# $P^2Ag$: Perception Pipeline in Agriculture for Robotic Harvesting of Tomatoes

Soofiyan Atar[1][a], Simranjeet Singh[1][b], Jaison Jose[2][c] and Kavi Arya[1][d]
[1]*Indian Institute of Technology Bombay, Mumbai, India*
[2]*St. Vincent Pallotti College of Engineering and Technology, Nagpur, India*

Keywords: Robotic Harvesting, Semantic Map, Instance Segmentation, Image Classification.

Abstract: Harvesting tomatoes in agriculture is a time-consuming and repetitive task. Different techniques such as accurate detection, classification, and exact location of tomatoes must be utilized to automate harvesting tasks. This paper proposes a perception pipeline ($P^2Ag$) that can effectively harvest tomatoes using instance segmentation, classification, and semantic mapping techniques. $P^2Ag$ is highly optimized for embedded hardware in terms of performance, computational power and cost. It provides decision-making approaches for harvesting along with perception techniques, using a semantic map of the environment. This research offers an end-to-end perception solution for autonomous agricultural harvesting. To evaluate our approach, we designed a simulator environment with tomato plants and a stereo-vision sensor. This paper reports results on detecting tomatoes (actual and simulated ) and marking each tomato's location in 3D space. In addition, the evaluation shows that the proposed $P^2Ag$ outperforms the state-of-the-art implementations.

## 1 INTRODUCTION

During the last decade, the agriculture sector in India has experienced a sharp drop in the availability of labour despite the sector contributing significantly to the overall growth of the Indian economy. Even though India has the second-largest workforce globally, all sectors of the economy have been affected by the scarcity of labour, the impact being felt more in the agricultural sector (Prabakar et al., 2011; Deshingkar, 2003). The agricultural workforce reduced by 30.57 million, with numbers dropping from 259 million in 2004-05 to 228 million in 2011-12 (ICRISAT, 2020). Automation in agriculture (Ku, 2019) has many benefits such as labour efficiency, reduced environmental footprint, increase in production, and many more.

In recent years there have been many contributions related to computer vision (Tian et al., 2019), (TOMBE, 2020) for agricultural environments, including classification (Ma et al., 2020), detection (Zhao et al., 2019), analysis and monitoring (Lakhiar et al., 2018) of fruits. Fruits have different characteristics such as maturity, firmness, uniformity of size and shape, defects, skin colour and flesh colour. According to these factors, decision-making algorithms act on these factors for further progress/action. In agricultural environments (Dokic et al., 2020), computer vision algorithms are now commonly used for autonomous farming. In recent years, deep learning has achieved remarkable success in image classification, object tracking, and object detection algorithms. Moreover, Graphics Processing Units (GPUs) evolution has empowered complex deep neural networks and resource-efficient techniques. However, these advancements occur with higher power consumption and cost, which is not good for small sized Autonomous Mobile Agricultural Robots (AMAR). Our AMAR, as shown in Figure 1 has a small number of sensors with limited hardware capability and power capacity. Low computational algorithms are better suited by not compromising on accuracy and efficiency. The server-based approach (Bakar Siddik et al., 2018; Zamora-Izquierdo et al., 2019) is the most common application to avoid computational dilemmas. However, this approach is server-dependent, leading to many problems such as no response from the server, low latency and hardware failure. The perception pipeline is necessary,

[a] https://orcid.org/0000-0002-0878-9347
[b] https://orcid.org/0000-0002-8297-1470
[c] https://orcid.org/0000-0002-4132-1359
[d] https://orcid.org/0000-0002-7601-317X



Atar, S., Singh, S., Jose, J. and Arya, K.
P2Ag: Perception Pipeline in Agriculture for Robotic Harvesting of Tomatoes.
DOI: 10.5220/0010786100003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 3*, pages 121-129
ISBN: 978-989-758-547-0; ISSN: 2184-433X

which includes low computational complexity, data efficiency and server independence.

Figure 1: Custom AMAR housing few sensors with limited hardware capability.

A standard approach for segmenting each fruit is to transform the image into diverse regions with discriminative features and transfer to the complex deep neural network, detailing each region with a distinct label. Images in the agricultural environment centred on fruits generally lead to a broad range of intra-class fluctuations due to illumination, occlusion, clustering, camera viewpoint and seasonal maturity (translating to different fruit sizes, shapes, and colours). Former work on these challenges utilizes hand-engineered features to encode visual properties for classifying fruits (Payne et al., 2014). However, these algorithms are designed for a particular dataset by encoding features for specific fruit and conditions, making these approaches non-transferable to other crops/datasets. For the detection and delineation of each distinct fruit using masks, instance segmentation (Bolya et al., 2019; Hafiz and Bhat, 2020) approaches are used. Moreover, this approach increases accuracy and recall. It also reduces the detection of false-positive and false-negative. Object detection techniques using bounding box detection for detecting fruits increases complexity in occluded, and clustered conditions (Afonso et al., 2020). Another approach uses thermal images for detecting fruits using thermal cameras (Stajnko et al., 2004) which are costly as compared to RGB or depth cameras. Harvesting fruits after detection and classification is a significant issue without mapping the environment. Using the environment map, we can effectively plan the path of the robotic manipulator even when the agricultural environment is stochastic. It helps to identify hidden rotten/ripe fruits in harsh environments.

