# CS7GV1 Computer Vision Group 6 Project
# Strawberry Counting and Ripeness detection

Liam Bartsch, Liam Byrne, Nisarg Kalyani
Trinity College Dublin
bartschl@tcd.ie, byrnel45@tcd.ie, kalyanin@tcd.ie

## Contents



## Abstract

*In this paper, we present a group project on strawberry counting and ripeness detection. We describe the motivation for the project, which is to develop a system that can accurately count the number of strawberries in a given image and determine their ripeness level. We then discuss the methods we used to develop the system, including image pre-processing, object detection and classification, and post-processing. We also present experimental results on a data-set of real-world images, showing that our system achieves high accuracy in both strawberry counting and ripeness detection. Overall, our project demonstrates the feasibility of using computer vision techniques for automated strawberry counting and ripeness detection.*

## 1. Introduction

Strawberry counting and ripeness detection are important tasks in the strawberry industry, as they help growers and harvesters to efficiently and effectively manage their crops. Automation, high accuracy, real-time processing, scalability, and versatility are some of the reasons why computer vision is a prime candidate for this application.

By accurately counting the number of strawberries on a plant, growers can estimate their yields and plan accordingly. Additionally, being able to detect the ripeness of strawberries can help ensure that they are picked at the optimal time, leading to higher-quality fruit and better yields.

In recent years, computer vision and machine learning advances have made it possible to automate these tasks using algorithms that can accurately detect and count strawberries in images and videos. This has the potential to improve the efficiency and productivity of the strawberry industry greatly.

## 2. Background

The first machine vision system presented [3] utilizes a deep convolutional neural network for segmentation to detect the strawberries. This is then paired with an RGB camera that captures depth to allow for more accurate results. Using a Hough transform algorithm this system achieves a 74.1% accuracy overall.

A different approach is to use a faster CNN (Convolutional Neural Network) [9]. This approach achieves an 86.0% accuracy on average. Overall the paper presents a promising approach to help alleviate issues with labor costs and automated detection of strawberries.

A fruit harvesting robot is used as the basis for the next system [4]. The system starts off with a segmented base image, the strawberries are then classified based on their shape feature. Finally, histogram matching is performed to judge the shape of each fruit, these results achieve 93% on the stem detection of the fruit as well as 90% on the ripeness. This is very helpful to help detach the fruit from the plant.

A similar approach with a system mounted on a robot uses machine vision to detect and classify the strawberries in three categories: ripe, intermediate, and unripe [5]. This process used HSI (Hue, Saturation, Intensity) images which yielded a coefficient of 0.84 in comparison to human judgment.

Similarly using HSI, a different paper decided to focus on the wavelength to analyze the ripeness of strawberries [2]. In this paper, they use an SVM (Support Vector Machine) since this helps with classifying the fruit's ripeness. They end up with a high ROC (Receiver Operating Characteristic) of 0.95 which means their model is performing very well, this ends up with a 98.6% accuracy rate which is very impressive.

A modified version of the CNN based on the DCNN (Deep Convolutional Neural Network) leads to a high accuracy score of 0.88 [7]. This paper does some interesting work in the occlusion of the fruits and detection of partially hidden fruit. This model can still be improved by looking at better-occluded situations and coping with those outliers.

A simpler model uses 3 RGB cameras and fully autonomously picks the fruit, this is a very promising system since it needs no human intervention [1]. One main limitation of this system, however, is that it relies on the tabletop method which is not universally implemented, implementing the tabletop is costly and not very adaptable.

In another study, they also use SVM's as well as spectral analysis to determine the ripeness of the fruit [8]. The results of this study however are covering a short limited range of wavelength. This means that it's highly effective for certain conditions but not in all, this can be further improved by making it more flexible.

The final study looks at a novel aspect of the strawberry collection, with an early detection method [6]. This novel method takes time to predict results but can be very useful in the long term for predicting yields and having more efficient robots that harvest autonomously.

Overall, these studies present promising approaches for automating the detection and classification of strawberries, but also identify areas for further improvement, such as better handling of occlusion and a more flexible approach to wavelength analysis.

## 3. Approaches

Based on the background review in 2 we initially tried to use a CNN to train a convolutional neural network (CNN) using object detection techniques to count the strawberries. After some reflection however we decided to try YOLOv7 and YOLOv5. We managed to use YOLOv7 to detect the strawberries. A few other possible systems were investigated as well.

