# Automated Strawberry Plucking - IIT Bombay

Vipul Ramtekkar, Dhananjay Tiwari, Prithvee Raj, Shreyas Dethe

December 2018

## 1 Abstract

In order to optimise harvesting process and reduce intensive labour work on strawberry fields, we have designed a strawberry harvesting bot. It is mounted on a moving platform. A combined harvesting and a moving platform demonstrates the potential of strawberry harvesting autonomously in a area. As a part of a larger project to build a device that is capable of plucking strawberries at a low cost suitable for Indian market, we have identified the ripeness of strawberries using *Yolo V3* and *Image Processing*.

## 2 Introduction

Strawberry is a high value fruit. In India, 85% of the strawberry production occurs in the Panchgani-Mahabaleshwar belt. The farms here are large in size and maintenance is a difficult task. According to data from our field survey in Mahabaleshwar, current strawberry production is highly dependent on human labour, out of which plucking takes upto 60% of labour cost. Also strawberry plucking is done for a major part of the year, with some farms harvesting for upto 6 months. In order to reduce production cost as well as manual labour, a need for automated solutions arises.

Plucking is the most intensive of all stages of strawberry production. Some of the main reasons for it are as follows.

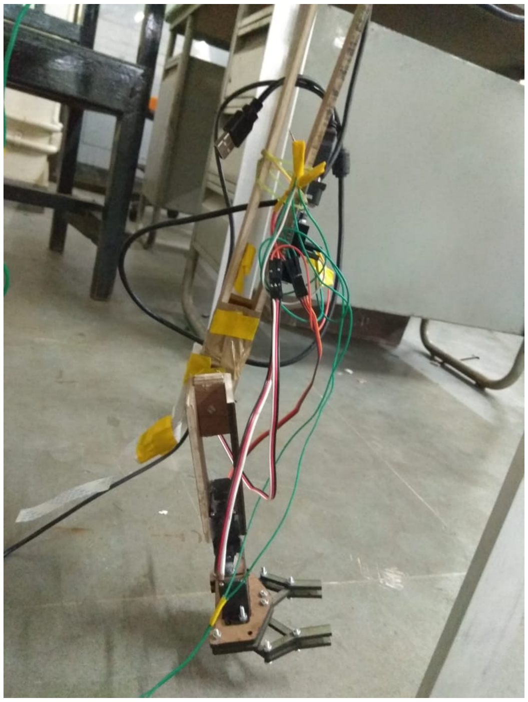
1. Strawberries from the same plant have uneven ripe-cycles and have large variations in colour and size at any given point of time. Hence they require multiple iterations for harvesting.
2. Strawberries are easily bruised, and require gentle handling.
3. Strawberry harvesting requires selective procedures depending on the type of demand, region of export, etc.

Prior to plucking strawberries, detecting which ones to pluck is a necessary step. The classification generally depends on the type of demand of strawberries and the area to which they are exported. For example, the strawberry produce in Panchgani, Maharashtra is often harvested a little early before full ripening if they are to be exported to, say, Kerala.

In this report, we describe our method for detecting strawberries of different ripeness levels. Compared with other approaches, we provide a way to detect the characteristics of a strawberry pertaining to whether it is ripe or unripe, healthy or diseased, etc. All these classifications are done using image processing paired with a yolo single shot detector which has been trained with thousands of images of different strawberries.

# 3 Mechanical characteristics of the prototype

Our prototype consists of a square metal plate placed over four metal rods. The side of the metal plate is 50 cm long while the height of the rods on which they stand are of 60 cm. These dimensions have been taken while considering the fact that the strawberry plants are short in height as well as the trying to minimise the overall size of the bot. We attached a disc to the square plate, the diameter of which is 40 cm. The disc is mounted using a motor and a shaft.



(a) Arm View

(b) Complete Design

For the plucking mechanism we have used a servo-based robotic arm. The arm consists of two joints, namely elbow-joint and wrist-joint. We have used servo motors for both the joints which are controlled by an Arduino. A counter-weight is attached diametrically opposite end of the arm for stability.

