Training AI to Recognize Objects of Interest to the Blind and Low Vision Community

Tharangini Sankarnarayanan[1], Lev Paciorkowski[1], Khevna Parikh[1], Giles Hamilton-Fletcher[2], Chen Feng[3], Diwei Sheng[3], Todd E. Hudson[4], John-Ross Rizzo[4], Kevin C. Chan[5*]

Abstract— **Recent object detection models show promising advances in their architecture and performance, expanding potential applications for the benefit of persons with blindness or low vision (pBLV). However, object detection models are usually trained on generic data rather than datasets that focus on the needs of pBLV. Hence, for applications that locate objects of interest to pBLV, object detection models need to be trained specifically for this purpose. Informed by prior interviews, questionnaires, and Microsoft's ORBIT research, we identified thirty-five objects pertinent to pBLV. We employed this user-centric feedback to gather images of such objects from the publicly available Google Open Images V6 dataset. We subsequently trained a YOLOv5x model with this dataset to recognize these objects of interest. We demonstrate that the model can identify objects that previous generic models could not, such as those related to tasks of daily functioning – e.g., coffee mug, knife, fork, and glass. Crucially, we show that the careful pruning of a dataset with severe class imbalances leads to a rapid, noticeable improvement in the overall performance of the model by two-fold, as measured using the mean average precision at the intersection over union thresholds from 0.5 to 0.95 (mAP50-95). Specifically, mAP50-95 improved from 0.14 to 0.36 on the seven least prevalent classes in the training dataset. Overall, we show that careful curation of training data can improve training speed and object detection outcomes. We show clear directions on effectively customizing training data to create models that focus on the desires and needs of pBLV.**

I. Introduction

Object detection refers to modern computer vision techniques that both locate and assign object labels to all detected objects within a given image. Object detection algorithms typically show the results of these functions by drawing bounding boxes around recognized objects with their corresponding class labels and confidence ratings. The use of object detection models has been demonstrated across a variety of domains, including self-driving cars [1], medical imaging [2], and visual assistive technologies [3].

With the continually improving speed and performance of object detection algorithms via deep learning techniques, recent models have become suitable for mobile applications that require real-time operations. In particular, the YOLO (You Only Look Once) architecture [4], first introduced in 2015, uses a regional-convolutional neural network (R-CNN) to efficiently identify image regions that are likely to contain objects, followed by identifying and locating all objects within the image regions. YOLO stands out for its fast inference time and low computational costs, making it suitable for efficient object detection on smartphone devices, further increasing the accessibility of this information in real time (Figure 1) [5].

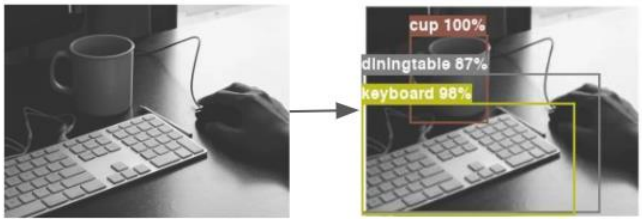

Figure 1. Object Detection using YOLO. Input image (left) is processed by YOLO (right). This provides the location and size of identified objects using bounding boxes as well as identity and confidence information via text/color.

Although YOLO can quickly identify some objects in images, its performance deteriorates when trained on unnecessary object categories, localizing smaller objects, or detecting objects for which there is less training data available [6]. Of note, training object detection models to recognize new objects for specific user scenarios can be difficult if available training data is sparse or imbalanced. Training YOLO to recognize objects involves providing sets of RGB images with corresponding bounding box data, which provides the location and size of objects within the image. Object detection models such as YOLO can provide beneficial information to persons with blindness or low vision (pBLV). However, when used to facilitate tasks of daily living, default models are trained on a subset of object categories that may be less relevant to pBLV than other objects for which training data exists.

