HIGH SCHOOL EDITION
Journal of Student Research
Volume 10 Issue 4 (2021)

Artificially Intelligent Food Assistant for the Visually Impaired

Sarthak Jain[1] and Evan Brociner[#]

[1]Los Gatos High School, Los Gatos, CA, USA
[#]Advisor

ABSTRACT

Vision disability is a prevalent condition that affects the lives of many adults and children. Previous research has established that it is harder for the visually impaired to evaluate the nutritional value of food. Therefore, by applying concepts used in agriculture and optical character recognition, we engineered a system that can make these perceptual evaluations on a variety of fresh and packaged foods and linguistically relay that info to a visually impaired user. We utilized some of the most memory-cheap and accurate object detection models to evaluate and then detect lesions in peaches, apples, tomatoes, and strawberries. Table 3 shows that our models performed with high accuracy as EffecientDet d0, SSD Lite Mobilenet, and Faster RCNN MobileNet all had higher than 40% mAP and 50% mAR. Figure 2 portrays how our interface was able to detect, determine, and relay surface spoilage percentage(s) to a user. Figure 3 shows that our OCR integration successfully was able to gather nutritional data on packaged goods and relay nutritional information to a user. Our system provides the brain for future applications that plan to deploy our code to devices like smart glasses or other hardware. We have made our source code available on GitHub through this link: https://github.com/SarthakJaingit/Artificially-Intelligent-Food-Assistant-for-the-Visually-Impaired. Our repository provides instructions about running our system through the command line and also a notebook demo that a less technical person can run to see how one of our models performs on a computer webcam with no video optimization.

Introduction

Vision disability is a prevalent condition that affects the lives of many adults and children. In the United States, an estimated 32.2 million adults and 2.19 million children suffer from visual impairment.[1,2] Moreover, visual impairment is correlated with higher BMI, obesity, and malnutrition. The negative health effects are known to hamper one's means to shop, cook, and consume food. Shopping and meal preparation are two activities that are compromised and are overall important to maintaining better nutrition.[3,4] An essential factor in both these activities is looking for fruits with blemishes or lesions and evaluating the overall freshness of the food that is either bought or consumed. Additionally, for items with nutrition labels, evaluating the content of such food products and extracting information about calories, fat, protein, and sugar is also important to maintaining health.

Such evaluations are made harder or even impossible due to one's visual impairment.[5] Despite this, there is a lack of a fast and reliable system built to evaluate food for the visually impaired. Consequently, there is a strong need for technology to enable the visually impaired to be in control of what food they buy, eat, and cook with. Machine learning advances offer opportunities to create quick and robust systems that learn from and make intelligent decisions from images. For example, the same methodology used to classify cats and dogs or localize cars on a road has had many significant implications for countless fields. In particular, novel use of previously produced and tested agriculture work has led to the emergence of faster and efficient food evaluation systems that heavily rely on machine learning.

The proposed device in this work aims to utilize and advance such technology proposed for agriculture and apply it to gauging freshness of agriculture to aid the decisions of the visually impaired who cannot or may have

trouble making such evaluations themselves. This work also utilizes advances in Optical Character Recognition (OCR) to create an image to nutritional data system that can be used by the visually impaired to evaluate processed foods that come in packages and have nutritional labels. The proposed device allows anyone visually impaired to shop, cook, and consume food significantly easier.

Related Work

Food Evaluation Deep Learning Systems

Previously, papers have utilized computer vision to evaluate food and its calories, size, freshness, or many of the other metrics used to quantify food. Numerous of these papers use some variation of a modified algorithm known as Convolutional Neural Network (CNN) along with a big labeled dataset to achieve their task. Formerly, traditional methods of incremental learning such as Support Vector Machines, Bayesian Learning, Decision Trees, and Fuzzy Logic have worked well to fit training data and mathematically estimate predictions using probabilistic, logical, or regression techniques. However, CNNs are able to examine feature representations or how groups of pixels function in an image, and therefore consistently outperform traditional incremental learning algorithms.[6–10] While a traditional linear neural network is built of different weights connecting to all the nodes, a CNN learns shared weights that are used to extract feature representations of an image. These shared weights are known as kernels, and when applied, they can transform a matrix's pixel values. These kernels can be useful for extracting a set of maps that correlate to the features of any image. Kernels can be high level features like edges, shape, and texture, or they can be lower level features like human eyes, car wheels, etc. A well-trained CNN contains kernels that best transform an image matrix to the most useful feature maps for a certain task like image classification or object detection. Training of the CNN is conducted through a calculation of total loss by comparing the ground truth to the model prediction in a loss function. The correlation between the possible weights and loss creates a loss curve and training helps approach the minimum of the loss curve in a process called Gradient Descent.