1. Image segmentation and object detection. This approach would use image segmentation to separate the strawberries from the background and then use object detection algorithms to count the strawberries. Once we've counted and identified the strawberries we would then classify them using Support Vector Machines.

2. The second approach would use feature extraction and convolutional neural networks (CNNs) to identify and classify the strawberries.

3. The third approach would be to define a set of rules based on the shape, size, and color of the strawberry and then use these rules to classify and identify each strawberry.

4. The fourth approach, Instance Segmentation and Localization, locates each strawberry in an image and give it a unique label, then trains a deep learning model to detect and classify each strawberry based on its label and bounding box.

In the end, the training took longer than expected which meant we weren't able to fully implement all the systems and test them.

## 4. Implementation

### 4.1. Approach 1: Counting with Yolo v7

YOLO v7 is a powerful and efficient algorithm for object detection that can be used for counting strawberries and other objects in images and videos. It is known for its high accuracy and ability to detect objects in images and videos with good precision and recall. YOLO v7 can also detect multiple objects simultaneously, which makes it a good choice for detecting and counting multiple strawberries in an image. The following steps were taken to implement the Yolo v7 approach.

- Create a virtual environment using the Anaconda prompt.
- Organize the images and their corresponding labels into train and validation sets.
- Install the required libraries to train the YOLO v7 model.
- Train the YOLO v7 model on the train set.
- Test the model on additional images and videos to see if it accurately counts the number of strawberries in each image or video.

YOLO v7 works by dividing an image into a grid of cells and using these cells to predict the presence and location of objects within the image. Each cell in the grid is responsible for predicting a set of bounding boxes around objects in its area, as well as the class of each object.

To detect strawberries, the YOLO v7 model is trained on a data-set of images that contain strawberries. During training, the model is presented with an image and must predict the location and class of each object in the image. The model makes these predictions using a combination of convolutional neural networks (CNNs) and fully connected layers.

The train.py script ran on an NVIDIA GeForce GTX 1080 using CUDA with the a batch size of 8 meaning that 8 training samples were used at a time during the training process, and for 100 epochs. An epoch is one complete pass through the entire data-set during training.

When the model is presented with a new image, it uses the CNNs and fully connected layers to extract features from the image and make predictions about the presence and location of objects in the image. The model then uses these predictions to draw bounding boxes around the detected objects and label them with their corresponding classes.

After conducting training on a data-set of labeled images of strawberries, a counting feature was implemented in order to determine the frequency of occurrence for each of the three classes of strawberries. This was achieved through manipulation of the detect.py file within the Yolo v7 project directory. The resulting class counts were subsequently annotated onto the original image of strawberries, along with the predicted bounding box and confidence score.

Figure 1. Output of Prediction


This implementation of counting enables the output of the number of strawberries classified as belonging to each individual class, as shown in figure 1. The decision to conduct counting in this manner was based on the belief that having the count of each individual class of strawberry would be more useful for applications of this technology than a total count of all strawberries in an image. This is due to the fact that strawberries classified as unripe may not reach maturity due to external factors such as predation or disease. Despite this, it is still valuable to have knowledge of the number of unripe or ripening strawberries in order to make predictions about future harvests.

The same technique could have been used to count the number of strawberries in an image irrespective of the class they belong to by simplifying the labels for the training data to just one class which represents a strawberry or by manipulating the detect.py file to only classify each strawberry as one class by choosing the class with the highest level of confidence.

### 4.2. Approach 1: Using Color

To identify whether or not each strawberry is ripe, unripe or partially ripe we can isolate the bounding box of the strawberry and build a mask that blocks out all the pixels not within the bounding box.

Once we have isolated the strawberry we can convert the image to HSV space. With the image in HSV space we threshold the image with a lower and upper bound for the red color. The reason we use HSV space is because it ignores the variances in luminance meaning we can more easily identify the red color of the strawberries (or lack thereof).

Once we have converted the image we can then count the number of red pixels, if the number of red pixels is above 50% for example then we can classify it as partially ripe and if it's above 75% we can classify it as ripe.

Instead of using 50 and 75 we can also sample through some of the training images and count the red pixels in those images which would then help give us a baseline for what percentage is required as a minimum.

### 4.3. Approach 2: Using Texture

The approach of using the texture of the strawberry to determine its ripeness is an interesting one, as it is based on the idea that the texture of a strawberry can be used as an indicator of its ripeness. The image is preprocessed by converting the image to grayscale and using a blur filter like Gaussian blur to reduce noise and make it easier to perform texture analysis.