To address this issue, we train a YOLOv5x system to recognize key objects from the perspective of pBLV. This work focuses on the specific selection of object training data, with an aim to demonstrably increase the speed of training, the quality of the object detection model, and the relevance of the model to benefit pBLV. To accomplish this, we selected new training datasets relevant to pBLV. Then we used a rule-based algorithm to selectively filter the training data to correct class

This work is supported in part by the U.S. Department of Defense Vision Research Program W81XWH2110615 (Arlington, Virginia); and an unrestricted grant from Research to Prevent Blindness to NYU Langone Health Department of Ophthalmology (New York, New York).

[1]Tharangini Sankarnarayanan, Lev Paciorkowski, and Khevna Parikh are with the Center for Data Science, New York University, New York, NY, USA 10017;

[2]Giles Hamilton-Fletcher is with the Department of Ophthalmology, NYU Grossman School of Medicine, NYU Langone Health, New York University, New York, NY, USA 10017;

[3]Dean Sheng and Chen Feng are with the Department of Civil and Urban Engineering and Department of Mechanical and Aerospace Engineering, Tandon School of Engineering, New York University, New York, NY, USA 11201;

[4]Todd E. Hudson and John-Ross Rizzo are with the Department of Rehabilitative Medicine, NYU Grossman School of Medicine, NYU Langone Health, and Department of Biomedical Engineering, Tandon School of Engineering, New York University, New York, NY USA 10017;

[5]Kevin C. Chan is with the Departments of Ophthalmology and Radiology, Neuroscience Institute, NYU Grossman School of Medicine, NYU Langone Health, and the Department of Biomedical Engineering, Tandon School of Engineering, New York University, New York, NY, USA 10017 (*corresponding author to provide e-mail: chuenwing.chan@fulbrightmail.org).

Manuscript 899 submitted to 45th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Received February 7, 2023.

imbalances. We show that this rebalancing improves model validation outcomes in both speed of training and accuracy of object detection for thirty-five objects of interest to pBLV.

II. Background

A. Assistive Technology, R-CNNs and YOLO

Early models utilizing object detection for the assistance of pBLV focused on use cases in targeted environments. This includes objects such as door detection in unfamiliar buildings [7] or identifying medicine in cabinets [8]. Another notable prototype system makes real-time object detection from live camera feeds accessible for pBLV by translating this information into acoustic signals for the user [9]. These projects indicate positive outcomes for models focused on specific domains and environments.

Assistive technologies requiring real-time feedback on live image processing have largely focused on R-CNN approaches. These approaches break down an image into ~2000 'region proposals,' so that instead of the whole image, the model only looks at a portion to classify whether an object is present within a particular region. However, due to their selective search algorithms to identify objects within the proposed regions, R-CNNs require extensive training time. In addition, increasing the total number of object categories that the model needs to select between degrades the inference time needed to categorize, slowing the whole model down.

For R-CNN models to be well suited for specific assistive technology use case scenarios, they should adopt several factors. Firstly, the specific object categories trained for detection should closely match the end users' needs. This helps streamline the model to improve inference time but can also avoid redundant information and reduce chances for misrecognitions. Secondly, dataset curations that reduce redundancies of either object categories or training data volume could improve model training time. Faster model training allows for more viable assistive technologies because new models can be created to meet the needs of specific users, populations, or scenarios. Overall, resolving these issues and optimizing the training process can therefore have a large influence on whether biomedical devices and Apps can be beneficial to the blind and low-vision community.

