller slices of the training data (Kaplan et al., 2020). We also see the large slice of the 15% error rate dataset outperforming a smaller slice of the 6.8% error rate dataset.

4.8 The Wake Vision Flywheel: Community-Driven Continuous Improvement

Data-centric TinyML treats the dataset, not just the model, as a maintained artifact. Yet most TinyML datasets are released once and then frozen, so label errors, distribution gaps, and missing edge cases discovered post-release have nowhere to go. Wake Vision instead employs a community-driven flywheel in which competitions surface targeted dataset and model improvements that the maintainer integrates into subsequent releases, turning the dataset into a renewable input for data-centric TinyML research rather than a static bench-



(a) TUD-MotionPairs (b) Kaggle1 (c) Kaggle2

Figure 9: Pareto frontiers for Wake Vision vs VWW on Out-of-Distribution datasets

mark. We partner with the Edge AI Foundation (EDGE AI FOUNDATION, c) to host Wake Vision competitions. The competitions held in 2025 focused on two tracks, one on developing high-performance, resource-efficient models, and the other on enhancing the Wake Vision person detection dataset, garnering significant support from the TinyML community with 67 participants across both tracks (EDGE AI FOUNDATION, a,b).

The first round produced two concrete data-centric improvements. On the model track, three new submissions advanced the Wake Vision Pareto frontier (orange triangles in Figure 6b), tightening the accuracy-versus-cost trade-off available to TinyML practitioners on this dataset. On the data track, a participant's label-correction technique (Bedioui, 2025), applied to the Wake Vision (Large) split, reduced its estimated label error rate from 15.2% to 9.8% in our experiments (Appendix O), a substantial single-round improvement to a property that conventionally requires expensive manual relabeling. The implication generalizes beyond person detection. Once a dataset is positioned as a maintained artifact, even a small community can supply data-centric improvements that the original authors would not produce alone, and at a cost orders of magnitude below manual relabeling. New competitions are being developed for the upcoming year, including integration with machine learning educational projects such as MLSys Book (Reddi, 2024) to engage a new generation of engineers and to keep the data-centric flywheel turning.

4.9 Lessons for Data-centric TinyML

The Wake Vision results yield four lessons that generalize beyond person detection and that we believe apply broadly to data-centric TinyML.

First, **small models are disproportionately sensitive to training-set label noise**. In our injection protocol (Section 4.7), the smallest model loses 1.3% accuracy when training error rises from 6.8% to 15%, versus 0.5% for the largest. When the deployment target is small, label-quality returns dominate scale returns earlier than work on large models suggests.

Second, **two-stage training combines the strengths of a noisy-large set and a clean-small set**. Pretraining on Wake Vision (Large) and fine-tuning on Wake Vision (Quality) reaches 85.72% test accuracy, versus 84.89% for Quality alone and 80.80% for Large alone (Section 4.1).

Third, **the manual-labeling budget is best spent where it most affects reported accuracy, not uniformly across the dataset**. Relabeling approximately 70K validation

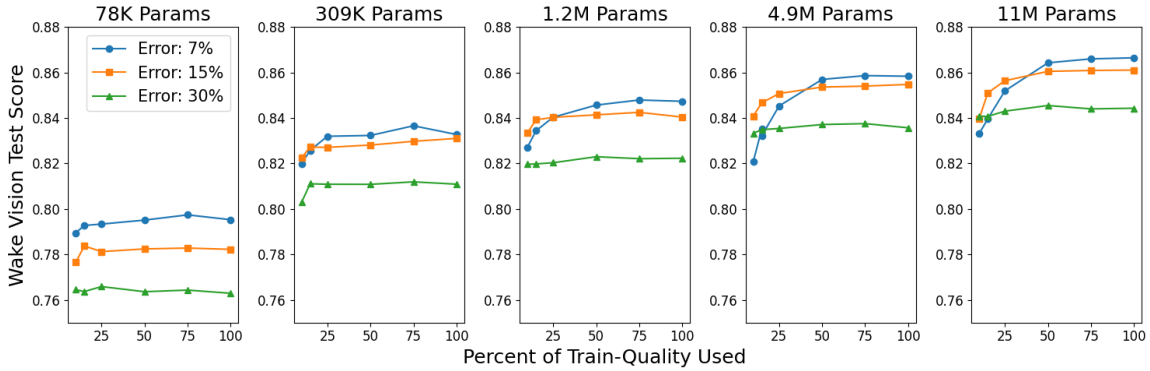


Figure 10: Impact of dataset error rate and size on models of varying capacity. The y-axis is the Wake Vision test accuracy and the x-axis is the percentage of Wake Vision Train (Quality) used in training. The error % indicates the expected rate of incorrectly labeled samples; the 7% line corresponds to the baseline Wake Vision (Quality) error rate of approximately 6.8% (Table 3).

and test images cost $7,000, while relabeling a 6M training set would cost roughly $600,000 (Section 3.2). For data-centric pipelines at scale, the practical question is not whether to manually relabel but where each labeled sample has the largest effect.

Fourth, **community competitions are a credible mechanism for continuous dataset improvement**. A single round of the Wake Vision Challenge produced a label-correction technique that reduced the Wake Vision (Large) label error rate from 15.2% to 9.8% in our experiments (Section 4.8). The cost economics that drive Lesson 3 also apply here. When manual relabeling at training-set scale is prohibitive, sourcing improvements from a community of practitioners is one of the few mechanisms that can keep dataset quality moving in the right direction over time.

5 Broader Impact Statement

Releasing a large-scale person detection dataset, a benchmark suite that includes demographic axes, and the Wake Vision pipeline that lowers the cost of building further such datasets carries responsibilities that the empirical results above do not address. We outline these by category below.

Potential Misuse: The Wake Vision pipeline, dataset, and benchmark suite are designed to advance TinyML research and applications. Intended benefits include occupancy detection systems that conserve energy by turning off the lights/HVAC/TV when a person leaves, and privacy preservation through on-device inference, avoiding cloud processing of camera data. We acknowledge that person detection technology can be misappropriated for harmful purposes such as weapons targeting or mass surveillance systems.

Data Rights and Privacy: Images are sourced from Flickr through Open Images under the CC-BY 2.0 license. Some images may have been uploaded without proper distri-

bution rights or may contain potentially identifying biometric information. Removal of images can be requested at the email address reported in Appendix P. Due to the scale of the dataset, exhaustive manual verification for offensive content or privacy concerns is infeasible.

Fairness and Representation: Our benchmark suite enables disaggregated evaluation across perceived demographics and deployment conditions, helping developers detect failure modes that aggregate accuracy obscures. Disaggregated evaluation does not by itself guarantee equitable deployment, and the suite should be read as a diagnostic tool rather than a certification of fairness; it inherits the demographic categories and limitations of MIAP (Schumann et al., 2021), where attributes are externally perceived rather than self-reported. Demographic benchmarks can also be misused to classify gender and age without consent, potentially enabling privacy violations or discriminatory practices. We release the suite for diagnostic evaluation of person detection models, not as endorsement of deploying demographic classifiers.

