Detection and Ripeness Classification of Bananas Using Deep Learning Methods

Dinis Rocha Nuno Machado Tiago Miranda

I. Problem description

Ripening is the process in fruits and vegetables that causes these to become more palatable. In general, fruit becomes sweeter, less green, and softer as it ripens. These characteristics are desirable for most animals as eating unripe fruit can often lead to stomachache or stomach issues. Therefore, determining the ripening stages of each fruit is a skill developed by most animals. Recent advances in human civilization lead to the global availability of many different varieties of fruit and vegetables, inevitably contributing to the increase of food waste by missing the optimal consumption period of these items (from the client perspective as well as the manufacturer and reseller). This project intends to tackle this issue and aid the client or the seller to reduce food waste.

II. Why is the problem important?

Food waste is a major issue in today's society that is often understated and underestimated. One third of all food produced is lost or wasted (around 1.3 billion tonnes of food) costing the global economy close to 940 billion US dollars each year. Almost half of all fruit and vegetables produced are wasted (US citizens alone throw away 5 billion bananas each year).

Another reason to tackle this issue is that food waste is terrible for the climate. Up to 10% of global greenhouse gases come from food that is produced, but not eaten. Wasting food is worse than total emissions from flying (1.9%), plastic production (3.8%) and oil extraction (3.8%). If food waste was a country, it would be the third biggest emitter of greenhouse gases after USA and China.

Therefore, for these and many more ethical and environmental reasons, food waste is a major societal problem that needs addressing ([1] and [2]).

III. How is the problem addressed?

For the sake of simplicity, this work was particularly focused on reducing the waste of bananas. To address this problem, we proposed a system that would classify individual bananas according to their ripeness stage and predict the time it would take for them to reach a desired ripeness level. It is clear that such a solution would allow consumers to make better decisions and the industry to allocate these resources more wisely.

The problem of ripeness classification is well-documented in the literature. Concerning the signals used, classification techniques may be divided into destructive techniques, usually measuring dielectric parameters of the fruit, and non-destructive ones, using color or infrared images, among others. The review and meta-analysis carried out in [13] offer a good overview of the different possibilities as well as their strengths and weaknesses, and suggest that color image classifiers are among the best, with the added benefit of being a low-cost, easy-to-access technology. As for the computational techniques used in classification, deep learning, and in particular CNNs, seem to be the better alternative for this choice of data, and we will discuss them in greater detail in the next section.
We were not able to find useful information for the time prediction problem specialized to our setting. It is clear that a deep learning approach to estimating "time until a certain level of ripeness is reached" would require abundant data, not only of bananas in different stages of maturation together with time information, but also of the different environmental conditions they might be subject to. Furthermore, these very environmental conditions would be necessary to know at time of prediction, which could be unfeasible in a consumer application. Instead, we opted for a simple regression approach which we describe in the next section.
Finally, from the point-of-view of the consumer, it could be useful to have an interactive test to gauge the preferred ripeness level. To that end, we developed a dating app-like test where several bananas are shown in sequence and the test-taker clicks left or right to indicate whether or not they like them.

IV. System architecture and main modules

The ripeness predictor, corresponding to the second stage outlined above, may be divided into three modules, as shown in Figure 1.
The process begins with the acquisition of image or video. Afterwards, it is necessary to locate the bananas in the frame, if any exist. This can be achieved with object detection tools, that identify and locate instances of one or more classes in an image. At the time of writing, deep learning is the preferred approach to object detection problems for which there is plenty of data available. Whereas previous solutions would have built-in "hard-coded" information about the objects, neural networks, namely CNNs, learn them from the large amounts of data [7]. In particular, the state-of-the-art YOLO series of models [3], [4] is a fast and robust option to

consider. It yields bounding boxes containing the instances found as well as confidence levels assigned to each one.
We observe that, this being an intermediate step of the process, the best way to approach it is determined by the needs of the ensuing procedure. Indeed, it may prove useful not only to locate the bananas but also to remove the background, i.e. to perform segmentation, so that the environment doesn't influence predictions. Fortunately, in the most recent version of YOLO (v8) [4], a subclass of efficient segmentation models is available. These models range from small (3M parameters) to quite large (70M parameters), offering a trade-off between accuracy - larger models are more accurate - and speed - smaller models are faster. It is worth noting that there are appealing alternatives to YOLO for segmentation, namely the recent Segment Anything (SA) model by Meta [6].