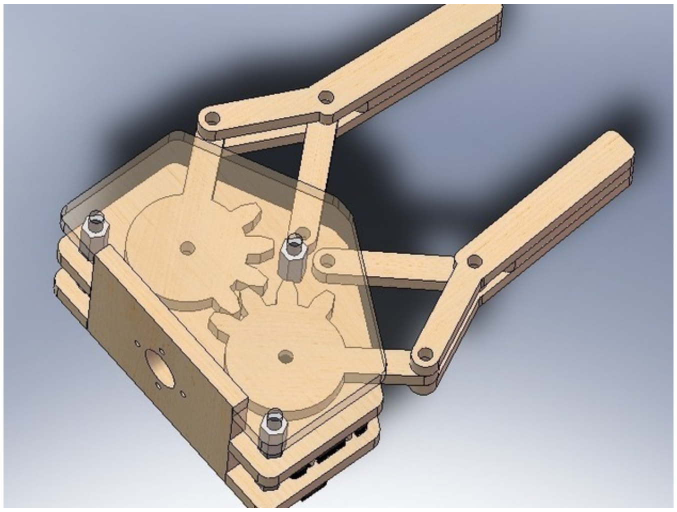


Figure 1: Gripper of our Prototype

# 4 Process flow

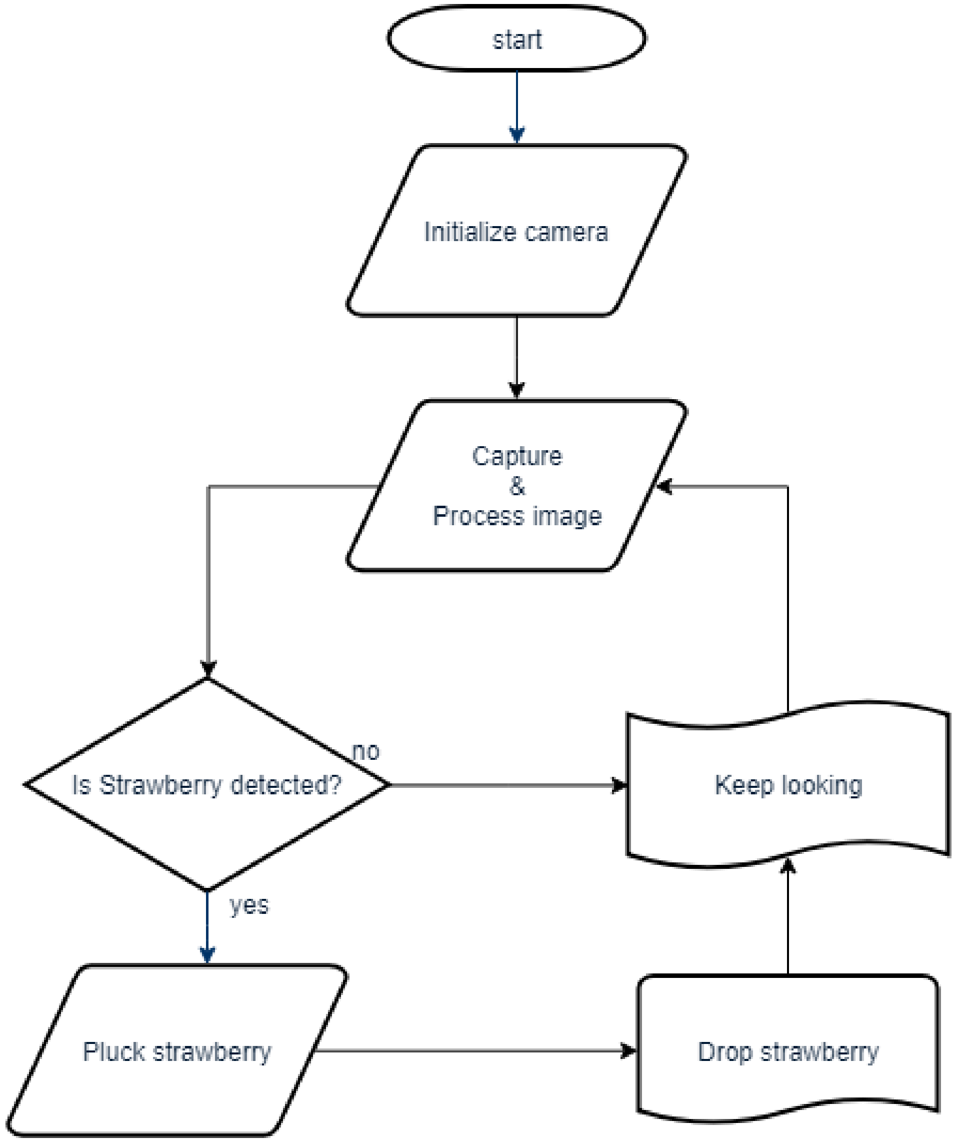



Figure 2: Flowchart

# 5 Software

## 5.1 Detection

**Understanding the problem**
The strawberry plucking process is tedious as the strawberries that are ripe need to be distinguished from the ones that are not. The colour varies in different lighting conditions and the strawberries are of varied shapes. Thus we train a deep classifier that learns whether an object is strawberry or not.