6 Conclusion

We presented the Wake Vision pipeline, an automated method for generating and curating large-scale binary classification datasets for TinyML, combining multi-source label fusion, confidence-, area-, and depiction-aware filtering, label correction on the evaluation splits, and fine-grained benchmark synthesis. Applying it to person detection, we released Wake Vision, a dataset of almost 6M images (close to 100× more person images than VWW) whose manually relabeled validation and test sets achieve a 2.2% label error rate, compared to 7.8% reported for VWW. Wake Vision-trained models improve test accuracy by up to 6.6% over VWW across MobileNetV2, MCUNet, MicroNets, and ColabNAS architectures, match or exceed VWW-trained models on 13 of 16 fine-grained subsets, and retain their advantage on three out-of-distribution datasets covering driving and overhead-surveillance imagery.

Two findings shape data-centric TinyML beyond person detection. First, small models are disproportionately sensitive to training-set label noise. The smallest model loses 1.3% accuracy when training error rises from 6.8% to 15%, versus 0.5% for the largest. Second, two-stage training, i.e., pretraining on the larger noisier set and fine-tuning on the smaller cleaner one, reaches 85.72% test accuracy versus 84.89% and 80.80% for the constituent sets alone.

Looking ahead, we see Wake Vision opening three lines of work for data-centric TinyML. The first is the long tail of bespoke TinyML tasks. Most TinyML deployments target binary or low-class-count problems for which no public dataset exists, and the Wake Vision pipeline reaches any of the 9.6K trainable classes of Open Images v7 (Section 3.5). Bird and car detection are first demonstrations; we expect the pipeline to remove the dataset bottleneck for many more such tasks and to make per-task dataset generation a routine step in the TinyML workflow. The second is treating the dataset itself as a maintained artifact. The first Wake Vision Challenge already reduced the Wake Vision (Large) label error rate from 15.2% to 9.8% in a single round, at a fraction of the $600,000 implied by manual relabeling at this scale. Subsequent rounds, hosted with the Edge AI Foundation, are intended to keep improving label quality and to fold winning contributions into successive Wake Vision

releases, so that progress on the dataset accumulates rather than restarting with each paper. The third is a research agenda the artifacts make tractable. The label-noise sensitivity above means dataset quality cannot be treated as a solved problem just because larger models tolerate it; tiny models need their own studies. A 6M-image dataset with manually relabeled evaluation splits, fine-grained subsets along demographic and environmental axes, and out-of-distribution probes provides a substrate on which the community can study label-quality-vs-scale trade-offs, robustness, and fairness for tiny models in a controlled way. We release the dataset, benchmarks, code, and the pipeline that produced them under CC-BY 4.0 (`https://wakevision.ai`), and invite the TinyML community to build on Wake Vision and to extend the data-centric TinyML toolkit to the tasks beyond person detection where it is most needed.

Acknowledgments and Disclosure of Funding

The Google TPU Research Cloud program partially supported this work via cloud computing credits. The PhD stipend of Emil Njor was supported by the Innovation Fund Denmark DIREC project (9142-00001B). Emil Njor's external research stay was furthermore supported by Fulbright Denmark, Stibo Fonden, Thomas. B. Thriges Fond, Otto Mønsteds Fond and Kaj og Hermilla Ostenfelds Fond. This work was also supported by the National Science Foundation (NSF) and the Semiconductor Research Corporation (SRC).

References

Colby Banbury, Vijay Janapa Reddi, Peter Torelli, Jeremy Holleman, Nat Jeffries, Csaba Kiraly, Pietro Montino, David Kanter, Sebastian Ahmed, Danilo Pau, et al. Mlperf tiny benchmark. *arXiv preprint arXiv:2106.07597*, 2021a.

Colby Banbury, Chuteng Zhou, Igor Fedorov, Ramon Matas, Urmish Thakker, Dibakar Gope, Vijay Janapa Reddi, Matthew Mattina, and Paul Whatmough. Micronets: Neural network architectures for deploying tinyml applications on commodity microcontrollers. *Proceedings of Machine Learning and Systems*, 3:517–532, 2021b.

Colby R Banbury, Vijay Janapa Reddi, Max Lam, William Fu, Amin Fazel, Jeremy Holleman, Xinyuan Huang, Robert Hurtado, David Kanter, Anton Lokhmotov, et al. Benchmarking tinyml systems: Challenges and direction. *arXiv preprint arXiv:2003.04821*, 2020.

Kais Bedioui. Wake vision challenge data-centric track kais bedioui submission. `https://github.com/kais-bedioui/Wake_Vision_Challenge_Data_Centric_Track`, 2025. Accessed: 2025-05-06.

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018.

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

Harshitha Bura, Nathan Lin, Naveen Kumar, Sangram Malekar, Sushma Nagaraj, and Kaikai Liu. An edge based smart parking solution using camera networks and deep learning. In *2018 IEEE international conference on cognitive computing (ICCC)*, pages 17–24. IEEE, 2018.

Aakanksha Chowdhery, Pete Warden, Jonathon Shlens, Andrew Howard, and Rocky Rhodes. Visual wake words dataset. *arXiv preprint arXiv:1906.05721*, 2019.

Robert David, Jared Duke, Advait Jain, Vijay Janapa Reddi, Nat Jeffries, Jian Li, Nick Kreeger, Ian Nappier, Meghna Natraj, Tiezhen Wang, et al. Tensorflow lite micro: Embedded machine learning for tinyml systems. *Proceedings of Machine Learning and Systems*, 3:800–811, 2021.

Olivier Debauche, Saïd Mahmoudi, Abdelaziz Marzak, Pierre Manneback, Frédéric Lebeau, et al. Smart nest box: Iot based nest monitoring in artificial cavities. In *2020 3rd International Conference on Advanced Communication Technologies and Networking (CommNet)*, pages 1–7. IEEE, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Katie E Doull, Carl Chalmers, Paul Fergus, Steve Longmore, Alex K Piel, and Serge A Wich. An evaluation of the factors affecting 'poacher'detection with drones and the efficacy of machine-learning for detection. *Sensors*, 21(12):4074, 2021.

EDGE AI FOUNDATION. Challenge edge: Wake vision. `https://edgeai.modelnova.ai/challenges/details/challenge-edge:-wake-vision`, 2025a. Accessed: 2025-05-06.

EDGE AI FOUNDATION. Challenge edge: Wake vision. `https://edgeai.modelnova.ai/challenges/details/edge-ai-challenge:-wake-vision-2`, 2025b. Accessed: 2025-05-07.

EDGE AI FOUNDATION. Edge ai foundation. `https://www.edgeaifoundation.org/`, 2025c. Accessed: 2025-05-06.

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

Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, 2020.

Flickr. https://flickr.com/, 2024. Accessed: 2024-03-06.

Andrea Mattia Garavagno, Daniele Leonardis, and Antonio Frisoli. Colabnas: Obtaining lightweight task-specific convolutional neural networks following occam's razor. *Future Generation Computer Systems*, 152:152–159, 2024.

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.

Christopher Gerhardson and Cheol-Hong Min. Design and implementation of an iot wireless sensor network for bird feeder monitoring. In *2022 IEEE 13th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 0124–0128. IEEE, 2022.

Sachin Goyal, Pratyush Maini, Zachary C Lipton, Aditi Raghunathan, and J Zico Kolter. Scaling laws for data filtering–data curation cannot be compute agnostic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22702–22711, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Geremy Heitz, Stephen Gould, Ashutosh Saxena, and Daphne Koller. Cascaded classification models: Combining models for holistic scene understanding. *Advances in neural information processing systems*, 21, 2008.

Sara Hooker, Aaron Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. What do compressed deep neural networks forget? *arXiv preprint arXiv:1911.05248*, 2019.

