Assistance for the Visually Impaired: a system to identify product information and read curved labels when grocery shopping

Laura Arias Fernandez ARIAS048@UMN.EDU
Sarah Schmoller SCHMO065@UMN.EDU
Michael Lucke LUCKE096@UMN.EDU
Maria Gini GINI@UMN.EDU
Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455

Editor: William Hsu, Farrukh Ali

Abstract

Grocery store navigation can pose a number of unique challenges to visually impaired shoppers. In order to purchase a desired product, store patrons may need to use systems optimized for sighted customers when locating the appropriate section of the store, choosing an item out of many options displayed, or obtaining product and nutritional information from the item's packaging. While automated text-to-speech applications have become widely available in recent years, these solutions are often general-purpose and not tailored to specific needs a user might encounter in a grocery store setting. Our work aims to address this use-case through two tool components, comprising (1) an object recognition interface that is able to distinguish between common classes of grocery items and obtain product details through barcode scanning, and (2) a system capable of extracting text from a curved bottle label, in the case that a barcode is not available.

Keywords: Support for visually impaired, grocery shopping, curved label reader

1. Introduction

Automated visual aid applications are a useful source of information for those with limited vision, not only in interacting with text-based media, but also for navigating physical displays organized on a visual basis. Microsoft's SeeingAI, for instance, provides a well-developed set of tools including text-to-speech document reading, barcode scanning, and object recognition. Smartphone apps TapTapSee and Lookout by Google provide similar functionality.

The World Health Organization estimates that there are 285 million people who are visually impaired, and among those 39 million are blind. The study in (Bourne et al., 2017) predicts that the numbers will rise to 115 million by 2050, if treatment is not improved by better funding. The main reason for this growth is the growing size of the ageing population and the fact that more people are living into old age. The economic impact of blindness is very high because of the reduction of the working population, the cost of medical treatments for conditions such as cataracts, and the costs associated with supporting blind people in their every day activities. In our project, we focus on building computing tools to enable blind people to do simple activities, such as shopping and accessing information from labels on bottles.

We identify two primary areas for improvement on the established applications for the specific task of grocery shopping. First, the object detection systems for these applications, while highly precise when operating on individual items, tend to deliver broad generalizations when analyzing a larger scene. In our use case, an image of a grocery store shelf might be described as a shelf with food, without providing details on what types of products are depicted in the scene. Identification of narrower categories of items is important, as it may allow shoppers to more easily locate aisles with the types of products they are searching for. We implement this functionality by fine-tuning a Convolutional Neural Network (CNN) model on a labeled database of images of common household items. Once the proper portion of the store has been identified, the customer may scan the barcodes of specific products to retrieve more information, in a similar fashion to other existing aids.

Second, the solutions currently available do not account specifically for the reading of curved labels. This can lead to difficulties upon attempting to analyze cylindrical packages like pill bottles, as bar codes are not always present. We include with our solution a proof-of-concept feature designed to take a series of images from around the cylinder axis of a label's surface and stitch them together, providing a flattened composite image from which text may be read. Our system performs a layout analysis on this composite, using a mask region-based Convolutional Neural Network (R-CNN) model fine-tuned on annotated images of pill bottle labels, and finally reads the text in the resulting regions using Optical Character Recognition (OCR) software.

2. Background

Object Recognition Interface. As the fields of object recognition and automated image captioning have grown, a range of software packages have made seeing-eye functionality available to both developers and consumers. Nonetheless, as noted in Gurari et al. (2020), much of the existing publicly-available infrastructure is not well tailored to real potential users. For example, shortcomings remain common when applications are used to help people with visual impairments, despite major technological advancements. In regards to potential grocery shopping aids, previous work has attempted to facilitate object identification using feature matching. Masood (2015) describes the use of Scale Invariant Feature Transform (SIFT) and image matching techniques to successfully identify and price multiple grocery items by simply taking an image of a user's cart. However, in using this method, each product must be compared to an image in a database to identify its category, which may prove unfeasible in large grocery stores; Masood's implementation would require users to obtain a picture of every existing product alongside each distinct product design. Moreover, the technique described has been proven to work only on flat packaging. This also poses a limitation to its practicality, as many objects, such as fruits and cans, do not have flat surfaces.