**Single Shot Detection vs Convolutional Neural Network**
As we input strawberry images from a vast dataset that contains them in multiple lighting conditions and shapes we get a high accuracy classifier. The problem with the standard R-CNN method is that it is painfully

slow and not a complete end-to-end object detector. While R-CNNs tend to very accurate, the biggest problem with the R-CNN family of networks is their speed — they were incredibly slow, obtaining only 5 FPS on a GPU.

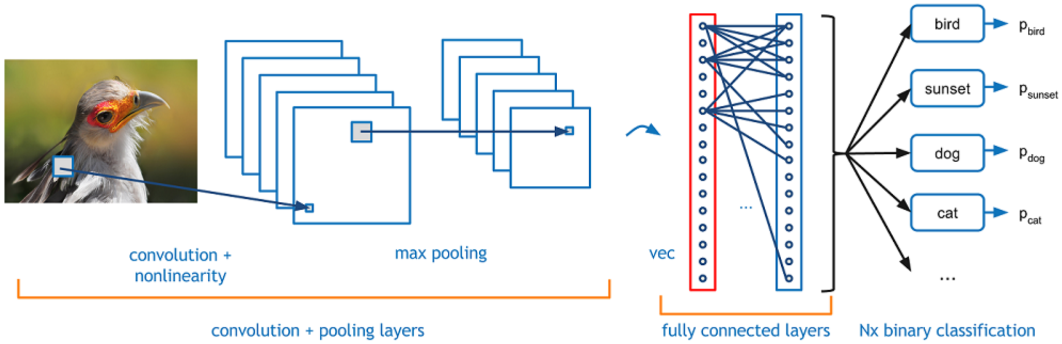



Figure 3: Example of working of a convolutional Neural Network

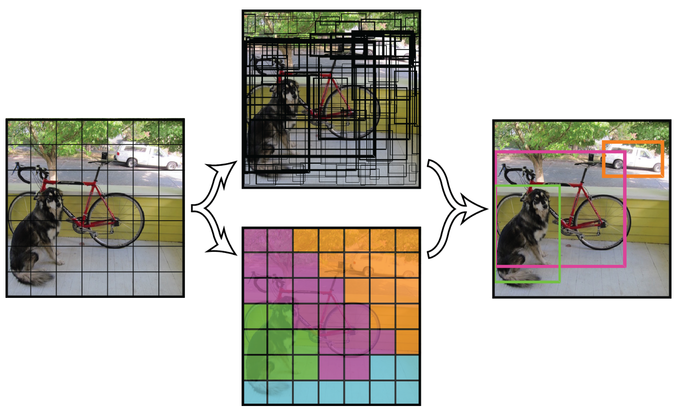
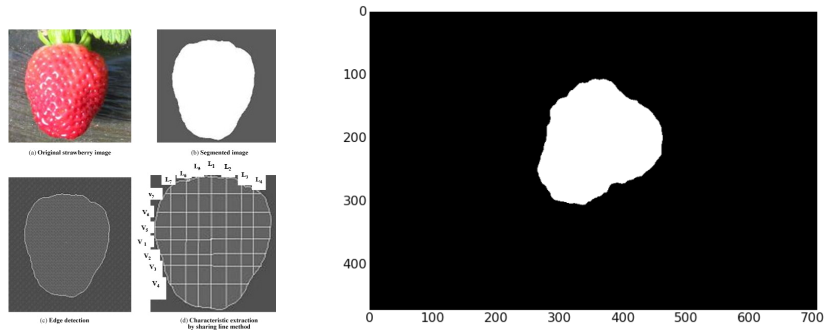


Figure 4: Working of yolo

**Yolo V3** is a single shot object detector and has been trained on multiple common objects, thus training a pre-existing classifier for a new object is a faster method and the second advantage is the point location which yolo provides. To help increase the speed of deep learning-based object detectors, both Single Shot Detectors (SSDs) and YOLO use a one-stage detector strategy. These algorithms treat object detection as a regression problem, taking a given input image and simultaneously learning bounding box coordinates and corresponding class label probabilities. In general, single-stage detectors tend to be less accurate than two-stage detectors but are significantly faster.

**Employing Computer Vision Methods** To improve the results we employ traditional CV Methods to improve the bounding box creation and thus employ the model after the application of the methods to improve accuracy.

Prepossessing the raw image: The pi camera captures an image, which is first re-sized to reduce the computations, then the colour space is changed and on which we further apply colour thresholding masks, the colour is one factor that makes the ripe ones stand apart from the unripe ones thus creating a mask based on the colour is the starting point. Finally when the contour detection is complete we employ various shape fitting techniques to complete the shape and thus identify the strawberry.
The various methods employed are:
**Contour Detection** The OpenCV provides a function that enables to find all the contours on the masked images and thus helps us to find all the available objects in the image. The contours are mapped according




(a) Edge Detection and Segmentation

(b) Masking

to the sizes and the maxed contour is found arranged according to the size. The Canny edge detector shows us all the edges and contours are the regions inside those edges. Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed. This shows us the edges of the images.