Optical Character Recognition (OCR) Advances and Applications For The Visually Impaired

OCR refers to the ability to extract textual information from just an image. Our paper utilizes modern OCR technology to extract health data like calories, sugar, protein, and fat from just an image. Most modern OCR technologies including the one the paper utilizes, are built through deep learning algorithms like CNNs that are able to detect and classify language on images. One major area of research that is missing from more traditional OCR applications is the ability to read unstructured text. Unlike structured text, unstructured text has no specific format or patterns. For example, most text on branded goods such as the brand name or the overall description of the food is unstructured since letters are never in straight lines and fonts vary.[11,12] OCR technologies have been used extensively as smart readers for the visually impaired; however, they still have a lot of potential in being used for more specialized tasks like helping the visually impaired shop for food.[13]

Experimental Datasets

Deep Learning Dataset

All the images that were used to train the model were web-scraped off Google images. Some of the words for the search utilized descriptors like rotten, spoiled, unfresh, black spot, etc. Throughout this investigation, the term "bad spots" will be used for non-fresh fruits. These bad spots consist of mold, rot spots, strong bruises, fungus, and any type of lesion that could be interpreted as making a fruit unfresh. Images that contained no fruits at all were included to improve the generalizability and robustness of the model, and were considered to be noise images. These images were collected from a dataset called ImageNet 1000 (mini) on Kaggle.[14] These images were then randomized choosing

one image per class. We then indexed 40 images to incorporate inside our dataset which effectively did not contain any images with fruits. We then architected a new data class that loaded 100 new noise images and then ran separate testing procedures on those images to evaluate the model's robustness.

The size of our train dataset was 546 images and the size of our valid dataset was 137 images. Bounding box labels totaled 2343 labeled bounding boxes. Out of the total 40 subsetted noise images, 32 were attributed to the train set while 8 were attributed to the valid set. The classes on our dataset consist of apples, strawberries, peaches, tomatoes, with their corresponding bad spots. The bounding boxes were labeled by the investigator of the study.

Proposed Method

To properly evaluate food for the visually impaired, our system must take into account two major statistics: how latent, efficient, and accurate the model is. The model would need to be efficient enough to make fast predictions while retaining high accuracy. In our case, this translates to finding what systems can best find rot amounts in fruits and health data on processed goods. Our paper proposes three models, the Faster RCNN MobileNet (FRMN), Single Shot Detection Lite Mobilenet (SSD), and the EffecientDet d0 (ED0) as solutions to evaluate food rot and freshness. Our models were built using Python packages such as Pytorch (1.9.0) and Torchvision (0.10.0). The ED0 model was built using an open source Pytorch implementation of Google's EffecientDet Models.[15] These models have been trained on an array of fruits: strawberries, apples, tomatoes, and peaches to exemplify model generalizability. These three models were chosen for their high detection performance despite having the least amount of parameters compared to other traditional approaches. To build upon the OCR functionality, the paper utilized Google's Cloud Vision API (2.4.2), text filtering algorithms, and finally a health data API called Calories Ninja.[16,17] The source code and also documentation on how to use the service can be found using this link: https://github.com/SarthakJaingit/Artificially-Intelligent-Food-Assistant-for-the-Visually-Impaired

Table 1. Parameter count of popular object detection models

Object Detection Models Utilized:	**Parameter Count**
EffecientDet d0	**3,842,663 parameters**
FasterRCNN MobileNet	**19,327,458 parameters**
SSDLite MobileNet	**3,440,060 parameters**
Base-line Object Detection Models:	**Parameter Count**
YOLOv5s	7,266,973 parameters
Faster RCNN Resnet 50	41,532,886 parameters
RetinaNet	33,792,599 parameters
SSD300_vgg16	35,603,106 parameters

Augmentations

Often in a deep learning task like object detection, images in a dataset are augmented with a variety of transformations. This process effectively increases the variance of the data and thus can help improve a model's generalizability and performance on unseen images.[18] To augment the images in the dataset first, the images in our train set went through a probability that either modified the HSV channel of the image or modified the brightness and contrast. Secondly, the image had a 50% possibility of being both vertically and horizontally flipped.[19] For the FasterRCNN Mobilenet and SSDLite Mobilenet, the images were resized internally based on default parameters exemplified by torchvision. Images tested and trained with the ED0 model were augmented through a color jitter and then a Resize Padding transformation. For the Resize Padding, the operation used bilinear interpolation for image scaling and filled the padded parts with pixels values of the image mean. Finally, the image was resized to 512 * 512 and then fed to the network.