A more robust approach would involve training a deep-learning model to distinguish between categories of relevant objects. A Convolutional Neural Network trained on a sufficiently large database of images would allow for the recognition of a large number of labelled classes, without needing to compile images of individual products. We see general-case object recognition of this kind in the original work on the You-Only-Look-Once (YOLO) one-stage detector algorithm, from a model trained on the PASCAL VOC 2007 dataset

(Redmon et al., 2015). This sort of generic functionality can be customized for our use-case by fine-tuning similar models, a strategy we employ in our solution. Here, we work with pretrained models based on later versions of YOLO, YOLOv4 and YOLOv5, to achieve easy identification of broad categories of groceries in an image (Bochkovskiy et al., 2020) (Jocher, 2020).

Curved Label Reader. Previous work has established several potential methods for capturing text from the curved surface of a bottle. Ye et al. (2013) describes a mosaic method of reconstructing a flattened image of the label from images captured around its cylinder axis. The flattened image can then be used in further processing, such as layout analyses and reading via OCR, with out concerns over loss of data due to the warping of the bottle. On the other hand, Sears et al. (2011). suggest that, in cases where images of the label are not intended for human consumption, there is no need to carry out image stitching prior to OCR. The authors capture images of the label in a similar way to (Ye et al., 2013), but apply OCR first, identifying the text visible in each image. They then apply a line stitch algorithm, which takes lines of text from two successive images and searches for the longest common substring (Sears et al., 2011). In this way, they concatenate the text appearing in the collection of images without stitching the images themselves.

The approach described in Sears et al. (2011) avoids the added complication of image stitching, and obtains an accurate representation of the label's text. However, it limits the work that can be done with layout analyses to ensure that all of the text being stitched belongs to the same column or region. If the text extracted from the image is to be used to effectively search for specific sections of the label, conducting an accurate layout analysis will be crucial. Product packaging can employ vastly different formats, colors and backgrounds depending on the item, and attempting to define a region of text as being bounded by "whitespace", as we would need to do in using Sears's method, is likely to fail in some cases. Creating a composite image prior to further processing, instead, allows us the option of using existing annotated datasets to aid in the layout analysis task, which is likely to be representative of a wider range of conditions than we might be able to otherwise represent. In later sections we describe our methods for creating this composite image and conducting our layout analysis for the purpose of the proof-of-concept application.

3. Object Recognition Interface

3.1. Overview

While a shopper might gather data about a specific product by scanning its barcode or reading its label via OCR, these kinds of features are unlikely to be helpful in identifying the broader region of a store in which a desired product is located. It is important, therefore, that our application be able to describe the kinds of objects in the immediate surroundings to a visually impaired user. Popular solutions in the market today often do include object recognition functionality, but this is rarely tailored to the specific task of grocery shopping. Seeing AI, for instance, can provide category information for an object scanned in isolation, but does not distinguish between classes of grocery items in a complex image of products displayed together on a shelf. In order to provide more robust object recognition specific to our desired use-case, we fine-tune a generic model based on the previously mentioned

YOLOv4 and trained on the MSCOCO dataset (Lin et al., 2014). The resulting model is capable of recognizing 21 distinct classes of items from within a complex image, and the addition of a voice user interface allows these to be read aloud to the user.

Upon identifying the desired region of the store, the shopper may begin the process of gathering information about individual products. In order to aid in this task, we have also developed a barcode scanning feature which identifies the location of a barcode in-frame, uses audio cues to direct the user to properly center it, and then scans it to obtain essential product information. A voice interface assists the user in navigating the the product details, obtained from an online database. Recognition of the barcode within a larger image was achieved by fine-tuning another object recognition model based on YOLOv5.

3.2. Implementation

Prior to fine-tuning the YOLOv4 model, 21 salient grocery classes were identified as desirable for detection. This list included the following:

Fruit	Pastry	Pasta	Cheese
Coffee	Bread	Salad	Cookie
Tea	Flower	Popcorn	Cooking spray
Milk	Fish	Toilet paper	Sushi
Ice cream	Juice	Sandwich	Candy
Vegetable			

Datasets for each of these classes were compiled from Google's Open Images (Kuznetsova et al., 2020), a publicly available repository of approximately 1.9 million pre-annotated images. Labels were programmatically converted to the COCO format, and fine-tuning was conducted using GPU resources provided by Google Colab. Checkpoints were saved on an hourly basis to allow for resumed training in the case of an unexpected halt. Due to the large size of the datasets and limitations in computational resources, only 80,000 iterations were achieved over the course of the fine-tuning process. Even in its current state, however, the model is able to accurately detect object classes in many cases, as demonstrated in Figure 1. While our model is functional for the purposes of this prototype, it is important to note that further training is needed to improve its performance. Inaccuracies similar to those displayed in Figure 2 occur at the time of inference for some classes.