Sara Hooker, Nyalleng Moorosi, Gregory Clark, Samy Bengio, and Emily Denton. Characterising bias in compressed models. *arXiv preprint arXiv:2010.03058*, 2020.

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 preprint arXiv:2001.08361*, 2020.

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.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Masters Thesis*, 2009.

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*, 2020.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, et al. Datasets: A community library for natural language processing. *arXiv preprint arXiv:2109.02846*, 2021.

Roskilde University Library. Creative commons - copyright - libguides at roskilde university library. `https://libguides.ruc.dk/copyright/cc`, 2024.

Ji Lin, Wei-Ming Chen, Yujun Lin, Chuang Gan, Song Han, et al. Mcunet: Tiny deep learning on iot devices. *Advances in Neural Information Processing Systems*, 33:11711–11722, 2020.

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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Luís C. R. Martins. Surveillance images for person detection. Kaggle dataset, 2025. `https://www.kaggle.com/datasets/luiscrmartins/surveillance-images-for-person-detection`.

Mark Mazumder, Colby Banbury, Xiaozhe Yao, Bojan Karlaš, William Gaviria Rojas, Sudnya Diamos, Greg Diamos, Lynn He, Alicia Parrish, Hannah Rose Kirk, et al. Dataperf: Benchmarks for data-centric ai development. *Advances in Neural Information Processing Systems*, 36, 2023.

Emil Njor, Jan Madsen, and Xenofon Fafoutis. Data aware neural architecture search. *arXiv preprint arXiv:2304.01821*, 2023.

Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021a.

Curtis G Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. *arXiv preprint arXiv:2103.14749*, 2021b.

Mateusz Piechocki, Marek Kraft, Tomasz Pajchrowski, Przemyslaw Aszkowski, and Dominik Pieczynski. Efficient people counting in thermal images: The benchmark of resource-constrained hardware. *IEEE Access*, 10:124835–124847, 2022.

Plumerai. Great tinyml needs high-quality data — plumerai blog. `https://blog.plumerai.com/2021/08/tinyml-data/`, August 2021. (Accessed on 11/13/2024).

Vijay Janapa Reddi. Mlsysbook.ai: Principles and practices of machine learning systems engineering. In *2024 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ ISSS)*, pages 41–42. IEEE, 2024. URL `https://mlsysbook.org`.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

Candice Schumann, Susanna Ricco, Utsav Prabhu, Vittorio Ferrari, and Caroline Rebecca Pantofaru. A step toward more inclusive people annotations for fairness. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, 2021.

Yiting Shen and Wei Qi Yan. Blind spot monitoring using deep learning. In *2018 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pages 1–5. IEEE, 2018.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33:18583–18599, 2020.

TensorFlow Datasets. TensorFlow Datasets, a collection of ready-to-use datasets. `https://www.tensorflow.org/datasets`, 2024.

Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

Pete Warden and Daniel Situnayake. *Tinyml: Machine learning with tensorflow lite on arduino and ultra-low-power microcontrollers*. O'Reilly Media, 2019.

Constantin Werner. Human detection dataset. Kaggle dataset, 2025. `https://www.kaggle.com/datasets/constantinwerner/human-detection-dataset`.

Christian Wojek, Stefan Walk, and Bernt Schiele. Multi-cue onboard pedestrian detection. In *2009 IEEE conference on computer vision and pattern recognition*, pages 794–801. IEEE, 2009.

Angelos Zacharia, Dimitris Zacharia, Aristeidis Karras, Christos Karras, Ioanna Giannoukou, Konstantinos C Giotopoulos, and Spyros Sioutas. An intelligent microprocessor integrating tinyml in smart hotels for rapid accident prevention. In *2022 7th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, pages 1–7. IEEE, 2022.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

WenLi Zhang, HongLu Li, and ZhuoZheng Wang. Research on different illumination image classification method. In *2017 2nd International Conference on Automation, Mechanical Control and Computational Engineering (AMCCE 2017)*, pages 574–581. Atlantis Press, 2017.

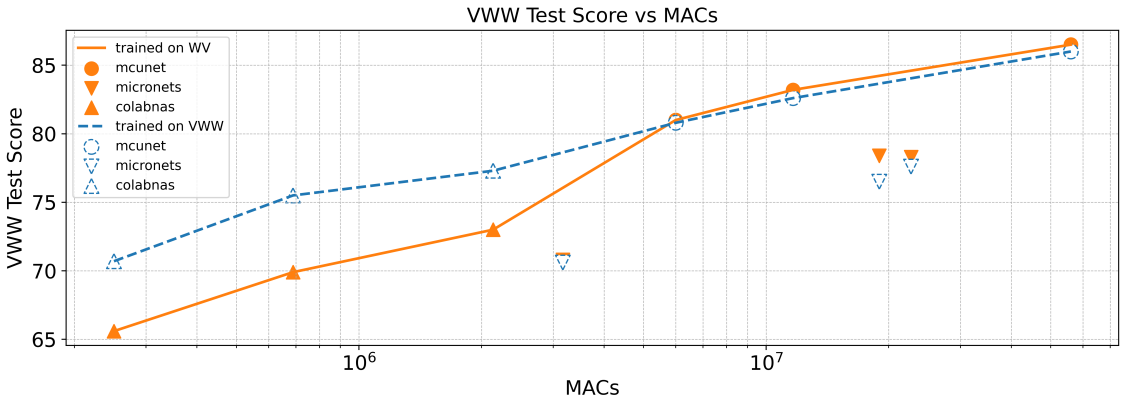


Figure 11: Cross-evaluation results on the Visual Wake Words test set

Appendix A. Additional Benchmark Results

Table 8 provides input shape, Flash, RAM, and MACs for each model presented in section 4.2 as well as the mean and standard deviation of test accuracies for both Wake Vision and VWW datasets. RAM and Flash have been measured using the "stm32tflm" utility of X-CUBE-AI 8.1.0, whereas MACs with "stm32ai". Links pointing to the original models used for training are also present for each model family involved in the experiment.

Figure 11 shows cross-evaluation results on the Visual Wake Words test set. The majority of models trained on the Wake Vision dataset obtain better test accuracy on VWW than the ones trained on VWW. Only the smallest models do not achieve a higher test accuracy when trained on Wake Vision, likely due to a lack of learning capacity to overcome the slight domain shift between the two datasets.

Appendix B. Code Repository

The code used to generate the Wake Vision dataset is available at the following GitHub repository: `https://github.com/harvard-edge/Wake_Vision`

This repo contains the code to generate Wake Vision and the benchmark suite, as well as the code to train and evaluate models. This code is sufficient to reproduce all results in the paper.

Appendix C. Inducing Label Errors for Quality & Size Evaluation

The goal is to make a single pass through the dataset and flip the labels of a binary classification dataset such that the expected label error rate is d. There is an underlying rate of label errors e. If we flip one of these underlying errors, we correct it, thereby inadvertently decreasing the label error rate. After flipping labels with a probability of p we can claim that the likelihood of a single label being correct is the probability that we flipped the label and it was originally an error plus the probability that we didn't flip the label and it was originally correct: $1 - d = p * e + (1 - p)(1 - e)$. Then solving for p gives

Table 8: Raw data of reconstructed models.