This paper proposes an efficient perception pipeline for tomatoes harvesting. To evaluate our perception pipeline, we created a simulated environment with tomato plants and tested $P^2Ag$. The major contribution of this paper is as follows:

- Implementation of instance segmentation of tomatoes with effective sensing
- Classification of rotten and ripe tomatoes
- Semantic mapping for efficient navigation

This paper focuses on the perception pipeline with optimized computational processing power and enhanced efficiency. The proposed pipeline is best suited for embedded applications with limited computational power and limited sensors.

The rest of the paper is structured as follows. Section 2 reviews the existing and related work on perception techniques in agriculture domain. Section 3 presents the proposed perception pipeline ($P^2Ag$). Finally, Section 4 provides the conclusion and outlook of this work in brief.

## 2 RELATED WORK

Perception techniques in the agricultural domain are an important area for research that eliminate errors in manual vision techniques. Previous research in this domain (Arakeri and Lakshmana, 2016) has shown better results than human vision precision. These computer vision techniques have been used for detecting and classifying different fruits (Unay et al., 2011; Al-Ohali, 2011; Nandi et al., 2013; Blasco et al., 2007). There is a positive correlation between ripening and rotting of tomatoes with the physical appearance of the tomatoes, such as colour, size (Baltazar et al., 2008). Research has also been done on detecting defects originating from different causes, where each defect is classified according to its appearance (Riquelme et al., 2008). For segmenting the fruit, instance segmentation algorithms are used. The state-of-the-art algorithm Mask R-CNN (He et al., 2018) segments each object with high precision, but the computation cost is too high for AMARs. Other instance segmentation implementations (Sun et al., 2021; Wu et al., 2019; Takikawa et al., 2019; Cheng et al., 2020) are more efficient or precise but do not simultaneously have these qualities. YOLACT (Bolya et al., 2019) is 394% more efficient than Mask R-CNN but 26% less accurate. Here the accuracy difference was lower compared to inference time and FPS, making YOLACT a better algorithm for our use

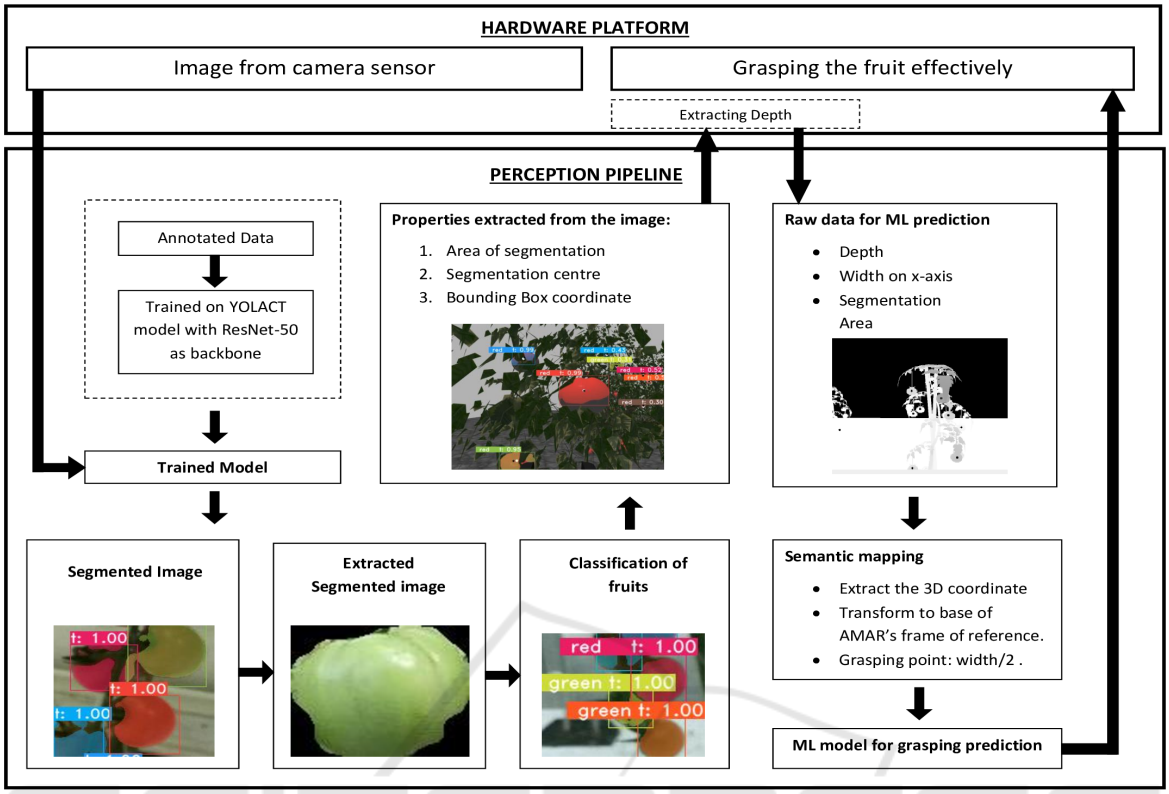


Figure 2: The flow of the system; showing segmentation of left and left-bottom side of pipeline classification, extraction of data in centre blocks, and finally, the depth and effective grasping of fruit on the right side.

case. Classifying these segmented objects within the multi-class has many implementations (Unay et al., 2011; Gehler and Nowozin, 2009; Cireşan et al., 2011; Zhang and Smart, 2004), but our application requires the most efficient algorithm without compromising accuracy.

