GrapeRob:
A Grape Localization Pipeline for Automated Robotic Harvesting

Advaith Balaji
UM Robotics
Univeristy of Michigan
Ann Arbor, USA
advaithb@umich.edu

Isaac Madhavaram
UM Robotics
University of Michigan
Ann Arbor, USA
imadhav@umich.edu

***Abstract*—Grapes are a globally significant fruit utilized not only for consumption but also for wine production. However, manual grape harvesting faces challenges due to labor shortages and technical intricacies. This paper proposes a robotic grape harvesting system using deep learning based robot perception and 3D reconstruction to compute grape bunch and stem poses for robotic picking. Drawing from previous research in agricultural robotics, this system extends existing methodologies by integrating a novel stem segmentation model into the vision pipeline, and testing on a robotic platform to gauge accuracy. Experiments on the Fetch mobile manipulation shows a grasp success rate of 85.71% and an average gripper pose error of 4 cm on less than 150 training images. Authors can be contacted via email regarding questions, clarifications, and data availability. The project page is available at: https://github.com/adi-balaji/grape_juice.**

I. Introduction

Grapes are a widely popular fruit worldwide, with 72 million tons grown each year. Many countries use grapes for wine production, such as the United States, France, Italy, Spain, and more. In addition, grapes have been associated with the prevention of cancer, heart disease, high blood pressure, and much more. However, it has been increasingly harder to harvest due to the technical pruning process and high labor requirements. In areas with labor shortages, due to the aging population and reduced agricultural labor force for example, there have been many many inefficiencies in the process of manual harvesting.

Thus, the purpose of this paper is to propose a system that supports the automated harvesting of grapes with robotic manipulators. Such a system has the potential to alleviate labor shortages and ensure sustainable grape production even in areas with insufficient availability of workers. Using deep learning, we propose an end to end grape localization system to compute bunch and stem poses for robotic manipulation. With labeled data of other crops, this pipeline could be generalized to any other harvesting tasks.

II. Related Work

Computer vision and deep learning for agriculture has been a growing topic over the past few years for its potential to greatly increase efficiency to increase crop yields in a time of labor shortage and high demand for food. For the task of fruit detection, previous papers have focused on addressing the cluttered and occluded nature of fruits in farms. *Santos et al.* employs a grape dataset to create a segmentation model for grape masking for [1]. The WGISD dataset proposed in this paper contains a diverse representation of grape bunches in a vineyard environment making it ideal for training a real-world applicable system. Additionally, for robotic pickling, the system needs good centroid estimates, which requires fruits to be localized by the system with high accuracy. *Gao et al.* proposed a kiwi localization system with YOLOv5 and binocular imagery which achieved millimeter localization precision [2]. The use of binocular imagery was similarly used in *Yin et al.*, which presents an approach for the detection and pose estimation of grapes using binocular imagery and deep learning [3]. As such, the main contribution lies in the translational and rotational pose estimation of these grape bunches – our system focuses on the ideas of *Yin et al.* to improve the capabilities of a grape detection system and evaluate its performance in a real world scenario.

A. Paper Review

The core problem addressed is the need for efficient fruit detection and accurate pose estimation for the furtherment of agricultural robotics. The key idea involves utilizing Mask R-CNN for fruit segmentation and point cloud extraction for pose estimation, and the implementation involves image preprocessing, fruit segmentation, point cloud construction, and pose estimation using a RANSAC cylinder model fitting approach. The evaluation methodology includes metrics such as precision, recall, and intersection over the union (IoU) to evaluate the performance under different lighting conditions. Finally, the paper concludes that the proposed approach achieves high precision and recall rates, an average IoU of about 82%, and a quick inference time of 1.7 seconds, demonstrating its viability for real-world application in grape harvesting robotics.