Once a user has located the kinds of products they are looking for using the object recognition interface, the barcode scanner may be used to obtain more detailed information on specific items. In order to allow for the recognition of barcodes within a larger image frame, we applied another pretrained CNN object recognition model based on YOLOv5. We fine-tuned this model using an annotated dataset of 1,500 barcode images, obtained from open-source Machine Learning community platform Kaggle (Riaz, 2020). After completion of fine-tuning, the model was able to detect barcodes (class = 1) accurately with 100% recall (percentage of barcodes correctly identified) and a 98% precision (percentage of correct predictions), as displayed in Figure 4.

In the scanner's current form, the barcode must be positioned in close proximity to the camera and be well-centered in frame in order to be properly read. This can be difficult to achieve for visually impaired users. Detection of the barcode's position without scanning,

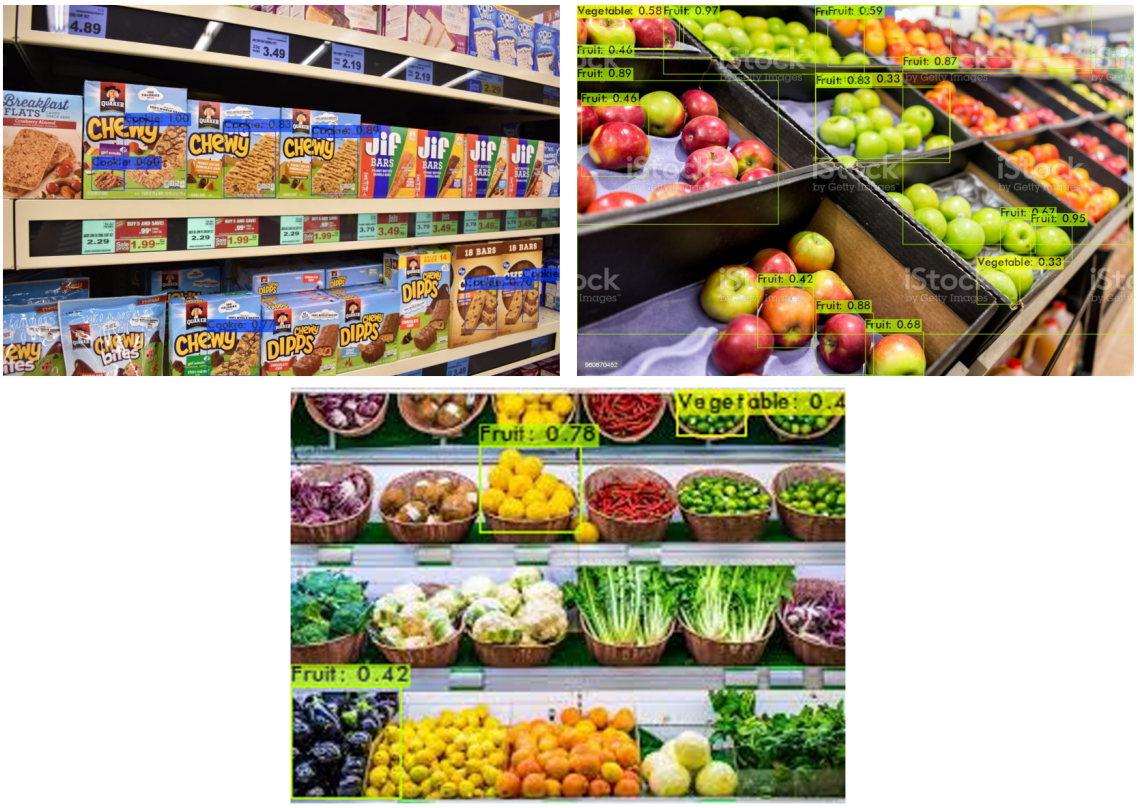

Figure 1: Object recognition of common grocery items

Figure 2: Inaccuracy in object recognition of grocery items

on the other hand, can be achieved at a greater distance and within a less strictly defined region. We have implemented a voice interface which assists the user in positioning the barcode by directing them through a series of verbal instructions, describing how an object

must be moved in order for it to be successfully scanned. This minimizes the need for users to guess at proper object positioning.