The main challenge in any perception pipeline for an agricultural environment is to find an optimized and appropriate path for the robotic manipulator to harvest the crop. Many implementations use mapping techniques for observing plant characteristics (Xia et al., 2015; Lu et al., 2020; Sodhi et al., 2017). These methods usually use 3D cameras or a collection of 2D cameras with precise transformation. These methods are used for plant monitoring (Xia et al., 2015) using RGBD cameras, visualizing the health and other parameters of the plant. These methods increase the computation processing cost, which makes the whole approach inefficient. Reconstruction of a plant using structure from motion (SFM) (Lu et al., 2020) is another approach for mapping the plant. In this approach, multiple images are taken from different perspectives, which creates a 3D map of the plant after post-processing. This method requires post-processing not suitable for AMARs. Another implementation (Santos et al., 2015) uses 3D imaging systems, which are costly. 3D mapping of the whole plant using a single handheld camera leads to an efficient mapping of the plant, but these methods take much time and do not concentrate on essential plant factors. These methods do not focus on finding the best path for the robotic manipulator, excluding the occlusions. Other approaches include multi-view monitoring, which is impossible using AMAR as our robot cannot depict a multi-view system.

This research focuses on efficient techniques for segmenting each fruit, classifying and mapping the plant using segmented fruit areas. $P^2Ag$ reduces the complexity along with increasing the precision and accuracy. $P^2Ag$ focuses on practical strategies for semantically mapping the plant and predicting the best path, avoiding occlusions.

## 3 PERCEPTION PIPELINE

The perception for harvesting tomatoes in an agricultural platform requires a human-like approach towards nature, as the human species has mastered the art from time immemorial. Implementing these

diverse AMAR systems, a pipeline with perception computation and manipulative motions of the robotic arm is required, fulfilling the need for low computation and compelling predictions at the same time.

## 3.1 Proposed System

The perception pipeline can segment the tomato from a group in the given image. To explain and evaluate our proposed pipeline, we simulated tomato plants and tested them on natural tomato plants. The goal of the perception pipeline is to classify the tomato from the given image and locate them in 3D space for grasping. Figure 2 shows the flow of the system.

In the first step of the $P^2Ag$, a captured image from the camera sensor is given to the trained model for segmentation of the tomato as explained in section 3.2. The segmentation of the tomato is then extracted from the image and passed onto the classification model to predict the class of tomato. The information about each tomato is extracted, such as area, centroid and bounding box coordinates for further processing such as coordinate transformation explained further in section 3.4.3. After that, the depth to the segmentation centre is extracted from the platform. The pipeline continues where a Machine Learning (ML) model predicts the tomato to be harvested based on depth, bounding box size and segmentation area. This method of ML prediction makes its decision close to human behaviour. The higher predicted tomato after mapping of a plant is selected for grasping. The grasping coordinate is calculated by transforming the tomato to the base of AMAR and knowing the depth of the tomato, which is further explained in section 3.4.4 and then the process continues.

The flow of the system is depicted as a set of images in Figure 5, where the first row is the original image, the second row shows the image after the instance segmentation (Hafiz and Bhat, 2020) process, i.e. after YOLACT (Bolya et al., 2019) process the instance segmentation over the image, also the text overlay depicting the classification of tomatoes at the same time showing segmentation and bounding box centre of each tomato respectively, finally the third row shows the depth image and predictive analysis of the best tomato which can be picked using ML model with its predicted value and 3D coordinates (x, y, depth) concerning the camera frame. The process is repeated over the end-points and centre of a quarter-spherical map of a plant with respect to the centroid of the plant, which is shown by a set of images as (a), (b), (c), and (d) column as the left-most, centre, right-most, and top-most points respectively. The same coordinate of the final depth analysis image (Figure 5(d)) and the predicted value is passed for final picking manoeuvres.

## 3.2 Image Segmentation

Image Segmentation is the process by which a digital image is partitioned into subgroups (of pixels) called image objects, which can reduce the complexity of the image, and thus analyzing the image becomes a straightforward process. Since an image is a collection of different pixels, pixels with similar attributes are grouped using image segmentation. Image segmentation creates a pixel-wise mask for each object in the image. To differentiate objects of the same class, i.e.tomatoes, we used the instance segmentation method to segment a pixel into its parent class.