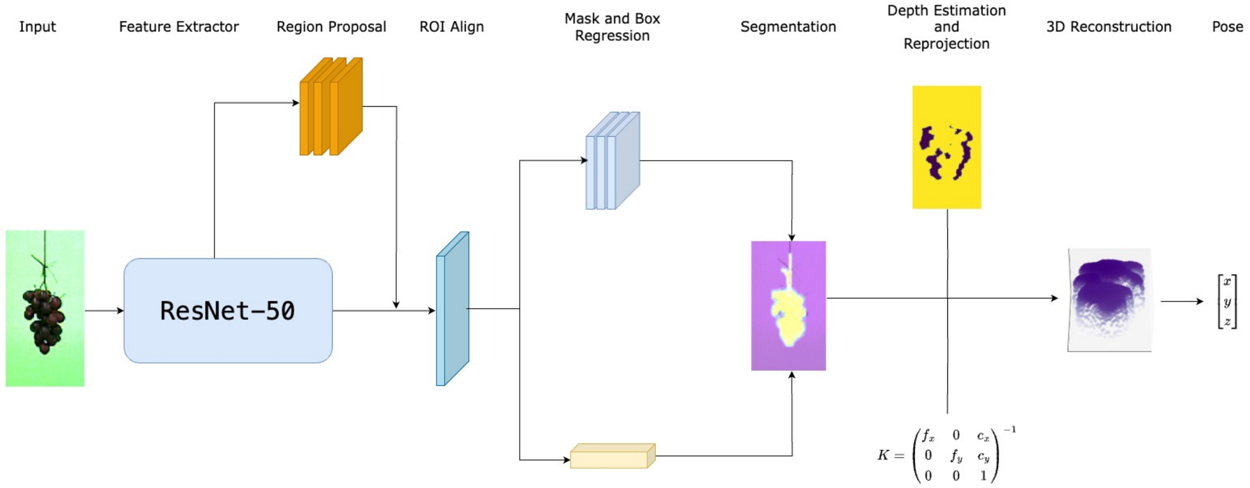
Input
Feature Extractor
Region Proposal
ROI Align
Mask and Box Regression
Segmentation
Depth Estimation and Reprojection
3D Reconstruction
Pose
ResNet-50
$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}^{-1}$
$\begin{bmatrix} x \\ y \\ z \end{bmatrix}$

Fig. 1: GrapeJuice localization pipeline. The model consists of a feature extractor, a region proposal network, an RoI align layer, and a mask and bounding box branch. The outputs are combined with the depth map to reconstruct a point cloud of the grapes for pose estimation.

B. Review Summary

Overall, *Yin et al.* presents a noteworthy contribution to the field of agricultural robotics and computer vision in relation to detecting the grape poses [3]. The paper is structured well, provides clear descriptions of the methodology used to estimate the pose of the grapes, and displays its results well. Additionally, the relevance of the paper to the field of agricultural robotics is clear, as it offers practical solutions to enhance the efficiency and automation of agricultural farming practices. Including more visual illustrations or examples of the detected fruit clusters and their pose estimation would increase the clarity of the paper, but this did not necessarily take away from the methodology proposed.

However, there are a few areas for improvement; while the paper effectively addresses the core problem and presents a novel approach, it could benefit from a more in depth evaluation of their methods. Their novel pose estimation pipeline was only evaluated on inference time, not accuracy, and does not test the implementation of such an algorithm on a physical robot - leaving the viability of the pipeline unexamined. Additionally, for a robotic implementation, the pipeline's detection capabilities are insufficient as a robot needs stem pose in addition to bunch pose for manipulation. These issues are addressed in our algorithmic reproduction and extension.

C. Points of Feedback

- *Clarifying the dataset selection process*: It would be helpful to provide more insights into how the dataset for training and testing was selected, including the criteria used for dataset annotation the diversity of the dataset in terms of fruit varieties, and lighting conditions commonly encountered in agricultural settings in order to contexualize the general environments this will be used in.
- *Elaborate on the pose estimation accuracy*: A detailed analysis of the accuracy of pose estimation under various environmental conditions, such as varying illumination, occlusions, and cluttered backgrounds, can be worthwhile. The paper could also discuss how the proposed method performs in challenging scenarios and potential strategies for improving the pose estimation accuracy in such cases.
- *Discuss the generalizability of the approach*: It would be beneficial to evaluate the generalizability of these methods across different fruit types beyond grape harvesting. The paper mentions about how the problem at hand can generalized to many different fruits given labeled data, although only grapes are discussed throughout. It would be valuable to discuss potential limitations for adapting the approach to different fruit species or agricultural tasks.
- *Discuss a robot implementation*: While the developed method is talked about, no mention of the applicability of using an actual robot, and the challenges associated with that is discussed. The main objective of the paper is to help improve agricultural robotics by developing a better pose detection method that can be applied to grape-harvesting robots, and the introduction focuses heavily on the need for robotics to help the agriculture scene in China. Thus, it would be worthwhile to discuss how the method could possibly be used and evaluated on a robot.

III. Algorithmic Reproduction & Extension

The key flaw in the system proposed in *Yin et al.* is that the pose evaluated is for the grape bunches [3]. However, a robotic harvester cannot and should not grasp the bunches, as they will get crushed without a custom soft gripper. More importantly, even with a grape gripper, the robot needs to additionally locate the stems for cutting and harvesting. From this arises another challenge: the stems are not necessarily located perfectly above the center of the grape bunches. In a farm environment, the stems of the grape bunches may have different orientations, shapes, sizes, or be occluded. To account for this, the vision pipeline was extended by adding a stem segmentation branch. By additionally training a MaskRCNN with labeled grape stem data, the grape localization pipeline can compute grape and stem masks which can then be combined with depth estimation to reconstruct a point cloud of the entire grape bunch. This provides a method to calculate the pose of the grapes to provide as input to a robot arm.