Figure 3: Barcode object detector

```
Model Summary: 213 layers, 7015519 parameters, 0 gradients, 15.8 GFLOPs
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 5/5 [00:02<00:00,  2.10it/s]
                 all        142        163      0.862      0.818      0.833      0.596
                   0        142         22      0.737      0.636      0.674      0.535
                   1        142        141      0.988          1      0.993      0.657
Results saved to custom_yolov5/exp
CPU times: user 25.8 s, sys: 3.42 s, total: 29.2 s
Wall time: 40min 51s
```

Figure 4: Barcode object detector training results

The software also integrates a web-scraping technique, using Python's BeautifulSoup library (Richardson, 2007) to collect product data. When a barcode is scanned, a European Article Number (EAN) for that product is obtained, which we use to query for information on the item in the database Open Food Facts (2022). Open Food Facts maintains up-to-date information regarding ingredients, nutritional information, name, possible allergies and quantity for over two million products across the globe, making it a viable resource regardless of the user's country of residence. After the text data returned by our search is scraped from the database site, it is read aloud to the user via a voice interface. A short demonstration of this functionality can be found in a video at `https://youtu.be/z4ixJT4SuLI` and source code is available on Github (Arias Fernandez, 2022).

4. Curved Label Reader

4.1. Overview

As previously noted, labels printed on curved surfaces present difficulties not often addressed by existing text-to-speech aids. In order for OCR to be applied, columns of text must be fully visible in the frame of an input image. Text must not be obscured or warped by the curvature of the label, and clear column boundaries must be established. In any program of this kind, it is also important to provide users the ability to locate specific information represented on the label.

Existing reading aid applications approach these challenges in a number of ways. CVS Pharmacy's Spoken Rx makes medical information on pill bottles more accessible by using RFID labels, which may be scanned in the CVS app to trigger playback of an audio summary (Blanchette, 2021). Other solutions, like Be My Eyes (Velux Foundation, 2022) and AccessaMed (2022), rely on readings provided by volunteers or pharmacists. SeeingAI, previously discussed, provides a highly accessible automated option; it reads a continuous stream of the text captured within the frame of a cellphone camera. Using SeeingAI, the user would center the information they wished to read from a curved label using audio queues. An audio stream consisting of coherent sentences would indicate that a column of text was properly centered in-frame (Garage, 2022).

This approach may not perform optimally in all cases. The format of some label packaging will not allow for a single column of text to fit neatly in a camera frame, producing incoherent realtime rendering. Furthermore, currently available solutions lack the search functionality useful in locating specific information, and it is difficult to implement this kind of feature using the aforementioned approach. Instead, we examine methods of saving an accurate body of label text to which a search might later be applied. Generally, we create a composite image of the label, stitch photos taken around its surface, and conduct a layout analysis, applying OCR to the final result. While the search implementation itself is beyond the scope of this proof-of-concept, the text yielded by this method is suitable for use in future search applications.

4.2. Implementation

Much of the functionality implemented in Ye et al. (2013) is available today in the form of Python libraries, which are suitable for use in a proof-of-concept application. We carried out image stitching in our prototype as described in the paper, and attempted to improve upon the layout analysis outlined there, with an eye towards facilitating search implementation in the future. The authors of (Ye et al., 2013) reconstruct flattened labels by detecting matching points between overlapping portions of the input images (extraction of Scale-invariant feature transform features), and then estimating the transformation matrix in order to stitch the images together (RANSAC algorithm). The same process is now made available for use through OpenCV's panorama-stitching functionality, making construction of a composite image trivial.

Sample images were taken at an orientation perpendicular to the camera's line of sight, with large areas of overlap between individual photos. Figure 5 displays sample images of a bottle's surface before and after stitching.

Figure 5: Example of label images prior to stitching, followed by a composite image

Following the creation of a composite image, a layout analysis must be applied to identify continuous columns of text. While there are a number of established, annotated datasets available that are suitable for training models to perform a layout analysis, none were found that specifically addressed the use case of bottle packaging. Labels employ formatting that is distinct from that of text documents like academic papers or magazines, leading to discrepancies between the variety of annotated data available and the items we wished to analyze. Additionally, it was rare to see labelling of full columns rather than paragraphs reflected in the existing data.

In order to approximate training on a large database of labels, we began by testing the performance of a model trained on a near-match dataset. Although the resemblance is not perfect, the PRImA Layout Analysis Dataset, composed of 305 annotated scans of magazine pages, contains documents with layouts reasonably similar to those of the labels we wished to analyze (PRImA, 2009-2022). Text is generally broken out into multiple columns, and images, tables, and colored titles appear in most items. We used an R-CNN model pre-trained on these documents to test the viability of PRImA as a representation of our desired use case. We ran the model on a clean sample image of a scanned label to avoid any complications due to poor image quality.