Results on a single Titan XP based on COCO dataset: Accurate Prediction (AP) shows that even though there is a bit drop in accuracy, YOLACT gives 3-4X Frames Per Second (FPS) than the Mask R-CNN (He et al., 2018) model. Even though Mask R-CNN has a classification score higher than YOLACT at the same time, the mask quality (Intersection of Union (IoU) b/w instance mask and ground truth) is low while comparing both. The AMAR system should focus on low computation methods, and here the YOLACT manifests its segmentation process positively. The comparison of the same is shown in Figure 3 where Figure 3 (a) is the actual test image, Figure 3 (b) is the YOLACT model processed image and Figure 3 (c) is the Mask R-CNN processed image.

Table 1: Mask Performance of YOLACT and Mask R-CNN (state-of-the-art method) for mask mAP and speed on COCO test-dev.

| Model Name | Backbone | Time | FPS | mAP |
|---|---|---|---|---|
| Mask R-CNN | ResNet101-FPN | 116.3 | 8.6 | **35.7** |
| YOLACT | ResNet101-FPN | 30.3 | 33.0 | 29.8 |
| YOLACT | ResNet50-FPN | **23.5** | **42.5** | 28.2 |

*on **Titan Xp** (GPU Geekbench link).

Table 2: Mask Performance of YOLACT for different embedded platform on custom tomatoes dataset.

| Embedded Device | Backbone | FPS |
|---|---|---|
| **Nvidia Jetson TX2** | ResNet50-FPN | 2.3 |
| Nvidia Xavier AGX | ResNet50-FPN | 8.4 |
| Nvidia GeForce GTX 1660Ti (mobile) | ResNet50-FPN | **22** |
| Nvidia GeForce GTX 1660Ti (mobile) | ResNet101-FPN | 16 |

### 3.2.1 Instance Segmentation

Instance segmentation classifies every object separately, even if they belong to the same class of objects. Different backbones for YOLACT have been tested, which worked as the base architecture of

(a) (b) (c)

Figure 3: (a) Actual Image, (b) YOLACT segmentation, and (c) Mask R-CNN segmentation.

$P^2Ag$. The backbone comparison, as shown in Table 1, states that although the mean Accurate Prediction (mAP) increases with more complex Residual networks, at the same time, there is a decrease in FPS values and increase in inference time. There is a decrease in inference time of computation by 283% for YOLACT (ResNet101-FPN) while comparing with Mask R-CNN having a backbone of ResNet101-FPN (He et al., 2016), which resulted in an increase of FPS in YOLACT by 283% while sacrificing the mAP value reduction by 20% for the same. The comparison of ResNet101-FPN and ResNet50-FPN for YOLACT shows a minor decrease in mAP values by 5% for ResNet-50-FPN while making the computation faster by a significant increase of FPS by 28% and decrease in inference time by 29% as shown in Table 1. Comparison of performance of YOLACT is shown in Table 2. YOLACT is very feasible for Jetson TX2 without compromising accuracy, being very low-end hardware for AMAR. 2.3 FPS, which is obtained from Nvidia Jetson TX2, is feasible for this perception pipeline. A significant decrease in FPS for ResNet-


Figure 4: Left-(actual image) right-(after classification) using LR ML model.

50-FPN can be observed compared with other backbones; therefore, ResNet50-FPN has been used to implement YOLACT, as computation factors also come into the role of implementation AMAR systems. The complexity of the models also has to be taken into account with FPS and mean accuracy. In contrast, the training period for later diverse sets of tomatoes has been reduced since the ResNet50-FPN backbone is trained via transfer learning methods (Torrey and Shavlik, 2009).

### 3.3 Tomato Classification

The classification of tomatoes is the deciding factor for reaching out for the fruits, as the basic segregation process. We compared three methods for classifying tomatoes as ripe or unripe and rotten. The basic ML models we used are: Logistic regression (LR) (Hruaia et al., 2017), Support Vector Machine (SVM) (Sun et al., 2015) and k-Nearest Neighbour (kNN) (Amato and Falchi, 2010). Comparison studies (Abd Rahman et al., 2015) state that LR is more stable in its prediction for binary classification, having less inference time for the classification process. The actual classification image result can be seen in Figure 4 left and right image, respectively. The model has been trained on datasets created by segmentation results from the 3.2.1, after reducing the size of the image and giving the image as input parameters to the model and label of images as the output. The LR generates a model with a file size of 100 KiloBytes or less, proving its capability to implement an efficient data process in the AMAR proposed pipeline system.