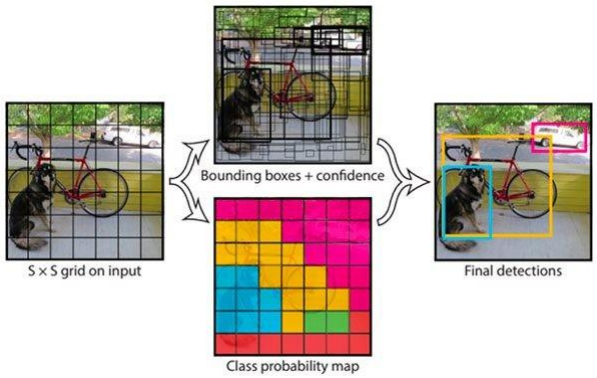

Figure 2. YOLO Detection [4]. Leftmost image shows an input image which has been divided into a grid of cells. This is used to create numerous object detection predictions of varying confidence ratings (top middle image) and a class probability map with the most likely classes (bottom middle image). This is used to produce final bounding box detections (right image).

One of the most popular modern R-CNN approaches is the YOLO series of models (Figure 1). This divides a whole image into grid cells. Each grid cell then aims to predict the class of an object present in that specific grid cell as well as the coordinates of the bounding box that specify the object's size and location. In terms of outputs, YOLO returns a vector including the bounding box width and height, the coordinates of the center of the bounding box, the probability that there is an object present in the bounding box, and the probability of objects associated with each class. In the case of two objects in the same grid cell, the grid cell is responsible for only detecting the object whose center is in that grid cell. This process is used to deliver YOLO's final bounding box results.

B. Evaluating Object Detection Models

To assess the algorithm's object detection performance, the bounding box predictions by the model are compared against ground-truth bounding box values (e.g., human annotations). This involves Intersection over Union (IoU – Figure 3) which evaluates the overlap between the predicted and ground-truth bounding boxes. An IoU value of 1 indicates perfect overlap, and 0 indicates no overlap. A threshold (α) of 0.5 is typically used to identify whether there is a 50% overlap between the two bounding boxes. If the IoU for the model's prediction is $\geq$0.5, then the object detection is classified as a True Positive (TP). Conversely, if the IoU is <0.5, it would be an incorrect detection, classifying it as a False Positive (FP). If the model fails to predict the object in an image, it would be classified as a False Negative (FN). These metrics are used to calculate the precision and recall for each object detected, which indicate the exactness and completeness of the model, respectively. Precision is determined by what proportion of positive identifications is correct [TP / (TP + FP)], whereas recall reflects what proportion of actual positives is identified correctly [TP / (TP + FN)]. In combination, these metrics create a Precision/Recall Curve for each object class, where the area under the curve (AUC) for each object class indicates the per-class average precision (AP). The average of all object classes is the mean average precision (mAP) and is typically evaluated at an IoU threshold of 50% (mAP50) or increasing increments of 5, from 50% to 95% IoU (mAP50-95).

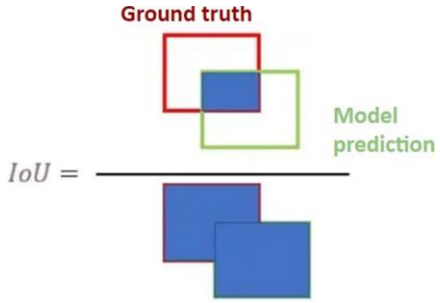

Figure 3. Intersection Over Union (IoU). To evaluate how well a given object detection model predicts objects in test images, IoU is calculated to provide the level of overlap between bounding boxes generated by the model and the ground-truth values. This is calculated by the proportion overlap between model predictions and ground-truth values, divided by the total area.

III. Method

Our goal is to train AI to detect thirty-five selected objects of interest to pBLV with a model that is computationally efficient enough to allow real-time inference on smartphones. We utilize the YOLO architecture as it can run on smartphones and provides object classification and localization. To select thirty-five objects of interest to pBLV, 15 classes were first

identified from the ORBIT dataset [10]. The ORBIT dataset tasked persons with blindness or low vision to select objects of interest to them, that they would like future object recognition software to detect. We identified objects from this selection that were both trainable from exemplars and for which suitable training data exists. The remainder of the objects were selected from prior interviews with pBLV [7], questionnaires of daily functioning for pBLV [11], indoor objects that are frequently moved around, and finally objects related to safety or navigation, such as doors and door handles [12].