Next, using a canny edge detector on the image, the edges of the strawberries are detected. The canny edge detector is a well-established method for detecting edges in images; it applies a series of convolutional filters sensitive to edges. The edges of the strawberry will appear as lines on the image, and these lines result from the intensity contrast between the strawberry and its background.

Once we have the edges, the next step is to count them; by counting the edges we can obtain an approximation of the density of the edges. The assumption is that unripe strawberries will have a higher density of edges since they are firmer and more pliable than ripe strawberries. If the density is high, it means that the strawberry is unripe; if it's low, it means that it's ripe.

### 4.4. Approach 3: Shape Analysis

The approach of using shape to determine the ripeness of strawberries is based on the idea that the shape of a strawberry can be used as an indicator of its ripeness. By converting the image to grayscale, the image is preprocessed to simplify the process of shape analysis.

The next step is to use Otsu's thresholding to segment the image into a binary image. Otsu's thresholding method seeks to find the threshold value that minimizes the variance between the two classes of pixels in the image. In this case, the pixels inside the strawberry and the pixels outside the strawberry.

After thresholding, the next step is to use contours to find the standard deviation and mean of the strawberry's shape. Contours are the boundaries of the segmented image; these boundaries are represented by a sequence of points that define the shape of the object in the image.By analyzing the contours, we can extract information such as the area, perimeter, and other shape-based features.

If the standard deviation of the shape features is high, there is a mix of ripe and unripe strawberries, which can be considered partially ripe. Once we have separated the partially ripe and unripe/ripe strawberries, we can use other shape-based features, such as the mean or the aspect ratio, to distinguish between ripe and unripe strawberries.

### 4.5. Other approaches

We also though about implementing some other methods for classifying the strawberries.

The first one would be with machine learning. This could be to train a machine learning model to classify strawberries as ripe, unripe, or partially ripe based on features extracted from the image. To achieve this we would use feature extraction, supervised learning and reducing the dimensionality of the image.

Another possible way would be similar to approach number 3, shape analysis. Here we would use morphological operations, to identify characteristics of the strawberries that are indicative of their ripeness. For example, ripe strawberries may have a more circular shape, while unripe strawberries may be more elongated or irregular in shape. This approach does have it's flaws though since there would be exceptions to this rule.

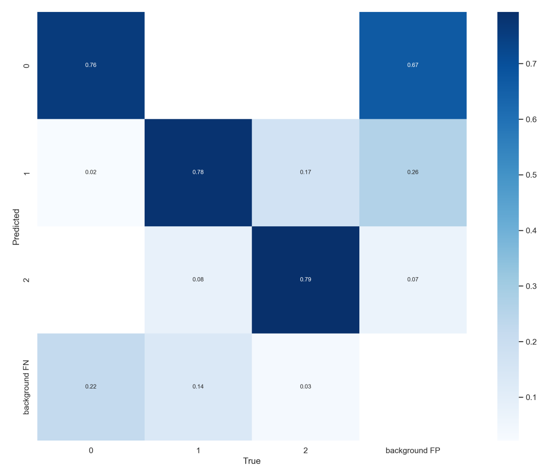
Yet another approach would be with image segmentation. Here we can separate the strawberries into individual objects and analyze the properties of each object to then classify it within the appropriate class. We can use clustering, and active contours. Active contours is useful here because it helps with all the strawberries that are partially occluded or hidden from sight.

The final method we could use is object recognition. Here we would use feature matching or deep learning, to identify and classify the strawberries in the image. This could be done by training a model on a data-set of labeled images of strawberries and then using the model to classify the strawberries in the image. For object recognition, we can use template matching or feature matching.

## 5. Results

We started off with training the model on a sample of the images. We then spent most of the time trying to figure out how to get the model working as well as how to successfully count the strawberries in each image. After running into several issues, (most notably Pytorch's incompatibility with AMD GPUs) we managed to implement some of our approaches successfully.