Several techniques were found to classify the ripeness stage of bananas using Machine Learning methods. Some by simply resizing the images and inputting them into a *CNN* [8] [9]. Others by converting the color space to *HSV* and then calculating the ratio of brown areas to the whole banana area and using this value and *HSV* values as inputs to a Neural Network [10]. The use of an *infrared* source could also provide additional data to the model [11]. Data augmentation techniques could also be implemented in order to balance the data and improve the robustness of the model. Another possible solution would be to combine several datasets, however some problems may arise because of the different labeling methods used. The exterior of a banana might not be the only indicator of ripeness, for example if stored in a fridge they might go brown while the interior remains unripe, so it might be interesting to try to use, for example, a measurement of its dielectric constant which is related to the ripeness level, however datasets for this kind of data might be limited [12] [13]. The use of attention based models such as transformers could also be interesting to explore, even though these are often considered to be less effective in classification tasks when compared to *CNN*'s.

An additional part of the project would be to first assess a user's preference regarding the ripeness of a banana, possibly using a decision tree, and then given a photo of a banana determine the time it will take for that banana to reach desired ripeness. This, however, will be challenging considering the lack of datasets suited for this time analysis. A possible solution would be for us to create our own dataset, but there's no guarantee that the results will be satisfying.

A. Dataset

We combined the dataset from ([8]) with one built by ourselves where four ripeness stages were considered (Green, Yellowish Green, Partially Ripe and Over-ripe). We built our own dataset because we needed to track the evolution of bananas through time and no dataset for this purpose was found. Additionally, since our goal was to train a robust classification model and the datasets available lacked variability regarding the lighting, angle, background and camera used to capture the images, this was an important add-on.

In our dataset the images were obtained with different cameras, angles and backgrounds (the dataset is available in [18]). We also tracked each banana through a period of about 7 days, in which several pictures were taken once per day, since the time of the day changed the total number of hours from the initial picture was registered for each picture taken.

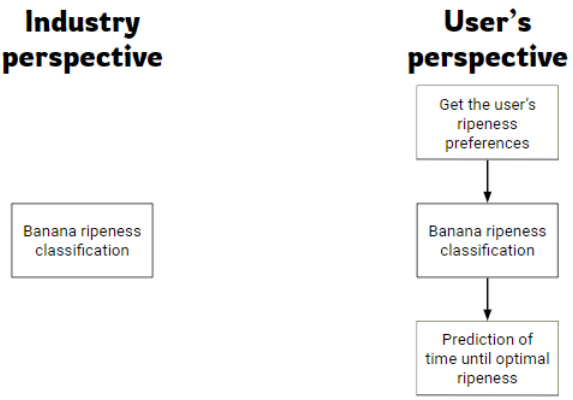

Fig. 1: Proposed workflows

B. GUI

In order to make the project more accessible to the end user we developed a Graphical User Interface (GUI) using *Tkinter*.

C. Ripeness preferences module

In this module the goal is to obtain the user's preferential banana ripening stage allowing the system to output personalized results. To achieve this we implemented a decision tree algorithm where the user is presented several pictures of bananas in different ripening stages and they answer whether they would eat the banana shown. After answering this question for a few bananas an ideal ripening stage score is obtained.

D. Object Detection and Segmentation Module

With the intent of delivering a real-time solution for the end consumer, speed was a decisive factor for picking YOLO rather than SA. Furthermore, there was an attempt to employ the smallest - and thus fastest - YOLO model available (*yolov8n-seg*). The model was originally performing very badly at segmentation of bananas. To address this, the following strategy was attempted: a larger model that performed very well (*yolov8l-seg*) was used to annotate the available data. Then, this data, together with the annotations, was used to fine tune the smaller model. This strategy ended up improving considerably the original model, but not enough to satisfy the needs of this project. We conjecture that the limiting factor was the lack of enough training data. In the end, the larger YOLO model was employed in the final version of the workflow.

Fig. 2: Before and after segmentation and cropping of bananas from the dataset built

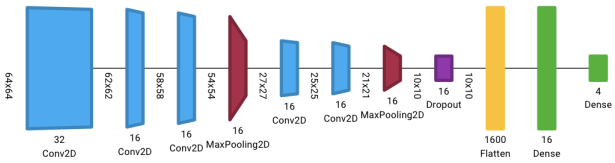

Fig. 3: CNN diagram (using [14])

E. Ripeness Classification Module

1) Dealing with Class Imbalance and Data Augmentation Techniques: Several techniques such as using different class weights for the loss function during training were considered. However we decided to use data augmentation since it usually leads to a more robust model.