### 3.4 Effective Tomatoes Localization

The AMAR aims to harvest the tomato, which should be within its radius of reach, but at the same time, it should be in the class which needs to be harvested out from the respective plant after the manoeuvre. The method requires a practical solution that may result in several iterations and interactions to actual physics of the nature for every step improvement; the same can be achieved using Gazebo. The system, therefore, has been simulated and tested with ROS (Quigley et al., 2009) inside Gazebo. For these iterations and testing of our pipeline, we have used a simulated environ-

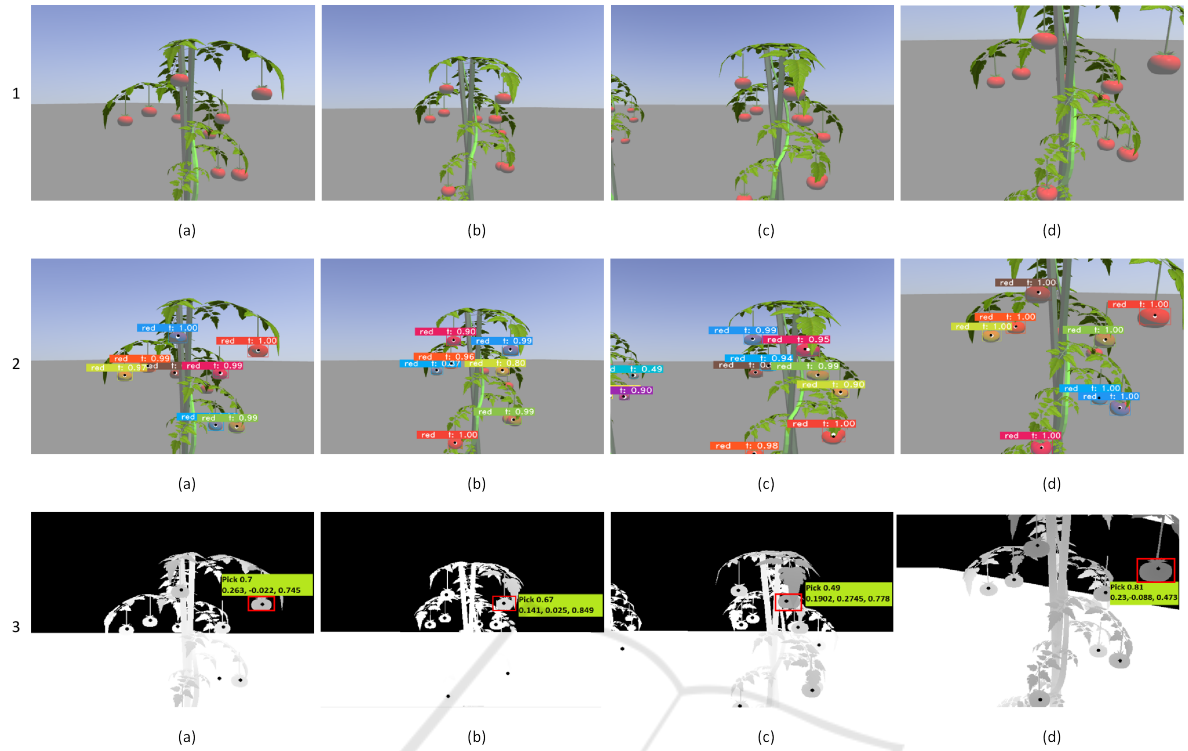


Figure 5: Semantic mapping and processing, (a) left, (b) centre, (c) right and (d) top view of plant; (1) original image, (2) Segmented image and processed image with classification, and (3) depth image with 3D coordinate and ML prediction.

ment using the Noetic version of ROS. As the process of implementation and coordination remains the same, i.e. the ROS system coordinates the actuation of hardware in response to perception; therefore, simulation of the same will not produce a high difference in practical implementations.

### 3.4.1 Generating Area and Centroid

The area of the segmented portion too adds an essential parameter for a decision to pick the tomato, even as the human judgment would make it primary. The method uses basic mathematical calculation for ease of computation. Therefore, the segmented portion of each tomato from the YOLACT is extracted by mapping the segmented portion over a null valued pixel image. After that, the mask is extracted with a threshold of the maximum pixel value for extracting a blob. The blob generation develops a more smooth way of processing the extraction of contours (Antunes and Lopes, 2013) and ensures the (Xu and Li, 2008; Davis and Raianu, 2007) uniform calculation of moments for each pixel. The contours are generated for which the blob can be utilized for computing the foreground area of tomato which is calculated by using moments of pixel which is given by $m_{ji} = \sum_{xy}(array(x,y).x_j.y_i)$. Where $m_{00}$ represents the area, $x_j$ and $y_i$ represents pixel values. This method proves to be an effective, fast (Yang and Albregtsen, 1995) and low computation process for the AMAR systems. The moment on the x-axis is given by $m_{10}$ and similarly for y-axis by $m_{01}$. The centre of contour in x-axis is obtained by the ratio of moments in x-axis by area (or moment of the blob), $x = m_{10}/m_{00}$ similarly for y-axis it is obtained by $y = m_{01}/m_{00}$. These $x$ and $y$ coordinates signify the segmentation centre.

### 3.4.2 Depth Information Extraction

The 3D coordinate of a point, i.e. segmentation centre of the tomato, is given by x, y, and z, extracted using a depth image from the depth camera embedded in the AMAR. The depth image is represented by [x, y, depth]. By knowing the $x$ and $y$ from the segmentation centre, we can retrieve the depth of that point using depth image. Since the values, $x$ and $y$ coordinate of the point and depth are present, we can calculate the 3D coordinate of that point on tomato in space with respect to the AMARs camera.

$$x_w = \frac{depth * (x_p - C_x)}{f_x}, y_w = \frac{depth * (y_p - C_y)}{f_y} \qquad (1)$$
$$z_w = depth$$

Equation 1 calculates $x_w$, $y_w$ and $z_w$, which are the 3D coordinate of segmentation centre of the tomato with respect to AMARs camera, where $x_p$ and $y_p$ is the x and y pixel coordinate of that point, $C_x$ and $C_y$ is the centre pixel and $f_x$ and $f_y$ resembles the focal length in x and y axes of a camera respectively.