Google Open Images [4] is one of the few publicly available annotated image datasets, with 16 million bounding boxes for 600 object classes. We utilize Google Open Images V6 to obtain images of the classes under consideration. Across the thirty-five objects of interest to pBLV, these objects appear at widely varying frequencies in Google Open Images V6. The most populous classes are thousands of instances of objects such as 'person' or 'car,'. Other key objects have much lower frequencies, with only a few hundred instances of objects such as 'remote control' or 'door handle.' This severe class imbalance hinders the ability to train a model across all objects using regular training and validation data-splitting methods.

In the present study, we used YOLOv5x [13] because of its availability on smartphone platforms while this model outperforms prior versions in terms of accuracy and ability to analyze multiple objects in the same grid. After training several epochs on the whole training dataset, we noticed poor performance for object classes with less prevalence in the training data. To address this, we implemented a rule-based algorithm to prune the dataset to improve training outcomes. The goal is to have a training dataset with a more balanced class distribution. We accomplished this by carefully removing images containing the most common classes, effectively amplifying the signal of the images containing the least common classes. This procedure has the added benefit of reducing epoch time by making the training dataset smaller. Our approach for pruning the training dataset is as follows:

1) Denote the whole dataset as D. Define a threshold x as the approximate maximum number of *instances* (there may be more than one instance of an object per image) desired for any class in the dataset. Classify object classes into two categories: "common" classes c with more than x instances in the dataset; and "rare" classes r with fewer than x instances in the dataset.
2) Randomly select an image from D, and apply the following rules:
   - If the image contains any examples of r classes, automatically keep and place in pruned dataset P.
   - Otherwise, if adding the image to P would cause any one class to exceed x total instances within P, discard the image.
   - Otherwise, add the image to P.
3) Continue until all images in D have been processed.

The images in D are processed in a random sequence to limit any potential bias resulting from the filtered dataset. As a result of using our pruning algorithm, we transitioned from the original training dataset of a 90% random split of the entire dataset to a more balanced training dataset (Figure 4).

To evaluate the model's object detection capability, we used mean average precision (mAP). For this we calculated the intersection over union (IoU) for bounding boxes generated by the model's prediction and ground-truth values. We predefined an IoU α threshold of 0.5 in classifying whether the prediction is a true positive or a false positive. Average precision (AP) describes the per-class Area Under the Precision-Recall Curve (AUC-PR). The average of AP values across all classes is the mAP. The model was trained until convergence, defined as 30 consecutive epochs without improvement in validation mAP.

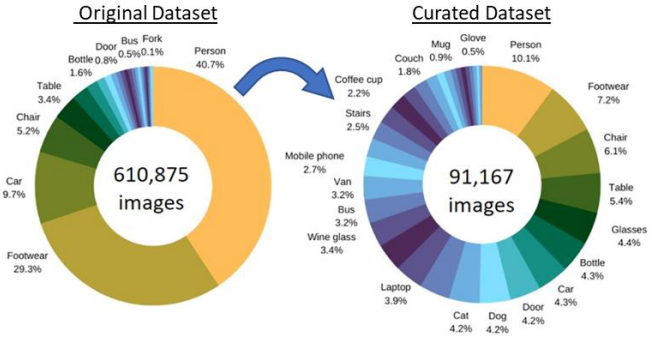

Figure 4. Original and Curated Datasets for YOLO model training. Left image shows the object class division of the original dataset from Google Open Images V6. This dataset is curated using our pruning algorithm (using x = 10,000) to produce the curated dataset (right image) with a more balanced object class distribution and fewer images overall.