Each image was generated by randomly flipping, rotating (angle from -90º to 90º), translating (from (-10, -10) to (10, 10)) and adjusting the brightness (from a factor of 0.5 to 1.5).

With the goal of creating a more robust model we also wanted to create more variability in the classes with fewer data points, specially in the images from ([8]). Therefore we also performed data augmentation on those.

The data augmentation was only applied to the training set since we wanted the validation data to have no artificially generated images. This ensures a more accurate assessment of the model.

The initial distribution per class was: Green-83, Yellowish green-88, ripe-96 and over ripe-40, after the data augmentation methods were applied we had 896 examples per class for the training set.

2) Pre-Processing: The images were read and converted to HSV using the *cv2* library. We converted the images to this colour-space to make it easier for colour based features to be extracted and make the model more invariant to different brightness levels in pictures.

Before the images are fed to the CNN these are resized to 62x62x3. To improve the convergence speed of the model the image data was normalized by dividing each value by the maximum of each HSV channel (179, 255, 255).

3) CNN: We based our CNN architecture on the one from ([8]), where it was compared to much larger ones such as *VGGNet16* and *ResNet50*, obtaining similar results and requiring less time to train. This was important considering our limited computational resources. The activation functions used were ReLu's, except for the last layer where a softmax was applied. The CNN outputs an array with four elements corresponding to the likelihood of the banana being at each ripening stage. To obtain a ripeness score we calculated the weighted average of these likelihoods.

In order to restore the model to the best epoch when looking at the validation set loss, we used the early stopping callback from *keras*. The *reducelronplateau* callback was also used to improve the model's convergence.

4) Loss: Since our labels are ordered we wanted a wrong prediction that was two classes away to be more penalized than one that was only one class away and so on. Therefore we decided to use an ordinal categorical cross entropy loss function ([17])

5) Hyperparameters tuning: Doing an exhaustive search for the optimal hyperparameters of a CNN is a computational demanding task so considering the resources and time available we only explored changes regarding the learning rate and batch size, using the *keras* tuner library ([16]), in which we used the *hyperband* tuning method ([15]). When compared to methods such as Bayesian optimization, random search and other traditional hyperparameter tuning methods, this method is usually more time efficient while providing similar results.

F. Ripeness Evolution Through Time Module

In order to estimate the time it'll take a banana to reach a certain level of ripeness we built a dataset with photos of a few bananas throughout the course of several days. Then, we assumed the ripeness stage evolution to be linear through time and so performed a linear regression on the CNN results obtained from the training dataset pictures. However, these results were unsatisfactory given the classification module volatility so we opted to use ground truths (manually labeled photos) as input to the linear regression.

G. Experimental Results

As a way of accessing the performance of the YOLO model, we verified that out of the 273 images from ([8]), only in 6 did the model fail to correctly segment the bananas. Furthermore,

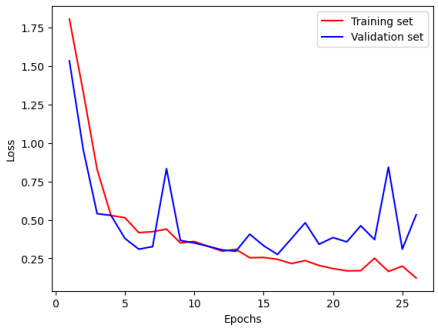

Fig. 4: Loss during training

it only failed in the *Over-ripe* category. On our own data, where conditions of image acquisition were more varied, we verified that the model produced a satisfactory segmentation in 83% of images.

To train the CNN we used the augmented version of the combined datasets. Before performing the data augmentation we did a 60-40 split into training and validation. The validation data was not augmented. After being resized and converted to the HSV color space the images' data were also normalized to improve convergence.

In figure 4 the training results of the CNN are presented for the training and validation sets, where the lowest validation loss was obtained for the epoch 16 with a value of ≈ 0.28 and a validation accuracy of ≈ 0.91

In order to better assess how well the model performed we calculated the F1-scores for each class obtaining the following results, where *Support* is the number of examples present in the validation set for each class:

Class	Precision	Recall	F1 score	Support
Green (0)	0.99	0.95	0.97	74
Yellowish Green (1)	0.77	0.91	0.83	47
Ripe (2)	0.95	0.83	0.88	63
Over-ripe (3)	0.92	1.00	0.96	22

TABLE I: Per class metrics of the model

Analysing the table we can see that the lowest F1-scores are for the *Yellowish Green* and *Ripe* classes, this is expected since the boundary between the two is hard to define. This was one of the issues we faced when building our own dataset and it's, without surprises, reflected in the model. We tried to mitigate the impact of this issue by using a ordinal categorical loss, described previously.