Model .tflite	RAM [kiB]	Flash [kiB]	MACs [MM]	Test Set	Training Set WV (Qual.)	Training Set VWW
MobileNetV2 (Sandler et al., 2018)						
MobileNetV2_0.25	1,244.5	410.55	36,453,732	**WV**	84.9 ± 0.11	83.8 ± 0.23
				VWW	88.6 ± 0.17	88.3 ± 0.29
MCUNet (Lin et al., 2020)						
10fps_vww	168.5	533.84	5,998,334	**WV**	81.7 ± 0.28	77.6 ± 0.31
				VWW	81 ± 0.12	80.8 ± 0.1
5fps_vww	226.5	624.55	11,645,502	**WV**	82.9 ± 0.29	79.6 ± 0.26
				VWW	83.2 ± 0.57	82.6 ± 0.16
320kb-1mb_vww	393	923.76	56,022,934	**WV**	85.6 ± 0.34	82.4 ± 0.62
				VWW	86.5 ± 1.01	86 ± 0.25
MicroNets (Banbury et al., 2021b)						
vww2_50_50	71.50	225.54	3,167,382	**WV**	71.9 ± 0.67	66.2 ± 0.88
				VWW	70.8 ± 0.68	70.6 ± 0.64
vww3_128_128	137.50	463.73	22,690,291	**WV**	77.8 ± 0.56	72.6 ± 1.01
				VWW	78.3 ± 0.91	77.6 ± 1.13
vww4_128_128	123.50	417.03	18,963,302	**WV**	77.9 ± 0.6	71.3 ± 1.03
				VWW	78.4 ± 0.51	76.5 ± 0.37
ColabNAS (Garavagno et al., 2024)						
k_2_c_3	18.5	7.66	250,256	**WV**	70.6 ± 0.96	69.3 ± 0.97
				VWW	65.6 ± 0.66	70.7 ± 0.08
k_4_c_5	22	18.49	688,790	**WV**	75.7 ± 0.18	74 ± 0.23
				VWW	69.9 ± 0.26	$75.5 \pm 0,64$
k_8_c_5	32.5	44.56	2,135,476	**WV**	77.3 ± 0.37	75 ± 0.15
				VWW	73 ± 0.91	77.3 ± 0.57

$p = (e - d)/(2e - 1)$. A current flaw of this method is that the injected label errors are not consistent between epochs, which would likely be less destructive to a model's accuracy since the same errors are not reinforced each epoch. This could also potentially explain why large models in Figure 10 don't overfit on training data with higher label errors, as the inconsistent label noise has a regularizing effect.

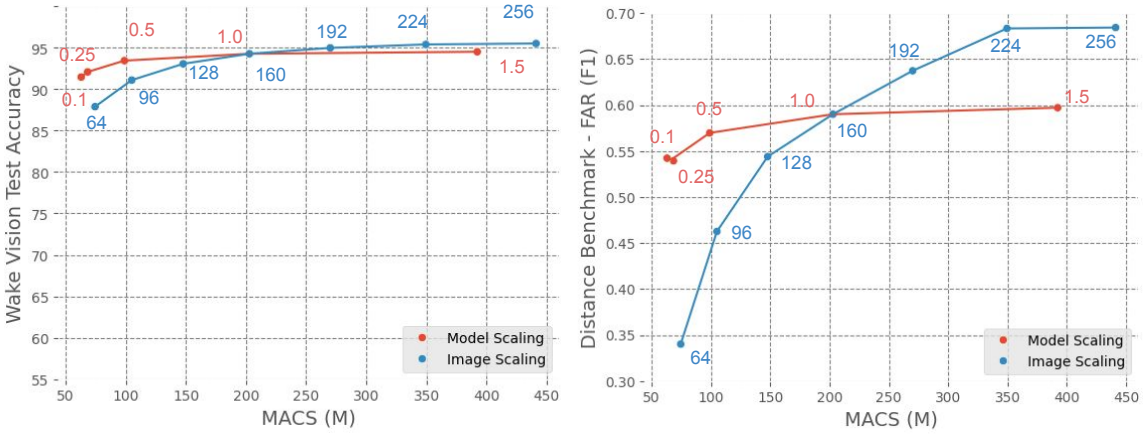


Figure 12: Effects of scaling the image size vs. the model width on Wake Vision test accuracy and the Distance-Far benchmark set. The Distance-Far F1 score (right) is far more sensitive to changes in the image size than the overall test accuracy (left). Image sizes refer to square image resolutions (e.g. 96x96) and model size refers to the MobileNetV2 width multiplier. When not specified the image size is 160 and the width multiplier is 1.0.

Appendix D. Model Design Case Study

Basic test set performance can often misrepresent a model's performance, given the typical domain shift between images scraped from the internet and real-world use cases. This issue is exacerbated when ML practitioners must trade off accuracy for model performance and size, which is necessary for TinyML use cases. A design decision might have seemingly little impact on the test accuracy but may destroy real-world performance depending on the deployment environment. For example, a person detection system may only operate in dark lighting conditions, but the test dataset has an insignificant number of dark samples; therefore, the test accuracy will not reflect the real-world accuracy.

The benchmark suite enables more holistic analysis during the design phase. To show this use we perform a series of scaling experiments employing typical TinyML compression techniques and identifying under which circumstances these techniques are appropriate. While these results can inform ML practitioners, our intention is to demonstrate the usefulness of the benchmark suite.

D.1 Image Size vs. Model Width Scaling

We train two series of MobileNetV2 models: one series that sweeps the input image size [64-256] and one that sweeps the width multiplier of a model [0.1-1.5]. We then benchmark these models on the Wake Vision test set as well as the far distance benchmark. We plot these results against the number of multiply-accumulate (MAC) operations in the model as a proxy for on-device latency (Banbury et al., 2021b).

The results in Figure 12 (left) show that when looking exclusively at the high-level metric (i.e., test accuracy), scaling the input image size has a similar impact as scaling the model size. However, as shown in Figure 12 (right), when we consider only samples where

the person is far away from the camera (i.e., the distance benchmark), we observe a much more significant impact when scaling the image size. In the case of distant subjects, the image size becomes the bottleneck.

These findings suggest that for ML developers targeting use cases where the subject is likely to be far from the camera, prioritizing larger input image sizes over wider models may be more beneficial. However, this critical design consideration could be obscured when solely relying on high-level metrics like overall test accuracy. The distance benchmark in Wake Vision effectively unveils the disproportionate impact of image size on model performance for distant subjects, enabling more informed decision-making during model optimization.

D.2 Quantization

Quantization is a crucial technique for deploying efficient TinyML models, offering substantial benefits in terms of reduced latency, memory footprint, and model size. However, prior work has suggested that quantization can disproportionately impact the performance of models on underrepresented subsets of data (Hooker et al., 2020). To assess the implications of quantization in the context of person detection, we investigate the impacts of int8 quantization on a model's benchmark results across Wake Vision's fine-grained benchmarks.

Our findings show negligible degradation in performance across all benchmarks (± 0.004 F1) when employing int8 quantization, even on outlier sets. This result contradicts the previously observed disproportionate impact of quantization on underrepresented subsets. We speculate that person detection may be a relatively simple task, potentially explaining why we do not observe this specific property of quantization in our experiments. Given the negligible performance degradation and the substantial latency, memory, and model size benefits of quantization, we conclude that quantization is well-suited for person detection.

D.3 Grayscale

Converting a model's input image channels to grayscale from RGB is a commonly employed optimization in the TinyML field (Banbury et al., 2021b) as it can substantially reduce a model's memory consumption. We observed, however, that the grayscale optimization disproportionately impacts images on the brighter end of the spectrum as illustrated in Figure 13. This further demonstrates the importance of fine-grained analysis, as some real-world deployment environments might be far brighter than the average Wake Vision test sample.