**Morphology** Erosion and dilation is done to remove and smoothen out all the small pores and incomplete holes. This helps to improve the overall quality for the detection and masking.

**Masking** A mask can be applied on an image of the same dimensions, but of any type. By applying a mask M on an image I, the pixels of I whose corresponding pixel in M are true are copied into a new image. The rest of the pixels in the new image are set to 0.

### 5.2 Communication

The serial communication between raspberry pi and arduino is achieved using pyserial. The raspberry pi communicates with two arduinos to enable the motion of stepper motor and the robotic arm. The arduino designated for the arm controls three servos for gripping and plucking action and the other for the stepper enables a high accuracy motion so that the entire plant is covered.

## 6 Electrical characteristics of the prototype

The device uses the following parts - a Raspberry Pi along with a camera which accounts for the image processing and detection. Also, we have used two Arduino Uno boards which help in the movement of the arm, multiple servo motors which serve as the 'joints' of the robotic arm, a stepper motor which rotates the auxiliary disc.

Following are some of the electrical characteristics of the parts.

- The stepper motor operates on 12 volts and consumes the most power. It requires 1A on average, in steps of 1 second.
- The Raspberry Pi is the second-most power-consuming part. However in our prototype, the board along with the camera and the Arduino Unos, during its peak operation, consumes just short of 1A of total current.
- The remaining servo motors during their operation consume about 500mA of current in short bursts of use.
- The maximum current observed during peak operation of the prototype was 1.45A, and the approximate average current drawn over some time is 1A.

We use a 12V Li-Po battery with capacity of 8000 mAh, thus providing a calculated approximate run-time of 8 hours. We further plan to use solar panels on the top of the prototype, which would offer to charge the on-board battery during field runs, ultimately extending the run-time of the device.

# 7 Financial Analysis

In a report by National Horticulture Board of India, average yield of strawberry is about 8 tonnes/acre under good management. The average sale rate is Rs.40,000 per tonne. Thus the gross return works out to Rs.3.20 lakhs per acre/annum. Considering the fact that a labour charges around 300 rupees for a day and plucking is done for 5-6 months a year, the total charge of a labourer turns out to be Rs. 45,000 to Rs. 54,000. Our prototype cost around Rs. 7,000 and assuming that a landlord hires at least two labourers for his one acre of cultivated land for harvesting, he could easily break even to the cost of our prototype in 12-14 days. Thus, he saves Rs. 40,000-45,000 per year for each labourer. Assuming the prototype to be working for 12hrs a day and it plucks 4 strawberries from each plant (during its harvesting season) and the prototype takes 100 sec to cover one plant, a simple mathematical analysis says that our prototype can pluck 1728 strawberries per day, which weighs 20kg (considering average weight of strawberry to be 12 gm ) which in turn give a return of 800 rupees per day. Thus, this prototype can be of great importance in the near future if deployed.

# 8 Conclusion

We have introduced a novel approach for detecting the picking point of strawberries and evaluated its effectiveness on a large number of images taken in the field. We present the design, analysis and the early experimental results supported with video and field report from Mahabaleshwar, which accounts for 85 % of total strawberry produced in India. This project is also very promising from business point of view, being economical and highly scalable to any Indian strawberry market as it cuts off labour charges drastically. There are several areas in which our approach can be improved. For now, like all existing approaches that we know of, cannot handle all the cases. There are many enhancement which could be implemented in bot like

- Instead of using a single gripper, multiple grippers can be used to increase number of strawberries harvested
- Instead of using a battery-powered-supply to make the bot move, we can use solar panels to power the bot
- Average picking time is approximately 20 secs for a strawberry, which could be reduced further by using better processors and more robust and precise motors

# 9 References


- Towards automated strawberry harvesting: Identifying the picking point by Zhuoling Huang et al.
- National Horticulture Board India Survey: http://nhb.gov.in/report_files/strawberry/STRAWBERRY.htm
- OpenCV: https://www.opencv.org/
- Yolo: https://pjreddie.com/media/files/papers/YOLOv3.pdf
- Pyserial: https://pythonhosted.org/pyserial/
- Python: https://www.python.org/
- Raspiberry Pi https://www.raspberrypi.org/documentation/
- Arduino https://www.arduino.cc/en/Main/Documentation
- Dataset: Horea Muresan, Mihai Oltean, Fruit recognition from images using deep learning, Acta Univ. Sapientiae, Informatica Vol. 10, Issue 1, pp. 26-42, 2018.