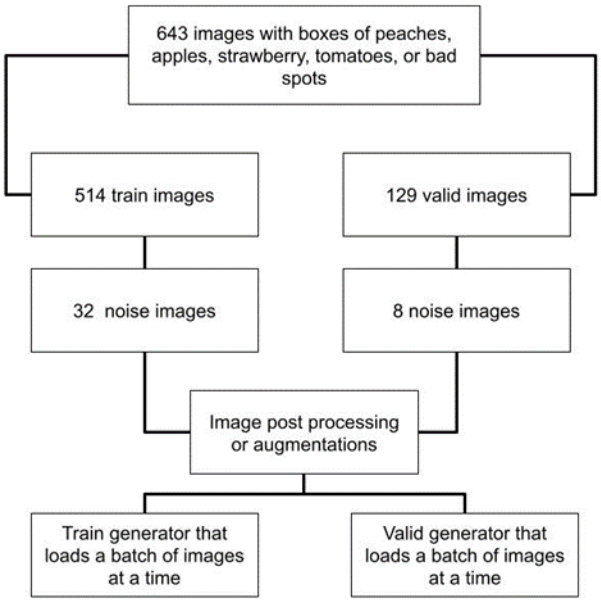

Figure 1. Entire Data Pipeline for Training Model

Model Initialization and Training

The hyperparameters, which are tunable values that determine how a model is trained, are portrayed down below in Table 2. Since the FRMN is a bigger model than the SSD Lite and the ED0, it takes significantly more time and resources to train the model for a long number of epochs. Our values for weight decay were chosen based on the fact that none of the models seemed to suffer terribly from overfitting, yet having a small weight decay we assumed would help. Through experimentation, we determined our chosen value for model ema decay seemed to perform well. Most other parameters were at their default package values.

Table 2. Hyper-Parameters for object detection models

Hyper-Parameter	FRMN	ED0	SSD	Explanation
Epochs	20 epochs	100 epochs + 10 cooldown epochs.	110 epochs	Epoch refers to a single iteration over all the train images. Cooldown epochs are extra iterations in which the model trains at a lower learning rate.
Warmup Epochs	0	3	0	Warmup Epochs in the amount of epochs that a model trains with a lower learning rate before entering its starting learning rate.
Batch Size	2	4	2	Batch size refers to the amount of inputs that are collectively passed into the network and optimized for. A higher batch size allows for faster but more memory-intensive computations while a lower batch size offers slower but less memory-intensive operations.
Starting Learning Rate	0.0005	0.0005	0.0005	Starting Learning rate refers to the size of the step the optimizer takes when updating your model at the beginning of the training process.
Learning Rate Scheduler	Cosine	Cosine	Cosine	Learning Rate Scheduler refers to a function that delineates how the learning rate value will change throughout the training process.
Weight Decay	0.001	4.00E-05	0.001	Weight decay is a form of regularization that penalizes your model for being too complex and containing too high values of weights.
Clip Grad	1	10	1	Clip Grad is a cut-off to prevent too extreme parameter updates in the model.
Model ema decay	None	0.0003	None	Value that places more importance on the most recent input into the model. Only ED0 utilizes this hyper-parameter.

Model Postprocessing and Inference

Non-max Suppression is a technique used to remove low confidence and duplicate boxes from the model output. For removing low confidence boxes, we allowed a confidence threshold parameter argument which can be determined by a user; however, thresholds from the range of 0.1 to 0.3 can be used for good performance for the object detection models. To remove duplicate boxes, we looked through the image to find and then remove weak confidence bounding boxes that have very high similarity or Intersection over Union(IOU) over a declared threshold with strong bounding boxes. To prevent detected bad spots from being dropped by this post-processing, we made the IOU thresholding process for bad spots and fruits independent of each other. Our inference script hardcoded thresholds of 0.35 for the fruits and 0.1 for the bad spots.