IV. RESULTS

To compare the impact of training on the full original dataset against training on the smaller curated dataset, we evaluated the model's efficiency and accuracy across all object categories as it was trained. Originally, the YOLOv5x model was trained using the full training dataset, taking approximately 6 hours per epoch (i.e., model update). The model was then trained on the smaller curated training dataset, which drastically reduced the time needed to update the model to ~45 mins per epoch. The YOLOv5x model was continually trained until convergence at 107 epochs. While the initial training on the unpruned dataset yielded an average validation mAP50-95 of 0.199, switching to the pruned dataset ended up yielding an average validation mAP50-95 of 0.413, improving performance more than two-fold.

Table 1: Final model performance on the test dataset after training on both the original and curated training datasets (107 epochs).

Class	Prevalence (by instance)	Precision	Recall	mAP 50	mAP 50-95
Cat	0.61%	0.863	0.884	0.913	0.793
Dog	1.34%	0.829	0.884	0.899	0.787
Mobile phone	0.26%	0.699	0.886	0.843	0.77
Coffee cup	0.23%	0.749	0.738	0.811	0.742
Bus	0.46%	0.581	0.751	0.752	0.661
...	...	...	...	...	...
Bottle	1.55%	0.217	0.389	0.232	0.189
Couch	0.16%	0.346	0.483	0.247	0.189
Glove	0.06%	0.125	0.351	0.166	0.125
Chair	4.98%	0.29	0.233	0.156	0.0969
Person	40.51%	0.288	0.043	0.119	0.0603
Overall Average	**100%**	**0.488**	**0.533**	**0.496**	**0.400**

As shown in Figure 5, switching to the pruned dataset produced the most noticeable improvements in the model's performance on the least prevalent classes. For example, when using the entire training dataset, the model could only achieve an average validation mAP50-95 of 0.024 on the seven least prevalent classes. However, in these same classes, the model

achieved an average validation mAP50-95 of 0.363 after switching to the pruned training dataset. Table I shows our final model's performance after 107 epochs on the held-out test dataset (which retains the original class imbalances). The model achieved an mAP50-95 of 0.400 with the highest performance on detecting 'cat' and 'dog' (0.793 and 0.787 mAP50-95, respectively). In addition, the model achieved mAP50-95 of 0.5 or greater on many smaller objects including 'forks,' 'knives,' 'handbags,' 'coffee cups,' and 'mobile phones' (Table 1). By contrast, performance on the seven most prevalent classes remained stable.

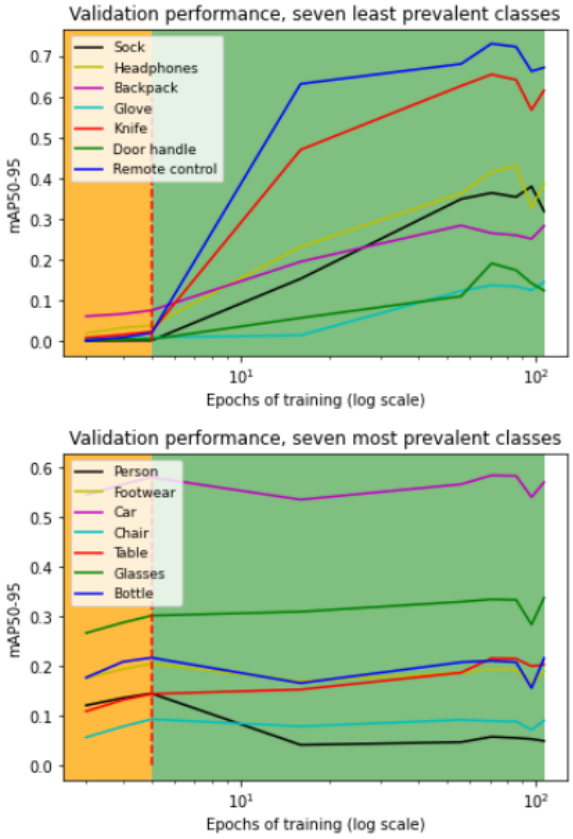

Figure 5. Validation performance for the seven least- (top image) and most- (bottom image) prevalent classes. The orange-shaded region depicts the usage of the complete training dataset. The green-shaded region depicts the subsequent usage of the pruned dataset for training. Here we see substantial increases in performance on mAP50-95 for the seven least prevalent classes after using the pruned dataset.