Figure 6: Example of a layout analysis by the PRImA-trained model

When run on the flattened label image, the PRImA-trained model broke text into small paragraphs as seen in Figure 6. The bounding boxes did not encompass entire columns or regions of text. A search implementation would require that the text surrounding a search result be read aloud to the user; if we were to use these results as they are, we would need to incorporate additional processing work to determine which bounding boxes existed in the same columns, for the sake of reading coherence.

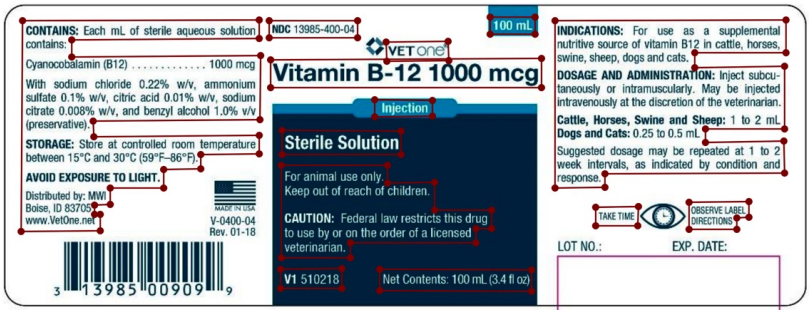

Figure 7: Example of a human annotation of the sample label

In order to improve the model's performance for our use case, we compiled a small dataset of 142 scans of flattened labels for the purposes of fine-tuning, sourced from dailymed.nlm.nih.gov, fda.report, and company websites. We hand-annotated these images in the COCO format using closely-fitting polygons, in the same way the original PRImA dataset was annotated, and split 100 of the images into a training set. Figure 7 displays a sample layout analysis by the resulting model, with bounding boxes now delineating broader regions of text. The remaining 42 images were used as a validation set, and a mAP score of 0.652 was calculated for the fine-tuned model.

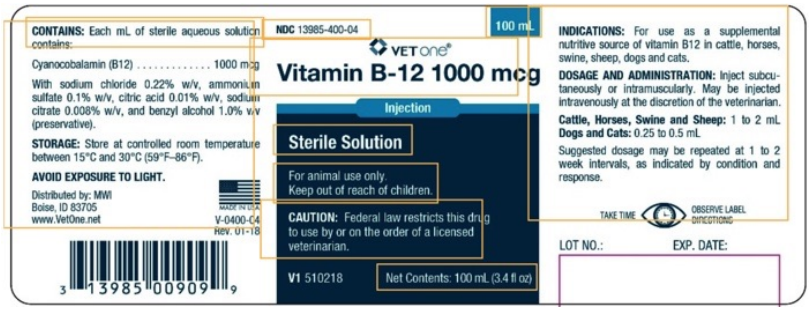

Figure 8: Example of a layout analysis by the fine-tuned model

The new regions may then be cropped and individually processed to extract text data. We used Tesseract OCR for optical character recognition, a sample of the results of which is displayed in Figure 9. In the clean sample image in Figure 8, Tesseract read text that was 94.6% similar to the actual label.