OCR Inference

The OCR system consisted of three primary building blocks: reading all text from an image; finding the most important text that refers to the processed food, and finally determining health data by using the name of the processed food as a query into a health database. To read all text from an image, we utilized the Google Cloud Vision API. After extracting information on all words and their respective bounding boxes coordinates on the image, we developed an algorithm to first calculate and sort the areas of each word from largest to smallest, than concatenated the first 6 largest text together into a list, then remove nonsensical strings with new line characters in them, and finally joined the list of processed text into a final string that could be passed to the Calories Ninja API. The Calories Ninja API is a database

that can report nutritional information. Therefore, the entire pipeline is able to output nutritional information based on the input of an image. The developed OCR and deep learning detection system is designed to be attached to a hardware device used by "smart glasses" and to be functional in a real-time setting.

Results

The accuracy of the model was measured through the mean Average Precision (mAP) and mean Average Recall metric (mAR). In simplified terms, both metrics measure how well predicted boxes of a class id intersect with ground truth boxes of the same class id. However, mAP takes into account the amount of boxes predicted that do not refer to a valid object, known as false positives, and mAR takes into account the amount of objects missed by the detector, known as false negatives. To calculate the mAP and mAR, we used the default COCO standard.[20] As exemplified by Table 3, all our models achieved acceptable accuracies, with ED0 being the best performing model for both mAP and mAR

Table 3. Model evaluation metrics

Model Name	Mean Average Precision (mAP)	Mean Average Recall (mAR)
SSD	41.2%	56.9%
ED0	**69.7%**	**78.1%**
FRMN	51.6%	59.1%

Our three models were able to detect lesions and unfresh bad spots in a variety of fruits. Figure 2 depicts the FRMN output on an example image on the web. This performance embodies human-level perception at identifying bad spots and their corresponding fruit. Additionally, our built interface is able to estimate the surface spoilage percentage in different fruits based on the results outputted by the model. In this example of Figure 2 the detected apple is estimated to have roughly 37.3% surface spoilage. Additionally, if there are fruits that the model recognizes in the image, then the interface is able to spatially describe the location, the fruit name, and finally the surface spoilage of each detected fruit.

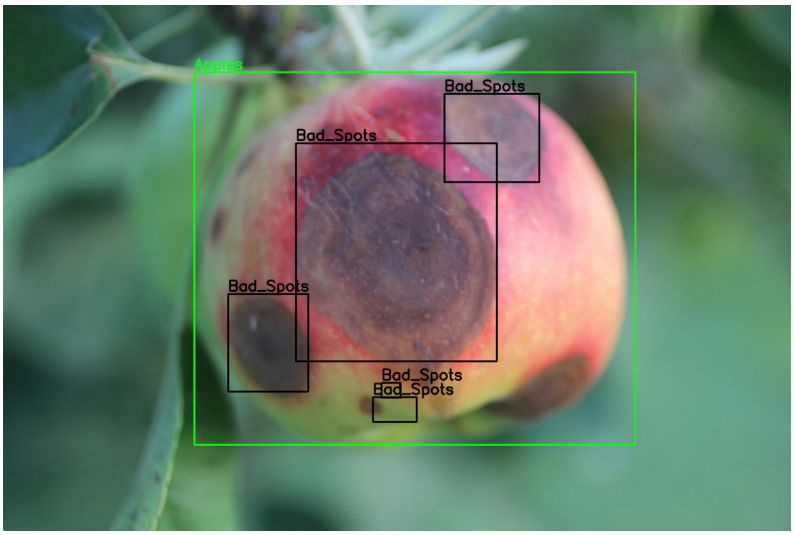

Figure 2. This side of the apple is detected to have roughly 37.3% of surface spoilage[21]

While the models performed at high accuracies, we found that the models sometimes misclassified round objects of non-trained labels as fruits. Previous unreported experiments had shown that the addition of the "noise" class partially subdued this problem in the model. Another problem we did not fully address is the possibility of occluded fruits. While the model still can detect partially occluded fruits, there is still no clear way to obtain its surface spoilage since a good portion of the fruit is hidden to the camera. Currently, if multiple fruits are close together such that the IOU of their bounding boxes is higher than 35%, then only the bounding box covering the most visible fruit, which would likely be the most confident detection, is kept, hence ignoring occluded fruits that were most likely not intended to be analyzed. However, this thresholding won't apply if fruits are occluded by the environment rather than another fruit. Although less probable due to the use case of this system, if the amount of undetected occlusions is high, then no caption will be produced, because our interface was programmed to not output surface spoilage percentages if too many fruits are in view. Additionally, we were able to deploy our model and inference abilities onto a webcam. One can run this webcam demo by following the instructions in our project's GitHub repository.