For the ripeness time evolution model we obtained a slope of $m = 0.00777$ class units/hour $= 0.1865$ class units/day by averaging the linear regression results obtained for the different bananas. We used this result in order to predict the time it'll take for the banana considered to reach the user's optimal ripeness stage.

V. Conclusion

In this work, we were able to produce a working application that yields satisfactory results. Importantly, each of the modules we developed seems to perform its task very well: segmentation yielded good results in most cases, the validation accuracy obtained with the CNN was on par with that found in the literature, and the method used for time prediction is simple yet robust. It must be said that when putting all the modules together, the results weren't as good as these individual performances would lead us to expect. We believe that a factor that played a major role in this was the lacking quantity and, on closer inspection, quality of the pre-exciting datasets. In future works, this problem should be addressed by gathering more data as well as by careful labelling. Another interesting aspect to improve would be to host the project in a web page, thus allowing easier and more widespread access.

References

[1] https://www.ozharvest.org/food-waste-facts/
[2] https://www.chicagotribune.com/opinion/commentary/ct-opinion-food-waste-20210409-3k3llled4fbmlp3nwhiej3o354-story.html
[3] C. Yang, A. Bochkovskiy, H. M. Lia0 (2022). 'YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors'. arXiv:2207.02696.
[4] J. Terven, D. Cordova-Esparza (2023). 'A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond'. arXiv:2304.00501.
[5] S. Rath (2022). 'Fine Tuning YOLO v7'. https://https://learnopencv.com/fine-tuning-yolov7-on-custom-dataset/
[6] A. Kirillov et al (2023). 'Segment Anything'. arXiv:2304.02643.
[7] Y. Amit, P. Felzenszwalb, R. Girshick (2020). 'Object Detection'. In: 'Computer Vision'. Springer, Cham. https://doi.org/10.1007/978-3-030-03243-2_660-1
[8] N. Saranya, K. Srinivasan, S. K. Pravin Kumar (2021). 'Banana ripeness stage identification: a deep learning approach'. Journal of Ambient Intelligence and Humanized Computing. https://doi.org/10.1007/s12652-021-03267-w
[9] Y. Zhang, J. Lian, M. Fan and Y. Zheng (2018). 'Deep indicator for fine-grained classification of banana's ripening stages'. EURASIP Journal on Image and Video Processing. https://doi.org/10.1186/s13640-018-0284-8
[10] F. M. A. Mazen, A. A. Nashat (2018). 'Ripeness Classification of Bananas Using an Artificial Neural Network'. Arabian Journal for Science and Engineering. https://doi.org/10.1007/s13369-018-03695-5
[11] L. LUIZ, C. A. NASCIMENTO1, M. J. V. BELL, R. T. BATISTA, S. MERUVA, V. ANJOS (2021). Use of mid infrared spectroscopy to analyze the ripening of Brazilian bananas. Food Science and Technology. https://doi.org/10.1590/fst.74221
[12] M. Soltani, R. Alimardani, M. Omid (2011). Evaluating banana ripening status from measuring dielectric properties. Journal of Food Engineering. www.elsevier.com/locate/jfoodeng
[13] P. Baglat, A. Hayat , F. Mendonça, A. Gupta, S. S. Mostafa and F. Morgado-Dias (2023). Non-Destructive Banana Ripeness Detection Using Shallow and Deep Learning: A Systematic Review. https://doi.org/10.3390/s23020738
[14] A. Bäuerle, C. van Onzenoodt and T. Ropinski, "Net2Vis – A Visual Grammar for Automatically Generating Publication-Tailored CNN Architecture Visualizations," in IEEE Transactions on Visualization and Computer Graphics, vol. 27, no. 6, pp. 2980-2991, 1 June 2021, doi: 10.1109/TVCG.2021.3057483.
[15] L. Li and K. Jamieson and G. DeSalvo and A. Rostamizadeh and A. Talwalkar (2018). 'Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization'. Journal of Machine Learning Research. http://jmlr.org/papers/v18/16-558.html
[16] O'Malley, Tom and Bursztein, Elie and Long, James and Chollet, François and Jin, Haifeng and Invernizzi, Luca and others (2019). https://github.com/keras-team/keras-tuner
[17] https://github.com/JHart96/keras_ordinal_categorical_crossentropy
[18] https://drive.google.com/drive/folders/1wvjGkn_xVQdRusSeN9IONGPJ6K_7MMij?usp=sharing