Additionally, the original pipeline did not contain an evaluated metric for their pose estimation system, since there is no ground truth data for grape bunch translation or rotation. While the pose cannot be explicitly evaluated without labels, the grape localization pipeline can be run on a mobile manipulator and be evaluated on the grasp success rate. We developed the enhanced grape and stem localization pipeline in Python and implemented it on the Fetch mobile manipulation platform.

A. Dataset

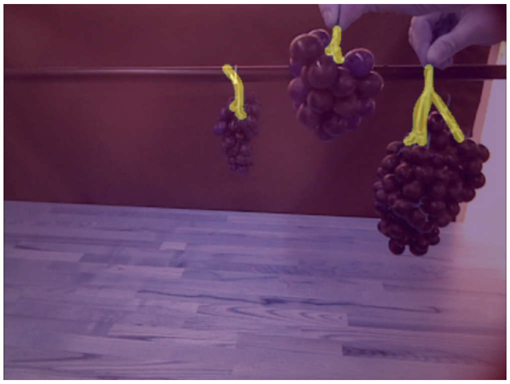

Fig. 2: Example stem mask label from custom grape stem dataset generated using Supervisely.

We employed two distinct datasets for training and evaluation purposes.

- *Wine Grape Instance Segmentation Dataset (WGISD)*: The Embrapa WGISD dataset consists of 300 images of size 1365×2048 containing a total of 4,432 grape clusters from a vineyard environment [1]. A subset of 137 images also contains binary masks identifying the pixels of each cluster, providing us with the masks of about 2020 clusters to train with. This dataset was specifically used to reproduce the bunch segmentation branch of the vision pipeline from *Yin et al.* [3].
- *A custom grape stem dataset*: Given resource constraints, we manually constructed a dataset of about 100 images of size 480×640 for stem masking, which is exemplified in 2. The dataset was generated using Supervisely, a free online dataset labeling service. The dataset contains images of grapes in a lab environment with labeled binary masks for stems. While the dataset represents a lab environment, a similar dataset can be constructed easily on Supervisely for a vineyard environment. This dataset was specifically used to train the stem segmentation branch of the vision pipeline.

B. Model Architecture

The first section of the pipeline consists of a neural network for bunch and stem segmentation. We adopted a MaskRCNN model architecture for both masking branches using the PyTorch and torchvision libraries in order to infer accurate masks for precise localization. The backbone network consists of a ResNet-50 feature pyramid network, which was obtained from the pre-trained models from torchvision. The embedded region proposal network uses the extracted feature map calculated by the ResNet FPN to propose RoIs for the mask and box predictors. A mask predictor and box predictor layer was thus added to the RoI Heads for mask and bounding box inference at the end. We made the following modifications to tailor the network to our task:

- The number of hidden layers was set to 256
- The number of classes was set to 1. Each branch needs to predict only one class independently (“grapes” or “stem”).

Branch	Optimizer	Batch Size	Learning Rate	Weight Decay
Bunch	Adam	4	1×10^{-4}	1×10^{-3}
Stem	Adam	2	5×10^{-7}	1×10^{-7}

TABLE I: Parameters used for training Bunch and Stem segmentation.

The branches were trained independently over 15 epochs using the parameters outlined in Table 1. The model ultimately achieved an **average IoU of 73.87%** on a test set that was not seen during training, after training on only 137 images. This replicates 90% of the performance with less than 1/8th the amount of data used in *Yin et al.* With a larger dataset, or data augmentation, even better performance can be achieved.

The second section of the pipeline consists of the pose estimation system. This section first relies on a well calibrated camera, ensuring a good estimate of the intrinsic camera matrix K. The depth map of the view is first estimated from the camera: we experimented with using the DepthAnything model proposed in *Kang et al*, for robust monocular depth estimation and qualitatively observed decent results [4]. However, for testing on the Fetch Mobile Manipulation platform, the depth map directly from Fetch’s infrared sensor was more reliable. Depending on whether the robot should navigate to the bunch or stem, the pipeline will first take the segmentation mask and crop the depth map to the bunch or stem. This provides the per pixel depth of the mask that can be used for 3D reconstruction. This 3D reconstruction algorithm is outlined in Algorithm 1.

Each pixel in the cropped depth map is first converted to homogeneous pixel coordinates, then transformed to homogeneous world coordinates by multiplying with the inverse camera matrix K. The third element of the transformed coordinates is then set to the depth value at the respective pixel coordinate from the depth map, producing a 3D point in the world frame. Performing this operation on each pixel of the cropped depth map results in a grape point cloud from which translation and rotation can be estimated. The mean point of the point cloud was estimated as the translation.