Figure 2. Yolov7 Confusion Matrix



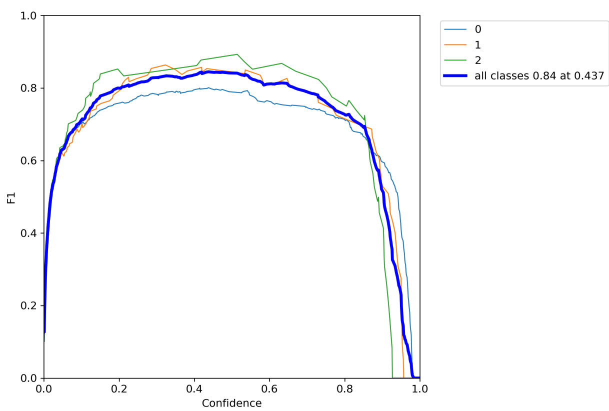

| Class | Labels | P | R | mAP@.5 | mAP@.5:.95 |
|---|---|---|---|---|---|
| all | 293 | 0.88 | 0.81 | 0.88 | 0.75 |
| 0 | 222 | 0.89 | 0.72 | 0.80 | 0.59 |
| 1 | 44 | 0.92 | 0.79 | 0.91 | 0.82 |
| 2 | 27 | 0.83 | 0.92 | 0.92 | 0.85 |

Overall, the YOLOv7 model appears to have a strong performance for classifying strawberries as unripe, ripening, or ripe. The model achieved high precision and recall scores for all three classes, with the highest precision and mAP scores being observed for class 1 (ripening strawberries).

IoU is a measure of the overlap between the predicted bounding box (or region of interest) and the ground-truth bounding box. A threshold of 0.5 indicates that the model's predicted bounding box must overlap with the ground-truth bounding box by at least 50 percent for the detection to be considered a true positive.

The mAP at an IoU threshold of 0.5 to 0.95 was slightly lower than the mAP at an IoU threshold of 0.5 for all classes, suggesting that the model may be slightly less effective at detecting strawberries when there is significant overlap between the predicted bounding box and the ground truth bounding box. However, the mAP scores at both IoU thresholds were still relatively high, indicating good overall performance.

Figure 3. Yolov7 F1 Curve




The YOLOv7 model has a high balance of precision and recall when the confidence threshold is low meaning the model is detecting a high number of true positive strawberries while maintaining a low number of false positives.

The steep drop in F1 score around confidence = 0.85 suggests that the model's performance is not as robust as it could be, as when the confidence threshold goes up, the performance drops. The model may be overfitting the data and not generalizing well causing this behaviour. A Possible solution would be to increase the sample size of the data-set however, This would also increase training times already at an hour and 20 minutes using the equipment available to us.

## 6. Limitations, Conclusions and Future Work

**Limited by the quality and quantity of the labeled data:** One limitation of our proposed systems is that they heavily rely on labeled data which isn't always available. Labeling the data takes lots of manual time. If the labeled data is not diverse or representative enough, the model may not be able to generalize well to new, unseen images of strawberries. Additionally, if the labels are not accurate or consistent, the model may learn the wrong patterns and produce inaccurate predictions.

**Edge Cases:** Another limitation is that our model may have flaws in it that don't allow it to recognize strawberries in different conditions. It could also make generalizations that don't apply to unseen data. Strawberry ripeness can be complex and affected by a variety of factors such as the type of strawberry, the growing conditions, and the stage of the fruit's development. A model that is trained on labeled images of strawberries may not be able to accurately predict ripeness in all cases, especially in cases where the strawberries are affected by unusual conditions or diseases.

**Can be computationally intensive:** Training a YOLOv7 model on a large data-set of labeled images of strawberries may be computationally expensive consuming a lot of power and time. This could make it expensive to implement and keep running continuously, that's why it's important that in future work the algorithm is flexible and fast.

**Scale Invariance:** Object detectors like YOLOv7 are not scale-invariant, so they may not perform well when presented with objects that are much smaller or larger than the objects they were trained on. This can be particularly challenging when counting small objects such as strawberries.

**Occlusion:** The model may struggle to detect objects that are partially occluded by other objects in the image. This can make it challenging to count partially hidden strawberries behind leaves or other objects.

The YOLOv7 model has strong performance for classifying strawberries as unripe, ripening, or ripe. The model achieved high precision and recall scores for all three classes, with the highest precision and mAP scores being observed for class 1 (ripening strawberries).

It is worth noting that the sample size of 49 images may not be sufficient to accurately assess the model's performance, and in hindsight it would be beneficial to evaluate the model on a larger data-set to confirm its effectiveness.

Future work can look at ways to make models more flexible as well as figuring out the best way to count and detect strawberries. some areas for future work include:

- Improving the model's ability to generalize: This could be done using techniques such as data augmentation, transfer learning, or active learning to make the model more robust to different variations of images.
- Improving scale invariance: This could be done using techniques such as multi-scale training or more context in the images, like using the surrounding pixels or objects as additional information.
- occlusion: This could be done by using techniques such as using more context in the images, like using the surrounding pixels or objects as additional information.