V. Discussion

While careful curation of the model's training data improved object detection for less prevalent objects, its ability to recognize when *no* objects were present remains unknown. This is important to consider for live use scenarios where excessive false positives may reduce its utility as an assistive technology. It was also noted that the model's performance metrics for common objects like 'person' might be lower than expected. This could be the result of test images that involve large groups of people, which create many more opportunities to miss 'person' objects than there are for many other object classes. By contrast, the model performed best with 'cat' and 'dog' objects, for which there are fewer images of animal groups. As such, some lower performances may be intrinsic for the typical multi-object images, rather than the object class per se. While this may indicate better outcomes for detecting single objects, the viability for this approach in cluttered environments remains uncertain [13]. In terms of practicality, the ability for the curated training set to enhance the detection of small, everyday objects which can be relocated frequently as well as guide dogs could be particularly helpful to pBLV.

VI. Conclusion

It is not uncommon for object detection training datasets to contain severe class imbalances. Here we demonstrate that careful pruning to balance the class distribution allows models such as YOLO to more effectively learn the classes which have the fewest exemplars. We trained a YOLOv5x model to detect 35 objects of interest to pBLV, and found that the time taken per epoch decreased while the detection performance increased. We also found large improvements in performance for object classes with the least training examples. These findings indicate that careful training dataset curation may be a crucial step in specializing assistive technologies for individual users' needs.

References

[1] A. Gupta, *et al*, "Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues." *Array* 10 pp. 100057, 2021.

[2] J.-G. Lee, *et al*, "Deep learning in medical imaging: general overview." *Korean Journal of Radiology* 18, no. 4, pp. 570-584, 2017.

[3] J. Redmon, *et* al, "You only look once: Unified, real-time object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788, 2016.

[4] A. Kuznetsova, *et al*, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale." *International Journal of Computer Vision* 128, no. 7 pp. 1956-1981, 2020.

[5] G. Hamilton-Fletcher, *et al*, "" I Always Wanted to See the Night Sky" Blind User Preferences for Sensory Substitution Devices." In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 2162-2174, 2016.

[6] T. Diwan, G. Anirudh, and J.V. Tembhurne. "Object detection using YOLO: Challenges, architectural successors, datasets and applications." *Multimedia Tools and Applications*, pp. 1-33, 2022.

[7] X. Yang, *et al*, "Context-based indoor object detection as an aid to blind persons accessing unfamiliar environments." In *Proceedings of the 18th ACM international conference on Multimedia*, pp. 1087-1090, 2010.

[8] N. Dalal, and B. Triggs. "Histograms of oriented gradients for human detection." In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, pp. 886-893, 2005.

[9] L. Dunai, *et al*, "Real-time assistance prototype—A new navigation aid for blind people." In *IECON 2010-36th Annual Conference on IEEE Industrial Electronics Society*, pp. 1173-1178, 2010.

[10] D. Massiceti, *et al*, "Orbit: A real-world few-shot dataset for teachable object recognition." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10818-10828, 2021.

[11] R.P. Finger, *et al*, "Developing an instrumental activities of daily living tool as part of the low vision assessment of daily activities protocol." *Investigative Ophthalmology & Visual Science*, 55(12), pp. 8458-8466, 2014.

[12] L. Niu *et al*, "A Wearable Assistive Technology for the Visually Impaired with Door Knob Detection and Real-Time Feedback for Hand-to-Handle Manipulation," In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 1500-1508, 2017.

[13] Z. Ge, *et al*, "YOLOX: Exceeding yolo series in 2021." *arXiv preprint arXiv:2107.08430*, 2021.