Appendix E. Open Image Label Distribution

Wake Vision is derived from Open Images V7, therefore the diversity of subjects in the images should follow the distribution of labels in Open Images. The Label distribution of Open images can be found here: `https://storage.googleapis.com/openimages/web/factsfigures_v7.html#statistics`

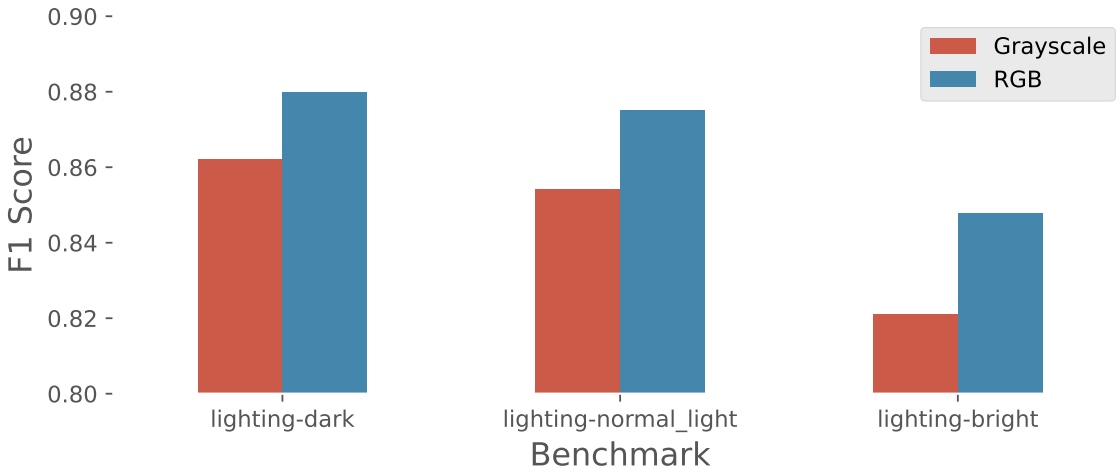


Figure 13: Impact of grayscale input images on the lighting benchmarks: dark, normal light, and bright. Models that use grayscale input images are more sensitive to bright lighting conditions than RGB.

Appendix F. Dataset Access and Organization

Wake Vision is available online at the Harvard Dataverse, via TFDS and Hugging Face Datasets. The version on Harvard Dataverse has been assigned the following DOI: `https://doi.org/10.7910/DVN/1HOPXC`. More information can be found on the webpage `https://wakevision.ai`.

The dataset is organized as a set of compressed tar files containing images and a series of label CSVs. The label CSVs are organized such that the file name of the image is the identifying index. CSVs for all dataset splits include the person and depiction label. The Validation and Test label CSVs also have flags that denote a sample's inclusion into a fine-grain benchmark set (e.g. Distance-Near). This structure makes it easy to access just the required data without requiring a full download. It also ensures the dataset can be easily updated as new versions are introduced.

Appendix G. Person Label Generation Details

Person Labels. The most straightforward Open Images label classes to label as a person in Wake Vision are the "person" label and its subcategories (Configurable, but defaults listed in Appendix I). All of these are relabelled as persons in Wake Vision. These labels are present as both image-level labels and bounding box labels.

We furthermore inspect the image-level label classes for synonyms and umbrella terms for all the person related labels. This search resulted in an additional six person related label classes. These person related label classes will only be used in the image level label configuration.

Lastly, the image level person labels in Open Images have an associated confidence score. This score ranges from 0-10. Labels that have been verified by humans to be

absent from an image have a score of 0, while labels that have been verified present by humans have a score of 10. Purely machine-generated labels have a fractional confidence score that is generally >= 5 (Kuznetsova et al., 2020; Krasin et al., 2017). By default, we only respect labels that have a minimum confidence of 7. Labels below this threshold are ignored.

Person Body Part Labels. Body parts are more challenging to relabel, as it is dependent on the use case whether a body part should be considered a person.

For example a camera that detects whether a person is inside a room to decide if the light should be switched on would want to consider body parts as a person, as this will keep the lights on even when the person is only partly in the camera frame. For waking up electronics, however, it may not make sense to consider body parts as persons. This could, e.g., mean that a computer would turn on when detecting a foot.

To cater to both use cases we include a flag in our open-source dataset creation code to set whether body parts should be considered persons. By default we consider body parts as persons.

Depictions. Open Images bounding box labels contain metadata about whether an object is a depiction, e.g., a painting or a photograph of a person. This presents the challenge of how to handle depictions. While most use cases would not consider a depiction a person, it could make sense to either exclude them to make training easier, or include them as non-persons to make a model resistant to seeing depictions when deployed.

In line with how we handle body parts, we therefore include a flag for our open-source dataset creation code to set whether depictions should be excluded or considered non-persons. By default depictions are considered non-persons.

Bounding Box Size. The VWW dataset only considered a Common Objects in Context (COCO) person to be a person if the bounding box around the person took up at least 0.5% of the image (Chowdhery et al., 2019). If the person took up less than 0.5% of the image, the image was excluded from the dataset. For Wake Vision, we observed a decrease in performance at the 0.5% limit. We theorize that the cause of this decrease is that a person becomes indistinguishable at such a low percentage of already small images. Therefore we adopted a 5% threshold for the Wake Vision dataset instead. For different requirements, users can change a configuration parameter in our open-source dataset creation code.

Appendix H. Open Images Download

The full Open Images v7 dataset is not hosted by the dataset authors. Rather it is provided as a collection of Flickr image Uniform Resource Locators (URLs) and their associated labels. As an unfortunate result of this, the dataset is not static over time as image owners can delete their images from the Flickr platform. As a result we were only able to download a subset of the original Open Images v7 dataset as shown in Table 9.

Table 9: Number of images downloaded from Open Images v7. Download occurred between the 28th of November to the 5th of December 2023

	Train	Validation	Test
Downloaded	7,936,979	36,406	109,305
Errors	1,055,669	5,214	16,131

Appendix I. Person Label Classes

In our default configuration, we consider the following Open Images v7 labels to be a person for the Wake Vision dataset:

- *Person*
- *Woman* (Subcategory of *Person*)
- *Man* (Subcategory of *Person*)
- *Girl* (Subcategory of *Person*)
- *Boy* (Subcategory of *Person*)
- *Human body* (Part of *Person*)
- *Human face* (Part of *Person*)
- *Human head* (Part of *Person*)
- *Human* (Person synonym - Only in Image Level Label Configuration)
- *Female person* (Woman synonym - Only in Image Level Label Configuration)
- *Male person* (Man synonym - Only in Image Level Label Configuration)
- *Child* (Umbrella term for *Girl* & *Boy* - Only in Image Level Label Configuration)
- *Adolescent* (Umbrella term for *Girl* & *Boy* - Only in Image Level Label Configuration)
- *Youth* (Umbrella term for *Girl* & *Boy* - Only in Image Level Label Configuration)

The following labels are considered body parts and are labeled according to the body part configuration:

- *Human eye* (Part of *Person*)
- *Skull* (Part of *Person*)
- *Human mouth* (Part of *Person*)
- *Human ear* (Part of *Person*)
- *Human nose* (Part of *Person*)
- *Human hair* (Part of *Person*)
- *Human hand* (Part of *Person*)
- *Human foot* (Part of *Person*)
- *Human arm* (Part of *Person*)
- *Human leg* (Part of *Person*)
- *Beard* (Part of *Person*)