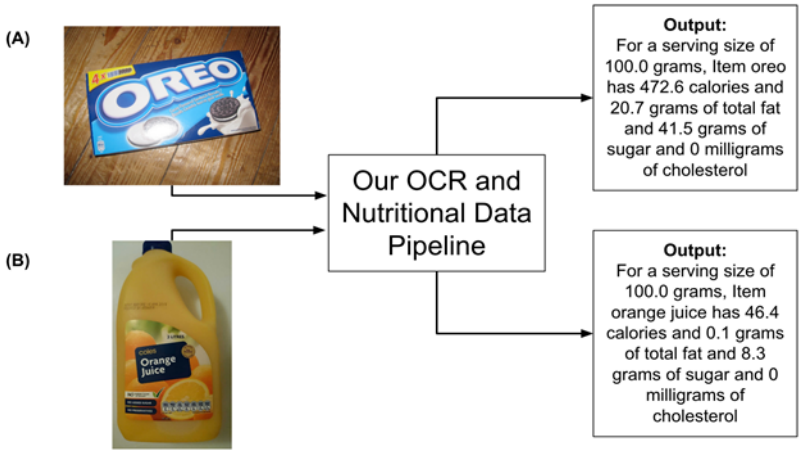

Figure 3. OCR output for two packaged food items: Oreos[22] and Orange Juice[23]

Our OCR integration was successful in gathering nutritional data for a wide variety of packaged foods. As shown in Figure 3a and 3b, the system successfully used the Google Vision API and our area sorting algorithm to gather text such as common brand names such as "Oreo" and generic food names "Orange Juice" in order to pass them to the Calories Ninja API. While the model performed well on many different packaged foods, our integration did struggle when dealing with less household brand names due to limitations in the database of Calories Ninja API. Additionally, there were limitations of the Google Cloud Vision API in detecting correct text in poor contrast lighting. Finally, the model sorting algorithms did have a little more difficulty in filtering text to obtain compound words like "fruitcake" or "peppermint"; however, as exemplified in Figure 3b, the system still functioned correctly in many cases.

Discussion

Our intelligent food assistant utilizes deep learning and OCR to be able to aid the visually impaired in making visual-based evaluations for many fresh and manufactured goods. Successfully, we built an end-to-end system that is able to detect fresh fruit and packaged food and relay the name, nutritional information, and potential surface spoilage to its

user. While previous work, particularly in the agriculture field, has already applied different deep learning functionality to evaluate freshness of produce, our paper was able to build an interface that is applicable to suit the needs of the visually impaired. Unlike previous work that evaluated the inside of fruits or used less accurate prediction models, our paper utilized both some of the most currently memory affordable and accurate models to evaluate fruits from only their surface and also utilizes the algorithms to linguistically relay that information to someone who is visually impaired.

The proposed system allows for a more useful functionality since when our interface is deployed to a camera, the interface will be able to generate real-time, useful feedback by viewing an available fruit that would be residing in a market, grocery store etc. Also, our paper developed an application for OCR that by integrating a state of the art API powered by Google, an algorithm that sorts text importance, and a nutritional database API, is able to relay nutritional information about packaged goods to the visually impaired. While there are still limitations in the total amount of fruits our interface can predict on and image configurations in which it struggles to obtain meaningful output, our paper is overall able to present a fully functional food assistant that can help the visually impaired shop, buy, eat, and overall interact with food.

Our research exemplifies how advancements in artificial intelligence can provide value to people who are visually impaired. Further research that would expand on the advancements of our paper should have an emphasis on finding innovative ways to deploy our interface such that a more usable product can be formulated. Future directions include, the development of an interface that interacts with users through the use of hardware attached to eyewear such as Google Glass or OrCam. We also are considering implementing a box installed with cameras that can evaluate food through a 360 degree view. Overall, our application of evaluating food can massively and positively change the way the visually impaired interact with food.

Acknowledgments

I would like to thank my advisor Evan Brociner for helping me with this project.