#### 3.4.3 Semantic Mapping

The mapping of the plant is the next major part of the decision as it contributes to the harvest of the tomato from the most suitable position and judges other parameters of the tomato at the same time while also making the processes in a low computational manner. The semantic mapping of a plant requires manoeuvring of AMAR; for efficient and straightforward mapping of the plant, we implemented the quarter-spherical movement around the plant, which resembles the basic human behaviour for plant mapping.

After the depth information extraction, the end-points of the bounding box in the x-axis with respect to the y-axis from the segmented centroid is converted (or transformed) to a 3D space coordinate. The generated coordinate represent the inertial frame of reference from the camera to the point of grasping tomato, which is meaningless until the coordinate system is transformed to the base of AMAR. Therefore the affine transformation (Švec and Farkas, 2014) of the relative frames are done by using rotational matrix and translation vector calculations.

To compute the grasping manoeuvre of tomato, AMAR needs to map the tomato plant. To do so, the arm of the AMAR should move in a quarter-spherical motion around the plant's centroid of its 3D structure. The robotic manipulator is made to compute in pre-defined positions to achieve the quarter-spherical structure, mapping the plant effectively. From the quarter-spherical motion, the four computation points for the pipeline are the leftmost, rightmost, centre (facing perpendicularly to plant), and topmost points. These points make the processing easy and low computation by iterating only four times, instead of continuous iteration of the perception pipeline for the whole manoeuvre period, resulting in a similar outcome for prediction but with an increase in computation. Figure 5 shows the order of the set of images as first and third row as original and depth process with (a), (b), (c) and (d) columns as left, centre, right, and top end-point respectively as explained in section 3.1. This method will help extract an influential picking position for a tomato, i.e. by executing this manoeuvre mentioned above, human exertions for finding the best position to pick the tomato can be achieved.

The process of pipeline inter-process like segmentation, depth extraction, and details like area and centroid will be stored for the following computation at every four-point of the quarter-spherical motion. The computation of such a simple and efficient method also fills the low-computation need of processing in AMAR. The parameters followed by mapping the plant for the tomatoes are depth, width on the x-axis by the base of AMAR's frame of reference transformed and segmentation area. The judgment for the harvesting of selective tomatoes should result from parameter analysis of information extracted from the mapping of each tomato.

#### 3.4.4 Identification of Efficient Pose

The model should be a state-of-the-art method that accomplishes the need for faster and low computation AMAR systems. The LR ML model trained on the parameters resulted in an accuracy of 99% for the tomato to be grasped. The probabilities and also the choice of model is from the study of comparison with SVM, kNN proving its stable algorithm and its low computation capability, similarly as explained in section 3.3. The trained model, when executed with the parameters, as shown in Figure 5 the third row, by mapping the content onto the depth image, displays the probability factor of the tomato, which can be picked with its respective 3D coordinate (Mussabayev et al., 2018) in AMAR base's frame of reference at each end-point. The left end-point depth image is Figure 5 (a) column, and similarly, all the other end-points as labelled are depicted in Figure 5's (b), (c) and (d) columns. The highest probability of 'pick' class is labelled next to the tomato by the program with its 3D coordinate from the frame of reference of AMAR's camera, in the format returned by the calculation done using 3D extraction of tomato.

The best tomato to pick is determined by taking the highest probability from the picked class as predicted by the ML algorithm. The 3D coordinate of that tomato is transformed with respect to the AMAR's base. Then the robotic manipulator plans its path to the 3D pose. The grasping position of the gripper from the calculated surface 3D coordinate is calculated by making the robotic manipulator move towards the tomato by $(y\ or\ x\ value) * factor + (w_f/2)$ where $w_f$ is the width of tomato, which is calculated by transforming the edges of bounding box in x-axis having y-axis as segmentation centre line, where 'y' is the position of the coordinate point pointing towards the tomato and the factor is generated by the difference in angle of orientation of AMAR and the tomato. The factor for the x-axis can be shown by $factor_x = sin(angle)$ while factors of y-axis by $factor_y = cos(angle)$ due to difference in tomato and AMAR orientation, where z is valued as the axis nor-

mal to the ground (base), after the tomato is picked up successfully the process iterates and goes on, where it also avoids picking if no tomato is found in the 'pick' class.

# 4 CONCLUSION

We have introduced a new, highly optimized perception pipeline whch can be used for harvesting fruits in agricultural environments. All perception techniques, that comprise most computations for robotic harvesting, including instance segmentation of fruit, semantic mapping and image classification, are realized for onboard computation, that is the requirement for autonomy and eliminating server dependencies.