## References


[1] Josse De Baerdemaeker Andreas De Preter, Jan Anthonis. *Development of a Robot for Harvesting Strawberries*. IFAC PapersOnLine 51-17 (2018) 14–19, 2018. 2

[2] Zongmei Gao, Yuanyuan Shao, Yongxian Wang Guantao Xuan, Yi Liu, and Xiang Han. *Real-time hyperspectral imaging for the in-field estimation of strawberry ripeness with deep learning*. Artificial Intelligence in Agriculture 4 (2020) 31–38, 2020. 2

[3] Yuanyue Ge, Ya Xiong, Gabriel Lins Tenorio, and Pål Johan From. Fruit localization and environment perception for strawberry harvesting robots. *IEEE Access*, 7:147642–147652, 2019. 1

[4] Cao Qixin Guo Feng and Nagata Masateru. *Fruit Detachment and Classification Method for Strawberry Harvesting Robot*. March 2008 International Journal of Advanced Robotic Systems 5(1), 2008. 2

[5] Shigehiko HAYASHI, Satoshi YAMAMOTO, Sadafumi SAITO, Yoshiji OCHIAI, Junzo KAMATA, Mitsutaka KURITA, and Kazuhiro YAMAMOTO. *Field Operation of a Movable Strawberry-harvesting Robot using a Travel Platform*. JARQ 48 (3), 307 - 316 (2014) http://www.jircas.affrc.go.jp, 2014. 2

[6] Jamie Selby-Pham Kimber Wise, Trent Wedding. *Application of automated image colour analyses for the early-prediction of strawberry development and quality.* Scientia Horticulturae 304 (2022) 111316, 2022. 2

[7] P al J. Yuanyue Ge, Ya Xiong. *Instance Segmentation and Localization of Strawberries in Farm Conditions for Automatic Fruit Harvesting*. IFAC PapersOnLine 52-30 (2019) 294–299, 2019. 2

[8] Chu zhang, Chentong Guo, Fei Liu, Yong He Wenwen Kong, and Binggan Lou. *Hyperspectral imaging analysis for ripeness evaluation of strawberry with support vector machine*. Journal of Food Engineering 179 (2016) 11e18, 2016. 2

[9] Chengquan Zhou, Jun Hu, Zhifu Xu, Jibo Yue, Hongbao Ye, and Guijun Yang. *A Novel Greenhouse-Based System for the Detection and Plumpness Assessment of Strawberry Using an Improved Deep Learning Technique*. Front. Plant Sci. 11:559. doi:10.3389/fpls.2020.00559, 2020. 2


## 7. Appendix

Figure 4. Yolov7 test batch real labels

Figure 5. Yolov7 test batch predicted labels

Figure 6. Yolov7 R curve

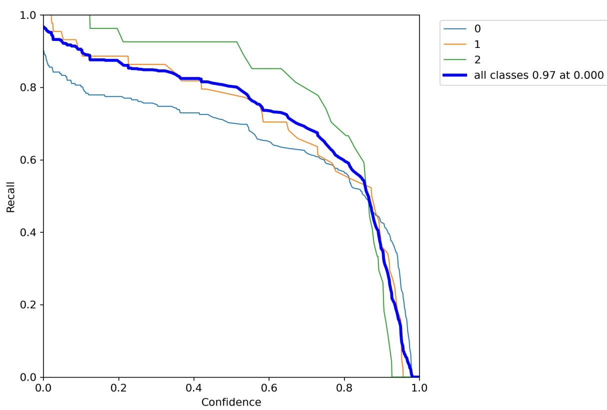



Figure 7. Yolov7 PR curve

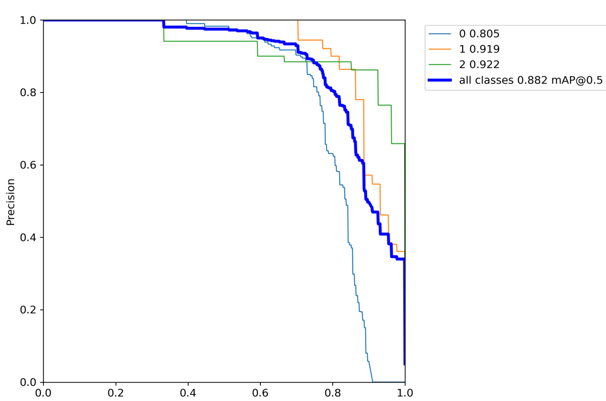



Figure 8. Yolov7 P curve