References

1. Blindness Statistics. The American Foundation for the Blind. Accessed October 14, 2021. https://www.afb.org/research-and-initiatives/statistics

2. Fast Facts of Common Eye Disorders | CDC. Published June 9, 2020. Accessed October 14, 2021. https://www.cdc.gov/visionhealth/basics/ced/fastfacts.htm

3. Bilyk MC, Sontrop JM, Chapman GE, Barr SI, Mamer L. Food experiences and eating patterns of visually impaired and blind people. Can J Diet Pract Res Publ Dietit Can Rev Can Prat Rech En Diet Une Publ Diet Can. 2009;70(1):13-18. doi:10.3148/70.1.2009.13

4. Jones N, Bartlett H. The impact of visual impairment on nutritional status: A systematic review. Br J Vis Impair. 2018;36(1):17-30. doi:10.1177/0264619617730860

5. Kostyra E, Żakowska-Biemans S, Śniegocka K, Piotrowska A. Food shopping, sensory determinants of food choice and meal preparation by visually impaired people. Obstacles and expectations in daily food experiences. Appetite. 2017;113:14-22. doi:10.1016/j.appet.2017.02.008

6. Zhu L, Spachos P, Pensini E, Plataniotis KN. Deep learning and machine vision for food processing: A survey. Curr Res Food Sci. 2021;4:233-249. doi:10.1016/j.crfs.2021.03.009

7. Koyama K, Tanaka M, Cho B-H, Yoshikawa Y, Koseki S. Predicting sensory evaluation of spinach freshness using machine learning model and digital images. PLOS ONE. 2021;16(3):e0248769. doi:10.1371/journal.pone.0248769

8. Du C-J, Sun D-W. Learning techniques used in computer vision for food quality evaluation: a review. J Food Eng. 2006;72(1):39-55. doi:10.1016/j.jfoodeng.2004.11.017

9. Karakaya D, Ulucan O, Turkan M. A Comparative Analysis on Fruit Freshness Classification. In: 2019 Innovations in Intelligent Systems and Applications Conference (ASYU). ; 2019:1-4. doi:10.1109/ASYU48272.2019.8946385

10. Horiguchi S, Amano S, Ogawa M, Aizawa K. Personalized Classifier for Food Image Recognition. ArXiv180404600 Cs. Published online April 8, 2018. Accessed October 14, 2021. http://arxiv.org/abs/1804.04600

11. Memon J, Sami M, Khan RA, Uddin M. Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR). IEEE Access. 2020;8:142642-142668. doi:10.1109/ACCESS.2020.3012542

12. Ahmed M, Abidi A. REVIEW ON OPTICAL CHARACTER RECOGNITION. In: ; 2019.

13. Zamir MF, Khan KB, Khan SA, Rehman E. Smart Reader for Visually Impaired People Based on Optical Character Recognition. In: Bajwa IS, Sibalija T, Jawawi DNA, eds. Intelligent Technologies and Applications. Communications in Computer and Information Science. Springer; 2020:79-89. doi:10.1007/978-981-15-5232-8_8

14. Ilya F. ImageNet 1000 (Mini).; 2020. https://www.kaggle.com/ifigotin/imagenetmini-1000/metadata

15. Wightman R. Efficientdet-Pytorch.; 2021. https://github.com/rwightman/efficientdet-pytorch.git

16. Cloud Vision documentation | Cloud Vision API. Google Cloud. Accessed October 14, 2021. https://cloud.google.com/vision/docs

17. CalorieNinjas - Easy, Free Nutrition Facts Search. Accessed October 14, 2021. https://calorieninjas.com/

18. Shorten C, Khoshgoftaar TM. A survey on Image Data Augmentation for Deep Learning. J Big Data. 2019;6(1):60. doi:10.1186/s40537-019-0197-0

19. Buslaev A, Parinov A, Khvedchenya E, Iglovikov VI, Kalinin AA. Albumentations: fast and flexible image augmentations. Information. 2020;11(2):125. doi:10.3390/info11020125

20. Cocodataset/Cocoapi. cocodataset; 2021. Accessed October 15, 2021. https://github.com/cocodataset/cocoapi

21. apple-5265125_1280.jpg (1280×853). Accessed October 15, 2021. https://cdn.pixabay.com/photo/2020/06/06/03/52/apple-5265125_1280.jpg

22. 4857682389_7b13e44deb_b.jpg (1024×768). Accessed October 15, 2021. https://live.staticflickr.com/4136/4857682389_7b13e44deb_b.jpg

23. front_en.10.full.jpg (1125×2000). Accessed October 15, 2021. https://world.openfoodfacts.org/images/products/930/060/145/1272/front_en.10.full.jpg