We demonstrated a balance of efficient and precise instance segmentation algorithms. YOLACT provided an mAP of 28.2, indicating that precision was high and can be increased using a more computational heavy backbone for these algorithms. Image classification using LR resulted in a more efficient classification algorithm for this pipeline for classifying fruit with different parameters such as rotten, ripe and many more. We also demonstrated the outcome of semantic mapping, which runs efficiently and performs better than directly picking the fruit.

In future work, we will optimize these algorithms and implement them for diverse categories of fruits and vegetables. In addition, we will also implement this pipeline in a more dynamic and cluttered environment using a small-sized quadcopter with a single camera that can enable more efficiency.

# ACKNOWLEDGEMENTS

This work was supported in part by Ministry of Education (MoE), Govt. of India in the project e-Yantra (RD/0113-MHRD001-021). We acknowledge the support of e-Yantra staff especially Rathin Biswas for reviewing the paper.

# REFERENCES

Abd Rahman, H. A., He, H., and Bulgiba, A. (2015). Comparisons of adaboost, knn, svm and logistic regression in classification of imbalanced dataset. pages 54–64.

Afonso, M., Fonteijn, H., Fiorentin, F. S., Lensink, D., Mooij, M., Faber, N., Polder, G., and Wehrens, R. (2020). Tomato fruit detection and counting in greenhouses using deep learning. *Frontiers in Plant Science*, 11:1759.

Al-Ohali, Y. (2011). Computer vision based date fruit grading system: Design and implementation. *Journal of King Saud University - Computer and Information Sciences*, 23.

Amato, G. and Falchi, F. (2010). Knn based image classification relying on local feature similarity. pages 101–108.

Antunes, M. and Lopes, L. S. (2013). Contour-based object extraction and clutter removal for semantic vision. In Kamel, M. and Campilho, A., editors, *Image Analysis and Recognition*, pages 170–180, Berlin, Heidelberg. Springer Berlin Heidelberg.

Arakeri, M. and Lakshmana (2016). Computer vision based fruit grading system for quality evaluation of tomato in agriculture industry. *Procedia Computer Science*, 79:426–433.

Bakar Siddik, M. A., Deb, M., Pinki, P. D., Kanti dhar, M., and Faruk, M. O. (2018). Modern agricultural farming based on robotics and server-synced automation system. In *2018 International Conference on Recent Innovations in Electrical, Electronics Communication Engineering (ICRIEECE)*, pages 323–326.

Baltazar, A., Aranda, J. I., and González-Aguilar, G. (2008). Bayesian classification of ripening stages of tomato fruit using acoustic impact and colorimeter sensor data. *Computers and Electronics in Agriculture*, 60(2):113–121.

Blasco, J., Aleixos, N., and Moltó, E. (2007). Computer vision detection of peel defects in citrus by means of a region oriented segmentation algorithm. *Journal of Food Engineering*, 81(3):535–543.

Bolya, D., Zhou, C., Xiao, F., and Lee, Y. J. (2019). Yolact: Real-time instance segmentation.

Cheng, B., Collins, M. D., Zhu, Y., Liu, T., Huang, T. S., Adam, H., and Chen, L.-C. (2020). Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation.

Cireşan, D. C., Meier, U., Masci, J., Gambardella, L. M., and Schmidhuber, J. (2011). High-performance neural networks for visual object classification.

Davis, P. and Raianu, S. (2007). Computing areas using green's theorem and a software planimeter. *Teaching Mathematics and Its Applications*, 26:103–108.

Deshingkar, P. (2003). Seasonal migration for livelihoods in india: Coping, accumulation and exclusion.

Dokic, K., Blaskovic, L., and Mandusic, D. (2020). From machine learning to deep learning in agriculture – the quantitative review of trends. *IOP Conference Series: Earth and Environmental Science*, 614:012138.

Gehler, P. and Nowozin, S. (2009). On feature combination for multiclass object classification. In *2009 IEEE 12th International Conference on Computer Vision*, pages 221–228.

Hafiz, A. M. and Bhat, G. M. (2020). A survey on instance segmentation: state of the art. *International Journal of Multimedia Information Retrieval*, 9:171 – 189.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2018). Mask r-cnn.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Con-*

*ference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Hruaia, V., Kirani, Y., and Singh, N. (2017). Binary face image recognition using logistic regression and neural network. pages 3883–3888.

ICRISAT (2020). International crop research institute for the semi-arid tropics , labor scarcity and rising wages in indian agriculture. https://www.icrisat.org/labor-scarcity-and-rising-wages-in-indian-agriculture/.

Ku, L. (2019). How automation is transforming the farming industry. https://www.plugandplaytechcenter.com/resources/how-automation-transforming-farming-industry/.

Lakhiar, I., Jianmin, G., Syed, T., Chandio, F. A., Buttar, N., and Qureshi, W. (2018). Monitoring and control systems in agriculture using intelligent sensor techniques: A review of the aeroponic system. *Journal of Sensors*, 2018:18.