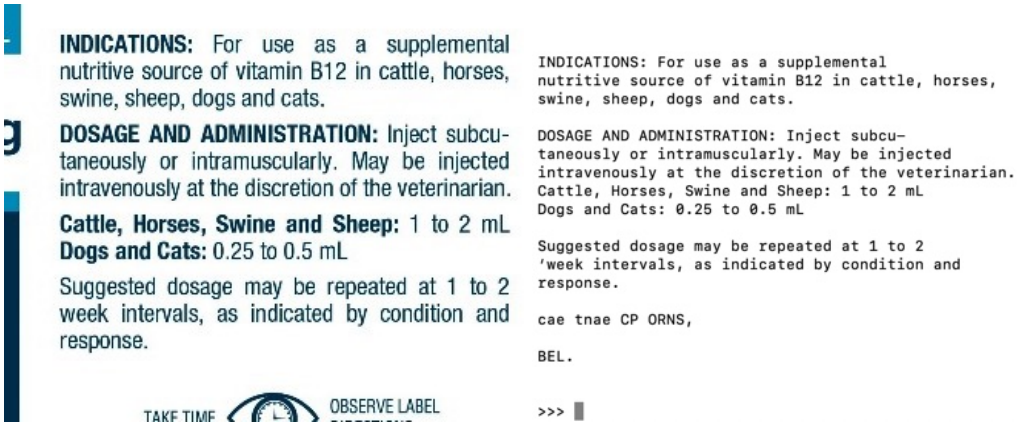

Figure 9: Example of OCR applied to a region detected with the fine-tuned model

5. Conclusions and Future Work

Future work might encompass a number of improvements to increase these tools' utility to visually impaired users. With greater processing capacity, the fine-tuning of the model used in the object detection interface could be significantly more extensive, which would diminish the number of erroneous classifications made. In addition, improvements could be made to the barcode scanning functionality; the accuracy of the centering directions provided by the voice interface could be improved, and the scanner itself could be made more flexible. Currently, the barcode needs to be held in close proximity to the camera in order to be detected and scanned effectively, which can complicate the system for the target demographic of users. The user experience would be improved if, instead, it were possible to recognize and scan the barcodes from further away.

Additionally, the text-search feature proposed for the curved label reader is yet to be implemented, and will need to be developed in further iterations of this work. Work on the label reader itself would also benefit from a number of other improvements. In real-world examples, warping of the text in an image can impede Tesseract's ability to perform OCR accurately. A clear next-step, then, would be to unwarp the label images prior to stitching to mitigate the curvature of the lines in the composite image and improve performance. In the layout analysis step, a larger training set and an increase in training time would improve the precision and accuracy of the model used. Finally, features are yet to be developed to assist users in centering objects in-frame. A voice interface might be developed in the next iteration of this work to improve accessibility.

References

AccessaMed. Digital audio label, 2022. URL https://www.accessamed.com/. Accessed: 2022-05-01.

Laura Arias Fernandez. GitHub, 2022. URL https://github.com/lauraAriasFdez/barcodeDetector.

Matthew Blanchette. Spoken Rx "talking" prescription labels now available in all CVS pharmacy locations, 2021. URL https://www.cvshealth.com/news-and-insights/press-releases/spoken-rx-talking-prescription-labels-now-available-in-all-cvs.

Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. arXiv:2004.10934v1 [cs.CV], 2020.

Rupert R A Bourne, Seth R Flaxman, Tasanee Braithwaite, Maria V Cicinelli, Aditi Das, Jost B Jonas, and et al. Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis. *Lancet Global Health*, 5(9):e888–e897, September 2017.

Microsoft Garage. Seeing AI: An app for visually impaired people that narrates the world around you, 2022. URL https://www.microsoft.com/en-us/garage/wall-of-fame/seeing-ai/#:~:text=Seeing%20AI%20app%20from%20Microsoft,nearby%20people%2C%20text%20and%20objects. Accessed: 2022-05-01.

Danna Gurari, Yinan Zhao, Meng Zhang, and Nilavra Bhattacharya. Captioning images taken by people who are blind, 2020. URL https://arxiv.org/abs/2002.08565.

Glenn Jocher. ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements. https://github.com/ultralytics/yolov5, October 2020. URL https://doi.org/10.5281/zenodo.4154370.

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. *International Journal of Computer Vision*, 2020.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollar. Microsoft coco: Common objects in context, 2014. URL https://arxiv.org/abs/1405.0312.

Aniq Masood. Recognition of grocery items in a shopping cart, 2015. URL http://web.stanford.edu/class/ee368/Project{_}Spring{_}1415/Posters/Masood.pdf.

Open Food Facts, 2022. URL https://world.openfoodfacts.org/.

PRImA. PRImA layout analysis dataset. University of Salford, 2009-2022. URL https://www.primaresearch.org/dataset/index.php.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2015. URL https://arxiv.org/abs/1506.02640.

Sana Riaz. Barcode detection annotated dataset. Kaggle, December 2020. URL https://www.kaggle.com/datasets/whoosis/barcode-detection-annotated-dataset.

Leonard Richardson. Beautiful Soup documentation, April 2007. URL https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

Leslie Sears, Ray Hashemi, and Mark Smith. Optical character recognition of non-flat small documents using android: A case study. In *Proceedings of the International Conference on Knowledge Engineering (IKE)*, 2011.

Velux Foundation. Be my eyes, 2022. URL https://www.bemyeyes.com/. Accessed: 2022-05-01.

Ze Ye, Chucai Yi, and Yingli Tian. Reading labels of cylinder objects for blind persons. In *2013 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, July 2013.