Appendix J. Manual Label Correction

We use the Scale Rapid tool from Scale AI to relabel the Wake Vision validation and test sets. Figure 14 shows a screenshot of the labeling menu presented to the human labelers. The labelers were instructed to label an image as a "person" if a person was

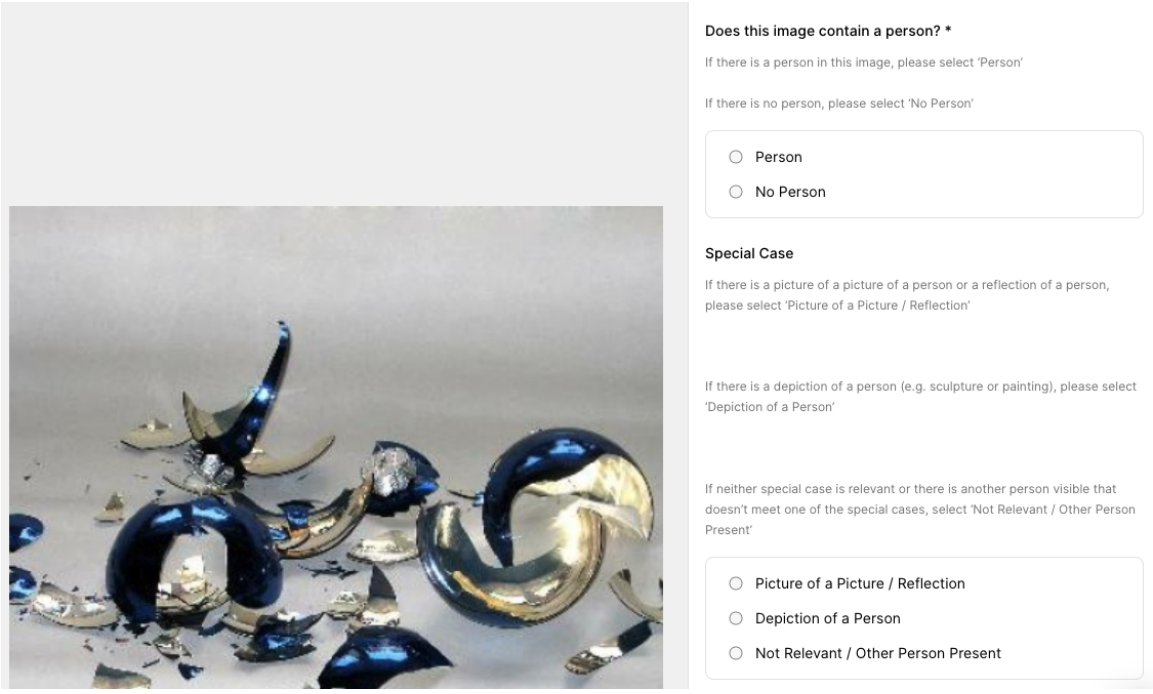


Figure 14: A screenshot of the labeling menu used to manually label the validation and test sets.

present anywhere in the image and "no person" if no visible person was present. The labelers also indicated if the image contained a depiction of a person or a "picture of a picture/reflection", which we use for metadata.

The cost per image was $0.10 USD and is set based on Scale's pricing structure. Each image is labeled by 3 different labelers to form a consensus. The total cost of labeling was $7089.8 USD. The authors do not know the hourly rate paid to each labeler, but Scale lists $18 USD/hr on job postings at the time of writing. For context on the difficulty of the task, the authors were able to average around 500 images per hour at a reasonable pace when estimating error rates.

Appendix K. Automatic Label Correction

To address the challenge of correcting Wake Vision validation and test labels we initially employed the Confident Learning technique (Northcutt et al., 2021a) to intelligently identify potential label errors. We selected Confident Learning as prior work has demonstrated its capability to find label errors in large datasets (Northcutt et al., 2021b). Confident Learning flagged suspected mislabeled instances, which we inspected and corrected through a manual verification process.

As shown in Table 10, the confident learning process identified a large amount of possible label errors in Wake Vision's Validation and Test sets. However, only between 12 and 16% of these possible label errors were legitimate errors. We corrected these identified label errors in the Wake Vision validation and test sets.

Table 10: Amount of label errors identified and corrected using Confident Learning.

Dataset Split	Total Size	Suggested Errors	Corrected Errors	Suggestion Accuracy
Validation	18,582	632	81	12.82%
Test	51,282	1672	267	15.97%

Given this low acceptance rate of label issues identified by Confident Learning, we concluded that we could not automate label cleaning to a point where no human in the loop is needed. This made this strategy too human-intensive to be applied to the much larger training sets.

To further correct label errors in the final Wake Vision Validation and Test sets we employed the Scale AI platform to crowd-source manual label corrections. These label corrections are described in Section 3.2 and deprecate the automatic label corrections described in this section.

Appendix L. Author Statement

The authors of this paper hereby confirm that we bear all responsibility in case of violation of rights, etc, and confirm the data license to be CC BY 4.0 for our labels, and that all images in the dataset have been uploaded to Flickr with a CC BY 2.0 license. Note: while the Open Images authors tried to identify images that are licensed under a Creative Commons Attribution license, we make no representations or warranties regarding the license status of each image and users should verify the license for each image themselves.

Appendix M. Hosting, License and Maintenance Plan

The dataset is hosted on Harvard Dataverse and HuggingFace Datasets. Additionally, the dataset is available through TensorFlow datasets and can be regenerated from Open Images using our open-source filtering code. The Edge AI Foundation will assume responsibility for hosting and maintaining the dataset long-term. In Appendix P we describe our procedure for image take-downs if individuals wish to remove their likeness from the dataset.

Wake Vision's labels and metadata are licensed under a CC BY 4.0 license. Images in the dataset have all been uploaded to Flickr with a CC BY 2.0 license. Note: while the Open Images authors tried to identify images that are licensed under a Creative Commons Attribution license, we make no representations or warranties regarding the license status of each image and users should verify the license for each image themselves.

Appendix N. DOI and Croissant URLs

The version of Wake Vision on Harvard Dataverse has been assigned the following DOI: `https://doi.org/10.7910/DVN/1HOPXC`. The following two croissant files describe the Wake Vision (Quality, Validation, and Test) and Wake Vision (Large) respectively: `https://huggingface.co/api/datasets/Harvard-Edge/Wake-Vision/croissant`,

https://huggingface.co/api/datasets/Harvard-Edge/Wake-Vision-Train-Large/croissant

Appendix O. Wake Vision Challenge Results

Submissions to the model-centric track have been trained using the same recipe adopted in Section 4.3. Table 11 provides input shape, Flash, RAM, and MACs for each model presented in Section 4.8 as well as the mean and standard deviation of test accuracies for the Wake Vision dataset. RAM and Flash have been measured using the "stm32tflm" utility of X-CUBE-AI 8.1.0, whereas MACs with "stm32ai".

Table 11: Raw data of model-centric track.