Lu, X., Ono, E., lu, S., Zhang, Y., Teng, P., Aono, M., Shimizu, Y., Hosoi, F., and Omasa, K. (2020). Reconstruction method and optimum range of camera-shooting angle for 3d plant modeling using a multi-camera photography system. *Plant Methods*, 16.

Ma, C., Xu, S., Yi, X., Li, L., and Yu, C. (2020). Research on image classification method based on dcnn. In *2020, ICCEA*, pages 873–876.

Mussabayev, R., Kalimoldayev, M., Amirgaliyev, Y., Tairova, A., and Mussabayev, T. (2018). Calculation of 3d coordinates of a point on the basis of a stereoscopic system. *Open Engineering*, 8(1):109–117.

Nandi, C., Tudu, B., and Koley, C. (2013). Machine vision based techniques for automatic mango fruit sorting and grading based on maturity level and size. *Sensing Technology: Current Status and Future Trends II*, 8:27–46.

Payne, A., Walsh, K., Subedi, P., and Jarvis, D. (2014). Estimating mango crop yield using image analysis using fruit at 'stone hardening' stage and night time imaging. *Computers and Electronics in Agriculture*, 100:160–167.

Prabakar, C., Devi, K., and Selvam, S. (2011). Labour scarcity â€“ its immensity and impact on agriculture. *Agricultural Economics Research Review*.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. (2009). Ros: an open-source robot operating system. volume 3.

Riquelme, M., Barreiro, P., Ruiz-Altisent, M., and Valero, C. (2008). Olive classification according to external damage using image analysis. *Journal of Food Engineering*, 87:371–379.

Santos, T. T., Koenigkan, L. V., Barbedo, J. G. A., and Rodrigues, G. C. (2015). 3d plant modeling: Localization, mapping and segmentation for plant phenotyping using a single hand-held camera. In Agapito, L., Bronstein, M. M., and Rother, C., editors, *Computer Vision - ECCV 2014 Workshops*, pages 247–263, Cham. Springer International Publishing.

Sodhi, P., Vijayarangan, S., and Wettergreen, D. (2017). In-field segmentation and identification of plant structures using 3d imaging. In *2017 IEEE/RSJ (IROS)*, pages 5180–5187.

Stajnko, D., Lakota, M., and Hočevar, M. (2004). Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging. *Computers and Electronics in Agriculture*, 42(1):31–42.

Sun, X., Liu, L., Wang, H., Song, W., and Lu, J. (2015). Image classification via support vector machine. In *2015 4th ICCSNT*, volume 01, pages 485–489.

Sun, Y., Gao, W., Pan, S., Zhao, T., and Peng, Y. (2021). An efficient module for instance segmentation based on multi-level features and attention mechanisms. *Applied Sciences*, 11(3).

Takikawa, T., Acuna, D., Jampani, V., and Fidler, S. (2019). Gated-scnn: Gated shape cnns for semantic segmentation.

Tian, H., Wang, T., Liu, Y., Qiao, X., and Li, Y. (2019). Computer vision technology in agricultural automation ——a review. *Information Processing in Agriculture*, 7.

TOMBE, R. (2020). Computer vision for smart farming and sustainable agriculture. In *2020 IST-Africa Conference (IST-Africa)*, pages 1–8.

Torrey, L. and Shavlik, J. (2009). Transfer learning. *Handbook of Research on Machine Learning Applications*.

Unay, D., Gosselin, B., Kleynen, O., Leemans, V., Destain, M.-F., and Debeir, O. (2011). Automatic grading of bi-colored apples by multispectral machine vision. *Computers and Electronics in Agriculture - COMPUT ELECTRON AGRIC*, 75:204–212.

Švec, M. and Farkas, I. (2014). Calculation of object position in various reference frames with a robotic simulator.

Wu, H., Zhang, J., Huang, K., Liang, K., and Yu, Y. (2019). Fastfcn: Rethinking dilated convolution in the backbone for semantic segmentation.

Xia, C., Wang, L., Chung, B.-K., and Lee, J.-M. (2015). In situ 3d segmentation of individual plant leaves using a rgb-d camera for agricultural automation. *Sensors*, 15(8):20463–20479.

Xu, D. and Li, H. (2008). Geometric moment invariants. *Pattern Recognit.*, 41:240–249.

Yang, L. and Albregtsen, F. (1995). Fast and exact computation of moments using discrete green's theorem.

Zamora-Izquierdo, M., Santa, J., Martinez, J., Martínez, V., and Skarmeta, A. (2019). Smart farming iot platform based on edge and cloud computing. *Biosystems Engineering*, 2019:4–17.

Zhang, M. and Smart, W. (2004). Multiclass object classification using genetic programming. In Raidl, G. R., Cagnoni, S., Branke, J., Corne, D. W., Drechsler, R., Jin, Y., Johnson, C. G., Machado, P., Marchiori, E., Rothlauf, F., Smith, G. D., and Squillero, G., editors, *Applications of Evolutionary Computing*, pages 369–378, Berlin, Heidelberg. Springer Berlin Heidelberg.

Zhao, Z.-Q., Zheng, P., Xu, S.-T., and Wu, X. (2019). Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3212–3232.