Model .tflite	**input volume**	**RAM [kiB]**	**Flash [kiB]**	**MACs [MM]**	**Test Acc [%]**
anas-benalla	(50,50,3)	48.5	104.55	4,899,292	79.7 ± 0.28
samy	(80,80,3)	73.5	34.55	5,718,064	79.5 ± 0.61
benx13	(48,48,3)	63.5	94.95	1,693,474	79.6 ± 0.22
cezar	(50,50,3)	245.5	57.23	20,435,868	79.6 ± 0.42
mohammad_hallaq	(80,80,3)	129.5	128.32	3,887,331	77.3 ± 0.5
ymac	(50,50,3)	34.5	27.77	431,985	76.7 ± 0.51
apighetti	(50,50,3)	48.5	278.36	693,818	74.3 ± 0.31

Table 12 shows test accuracy, precision, and recall of a MobileNetV2 having a width modifier of 0.25, trained on the original Wake Vision large split and on the same split enhanced with the technique proposed by the winner of the data-centric track (Bedioui, 2025). The training recipe is the same reported in Section 4.3.

Appendix P. Removal of Images

Removal of images can be asked via email at wakevision@edgeaifoundation.org.

Appendix Q. Datasheet for Wake Vision

This document is based on *Datasheets for Datasets* by Gebru *et al.* (Gebru et al., 2021). Please see the most updated version here.

MOTIVATION

For what purpose was the dataset created? Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.

Table 12: Main improvement of the data-centric track.

Metrics	Train Large Split	
	Original	**Enhanced**
MobileNetV2_0.25		
Test Accuracy	82.3 ± 0.2	87.1 ± 0.81
Precision	86.4 ± 1.49	87.6 ± 1.78
Recall	76.7 ± 1.83	86.5 ± 1.32

Wake Vision was created as a prototype for a new generation of datasets for TinyML. Where previous TinyML datasets were small due to limited resources, Wake Vision is a production-grade dataset consisting of almost 6 million images and with an extensively cleaned validation and test set. Wake Vision specifically targets the binary person classification task. Such a large dataset enables TinyML researchers to explore questions previously impossible. For example, the size of the Wake Vision dataset enables new research into the tradeoff that often exists between dataset size and sample quality.

Who created this dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?

The authors of this paper created the dataset.

What support was needed to make this dataset? (e.g. who funded the creation of the dataset? If there is an associated grant, provide the name of the grantor and the grant name and number, or if it was supported by a company or government agency, give those details.)

The Google TPU Research Cloud program partially supported this work via cloud computing credits. The PhD stipend of Emil Njor was supported by the Innovation Fund Denmark DIREC project (9142-00001B). Emil Njor's external research stay was furthermore supported by Fulbright Denmark, Stibo Fonden, Thomas. B. Thriges Fond, Otto Mønsteds Fond and Kaj og Hermilla Ostenfelds Fond. This work was also supported by the National Science Foundation (NSF) and the Semiconductor Research Corporation (SRC).

Any other comments?
No additional comments

COMPOSITION

What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)? Are there multiple types of instances (e.g., movies, users,

and ratings; people and interactions between them; nodes and edges)? Please provide a description.

The dataset is composed of images, labels, and metadata. The images are sourced from Open Images. The labels are binary, with '1' indicating there is a person somewhere in the image and '0' indicating that there is not a person in the image. The metadata is used to form the benchmark suite where we can evaluate models in specific challenging settings. Each image has a binary variable for each fine-grained benchmark where a '1' indicates that the image belongs to the fine-grained benchmark set.

How many instances are there in total (of each type, if appropriate)?

There are 5,760,428 labeled images in the dataset.

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).

The dataset uses a large subset of the Open Images dataset (6 million out of a possible 9 million images). We use a subset in order to have a balanced number of person samples and non-person samples. We also filter the dataset for quality. The filtering removes subjects that are too far and outside of a center crop. Open Images sources images from Flickr that are licensed as CC-BY 2.0. This is likely not a fully representative sample of the overall world population and likely skews heavily towards a European or North American population. See the Open Images paper (Kuznetsova et al., 2020) for more information.

What data does each instance consist of? "Raw" data (e.g., unprocessed text or images) or features? In either case, please provide a description.

The dataset is composed of images, labels, and metadata.

Is there a label or target associated with each instance? If so, please provide a description.

Yes, every image has a binary label to indicate if a person is visible in the image. Images in some dataset splits include more metadata labels, e.g., about the distance of the subject or whether the subject is a depiction.

Is any information missing from individual instances? If so, please provide a description, explaining why this information is missing (e.g., because it was unavailable). This does not include intentionally removed information, but might include, e.g., redacted

text.

The training set does not contain the metadata used for the fine-grained benchmark suite. The dataset contains two training sets, one of higher quality and one larger. The higher quality dataset and other sets contain information about depictions and body parts. The large training set does not contain this information.

Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)? If so, please describe how these relationships are made explicit.

No

Are there recommended data splits (e.g., training, development/validation, testing)? If so, please provide a description of these splits, explaining the rationale behind them.

Yes there are separate splits for training, validation, and testing. These splits are generated from equivalent splits in Open Images. The dataset contains two training splits, one focusing on quality of samples and one on number of samples.

Are there any errors, sources of noise, or redundancies in the dataset? If so, please provide a description.

Yes there are label errors in the dataset due to the labeling process. We discuss this at length in the paper.

Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? If it links to or relies on external resources, a) are there guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created); c) are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a future user? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.

The dataset is self-contained. We re-host the images, labels, and metadata. The images may be subject to different licenses than our labels and metadata. All images have originally been hosted on Flickr under a permissive CC BY license.

Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals' non-public communications)? If so, please provide a description.

No, the images were posted online with the CC-BY 2.0 license. It is possible an image was posted without the individual's consent. While we tried to identify images that are licensed under a Creative Commons Attribution license, we make no representations or warranties regarding the license status of each image, and a user should verify the license for each image themselves.

Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? If so, please describe why.

No, not to our knowledge. Due to the size of the dataset we have not had the resources to manually verify this.

Does the dataset relate to people? If not, you may skip the remaining questions in this section.

Yes, the dataset contains images of people.

Does the dataset identify any subpopulations (e.g., by age, gender)? If so, please describe how these subpopulations are identified and provide a description of their respective distributions within the dataset.

Yes the benchmark suite contains subsets that focus on perceived age and gender. These labels have been sourced from the More Inclusive Annotations of People (Schumann et al., 2021). The purpose of these benchmarks is to test the fairness of a trained model across these subpopulation and we do not condone using these labels to train a model. The size of these sets is listed in Table 6.

Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset? If so, please describe how.

It could be possible to identify a person by their face or other visible characteristics in the image. We do not share any other personal information besides the images.

Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)? If so, please provide a description.

No, not to our knowledge. Some personal or sensitive information may be visible in the images. However, the images were shared publicly under a CC-BY 2.0 license. We do not include any personal information beyond the images.

Any other comments?

The images used were published publicly under a CC-BY 2.0 license and are already used in the Open Images (Krasin et al., 2017) dataset and its derivatives. The authors of Open images tried to identify images that are licensed under a Creative Commons Attribution license but make no representations or warranties regarding the license status of each image and a user should verify the license for each image themselves.

COLLECTION

How was the data associated with each instance acquired? Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? If so, please describe how.

The dataset is derived from Open Images (Krasin et al., 2017), which in turn acquired its images from Flickr. Section 3.1 describes the process.

Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)? If not, please describe the timeframe in which the data associated with the instances was created. Finally, list when the dataset was first published.

The original data collection was conducted in November of 2015. The Open Images paper (Kuznetsova et al., 2020) details the image acquisition process.

What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)? How were these mechanisms or procedures validated?

The images were sources from Flickr in an automated pipeline. The original labeling pipeline was a combination of automated labeling and manual verification and labeling. The Open Images paper details this process (Kuznetsova et al., 2020). We detail our process for relabeling and filtering the dataset in Section 3.1.

What was the resource cost of collecting the data? (e.g. what were the required computational resources, and the associated financial costs, and energy consumption - estimate the carbon footprint. See Strubell *et al.* (Strubell et al., 2019) for approaches in this area.)

We used cloud TPU credits provided by the Google Cloud TRC program. The primary source of required resources was model training to evaluate the dataset and the storage and bandwidth required to process and upload the dataset to hosting locations. The total size of

the dataset is approximately 2 TB. To the authors' knowledge, there is no public information about the power draw of TPUs. This matches the conclusions drawn by Strubell *et al.* (Strubell et al., 2019). This makes it impossible to disclose the energy consumption and estimated carbon footprint of the project.

We used paid crowdsource labelers to relabel the validation and test sets. The total cost of labeling was approximately 7,000 USD.

If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?

The dataset is a large subset of the total Open Images dataset. The subset is generated by a deterministic filtering process described in Section 3.1.

Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?

We used paid crowdsource labelers to relabel the validation and test sets. The total cost of labeling was approximately 7,000 USD. We used the Scale Rapid labeling service. A total of 70,898 images were labeled at a cost of 0.1 USD per image and 0.033 USD per individual label. Appendix 3.2 gives additional detail.

Were any ethical review processes conducted (e.g., by an institutional review board)? If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation.

No

Does the dataset relate to people? If not, you may skip the remainder of the questions in this section.

Yes the dataset contains images of people.

Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?

We obtained the images via the Open Images dataset (Kuznetsova et al., 2020) which originally sourced its images from Flickr (Flickr, 2024).

Were the individuals in question notified about the data collection? If so, please describe (or show with screenshots or other information) how notice was provided, and provide a link or other access point to, or otherwise reproduce, the exact language of the notification itself.

No the individuals were not notified to our knowledge. The images were posted publicly under the CC-BY 2.0 license.

Did the individuals in question consent to the collection and use of their data? If so, please describe (or show with screenshots or other information) how consent was requested and provided, and provide a link or other access point to, or otherwise reproduce, the exact language to which the individuals consented.

Yes, the images were posted publicly under the CC-BY 2.0 license. It is possible that the images were uploaded by a third party without the consent of the individual in the image.

If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses? If so, please provide a description, as well as a link or other access point to the mechanism (if appropriate)

Appendix P describes how we can be contacted for removal of individual images. As with Open Images, we make no representations or warranties regarding the license status of each image, and a user should verify the license for each image themselves via the Flickr link in the metadata. Note that it is impossible to revoke a CC BY license after publication (Library, 2024).

Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis)been conducted? If so, please provide a description of this analysis, including the outcomes, as well as a link or other access point to any supporting documentation.

No formal analysis has been conducted. We describe the ethical considerations of the dataset in Section 5.

Any other comments?

No additional comments.

PREPROCESSING / CLEANING / LABELING

Was any preprocessing/cleaning/labeling of the data done(e.g.,discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? If so, please provide a description. If not, you may skip the remainder of the questions in this section.

Yes we filtered the data as shown in Figure 3.

Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? If so, please provide a link or other access point to the "raw" data.

The "raw" data is accessible in the Open Images dataset.

Is the software used to preprocess/clean/label the instances available? If so, please provide a link or other access point.

Yes, we open-source our filtering code. `https://github.com/harvard-edge/Wake_Vision`

Any other comments?

No additional comments

USES

Has the dataset been used for any tasks already? If so, please provide a description.

The dataset has been used for the experiments reported in the paper.

Is there a repository that links to any or all papers or systems that use the dataset? If so, please provide a link or other access point.

The usage of the dataset can be tracked via the citations of this paper. We also host competitions using the dataset and track participants.

What (other) tasks could the dataset be used for?

The dataset can be used as a benchmark for TinyML models and as the basis for data-centric AI research given the prevalence of label errors in the training set and the cleaned nature of the validation and test sets.

Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? For example, is there anything that a future user might need to know to avoid uses that could result in unfair treatment of individuals or groups (e.g., stereotyping, quality of service issues) or other undesirable harms (e.g., financial harms, legal risks) If so, please provide a description. Is there anything a future user could do to mitigate these undesirable harms?

Yes, the estimated error rates are reported in Table 3. We also discuss the high likelihood of false negatives in Wake Vision's training sets in Section 3.2. These errors

are due to the labeling and filtering process. We mitigate the impact of these errors by manually labeling the validation and test sets as well as study their impact for TinyML models in Section 4.7.

Are there tasks for which the dataset should not be used? If so, please provide a description.

The perceived age and gender benchmarks should not be used to train a gender or age classifier.

Any other comments?

No additional comments

DISTRIBUTION

Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created? If so, please provide a description.

The dataset is hosted on Harvard Dataverse and HuggingFace Datasets. Additionally, the dataset is available through TensorFlow datasets and can be regenerated from Open Images with our open source filtering code.

How will the dataset will be distributed (e.g., tarball on website, API, GitHub)? Does the dataset have a digital object identifier (DOI)?

Yes, the dataset is hosted on Harvard Dataverse and HuggingFace Datasets. Additionally, the dataset is available through TensorFlow datasets and can be regenerated from Open Images with our open source filtering code.

When will the dataset be distributed?

The Dataset is publicly accessible at the time of writing.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? If so, please describe this license and/or ToU, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms or ToU, as well as any fees associated with these restrictions.

The dataset labels are published under the CC-BY 4.0 license. All images in the dataset have been uploaded to Flickr under a CC BY 2.0 license. Note: while the Open Images

authors tried to identify images that are licensed under a Creative Commons Attribution license, we make no representations or warranties regarding the license status of each image and users should verify the license for each image themselves.

Have any third parties imposed IP-based or other restrictions on the data associated with the instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms, as well as any fees associated with these restrictions.

No

Do any export controls or other regulatory restrictions apply to the dataset or to individual instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any supporting documentation.

No

Any other comments?

No additional comments.

MAINTENANCE

Who is supporting/hosting/maintaining the dataset?

The dataset is hosted on HuggingFace Datasets. Additionally, the dataset is available through TensorFlow datasets and can be regenerated from Open Images with our open-source filtering code. The Edge AI Foundation will assume responsibility for hosting and maintaining the dataset long-term.

How can the owner/curator/manager of the dataset be contacted (e.g., email address)?

At the time of writing, the authors can be contacted at cbanbury@g.harvard.edu. Once maintenance is transferred to the Edge AI Foundation, there will be a point of contact there.

Is there an erratum? If so, please provide a link or other access point.

No

Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)? If so, please describe how often, by whom, and how updates will be communicated to users (e.g., mailing list, GitHub)?

The dataset may be updated if any problematic instances are discovered or the label errors are corrected. In this case, we will issue a new version of the dataset to ensure past results are contextualized under the version of the dataset that was used.

If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)? If so, please describe these limits and explain how they will be enforced.

There is no limit.

Will older versions of the dataset continue to be supported/hosted/maintained? If so, please describe how. If not, please describe how its obsolescence will be communicated to users.

Yes, we will continue to host older versions of the dataset. This is an inherent feature of the Harvard Dataverse.

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? If so, please provide a description. Will these contributions be validated/verified? If so, please describe how. If not, why not? Is there a process for communicating/distributing these contributions to other users? If so, please provide a description.

Yes, the dataset is licensed permissively, and the code used to create Wake Vision is open source.

Any other comments?

No additional comments