Algorithm 1 3D Reconstruction

```
Input: cropped depth map $D_{H\times W}$, camera matrix $K_{3\times 3}$
Output: Transformed point cloud $D'$
initialize $\mathbf{D'} = []$
for each pixel coordinate $(i, j)$ in $D$ do
    $\mathbf{p} = [i, j, 1]$
    $\mathbf{p'} = K^{-1}\mathbf{p}$
    $\mathbf{p'}[2] = D[i, j]$
    append $p'$ to $\mathbf{D'}$
end for
```

The rotational pose is obtained from the cylinder fitting process as shown in Fig. 3, reproduced from *Lin et al.* [3]. This method had decent results but showed potential for improvement. Since a grape cluster tapers towards the bottom, a cone would fit the point cluster better, so the RANSAC fitting algorithm was thus adapted for cone fitting instead as seen in Fig. 3. For both methods, the rotation matrix is computed based on data points to align a cone with its principal axis. The process begins by determining the direction and the principal axis using Principal Component Analysis. We then find the cross product and dot product of the direction vector with the positive Z-axis, which gives us an axis of rotation orthogonal to both of them. The Rodrigues' rotation formula is applied, which rotates a vector in three-dimensional space, using the previously mentioned cross and dot products, to give us our rotational matrix.

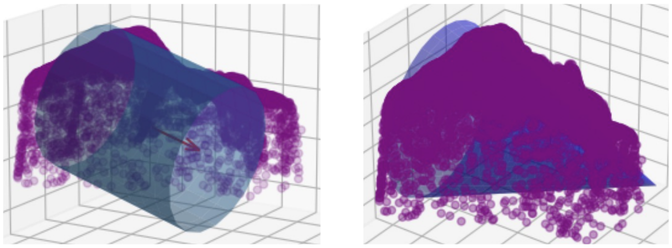

Fig. 3: Cylinder and Cone fitting for rotational pose estimation from grape point cloud.

IV. Experiments and Results

A. Experiment

A grape harvesting scenario was replicated in a lab environment to test the perception pipeline on Fetch. Grape bunches, either red or black, were suspended from a rod by their stem with painted twine to simulate grapes hanging from a vine. Fetch was placed in front of the grapes such that the bunch and stem were both within the camera view. The experiment setup can be seen in Fig. 4.

Two experiments were conducted to gauge the efficacy of this pipeline. For both experiments, a live RGB image of the scene was taken from Fetch's head camera and then run through the localization pipeline. Before motion planning, the resultant pose output is transformed from Fetch's *head_camera_rgb_frame* to the *base_link* frame using the ROS tf package. The rotational pose was assumed to be constant as the lab environment experiment can only offer a scene of a grape bunch hanging straight down.

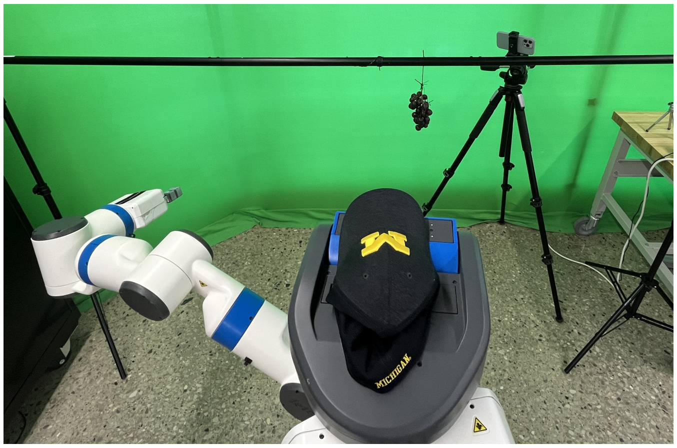

Fig. 4: Lab experiment setup. The grapes were suspended from a rod in front of the robot's view.

1) Grasp Experiment: The motion planner uses the pose evaluated by the localization pipeline to move the gripper to the stem (or bunch), and grasp. We recorded whether the robot was successful in grasping the grapes or not. This test was conducted a total of 42 times using images taken at a variety of orientations of the robot's head, base and camera tilt. The grasp success rate is simply evaluated as

$$\frac{\text{Number of successful grasps}}{\text{Number of attempts}}\%$$

2) Gripper Pose Error Experiment: Gripper Pose Error Experiment: Even if the robot is able to successfully grasp, there exists some error between the gripper pose and actual stem or bunch. So, an additional test was added to evaluate grasp with greater precision. The motion planner uses the pose evaluated by the localization pipeline to move the gripper to the grape location, but does not close the gripper. At this point, we manually measure and record the absolute error in the x, y and z direction between the actual grape location (whether it be bunch or stem) and the gripper frame (center of the gripper). This test was conducted a total of 16 times.

B. Results

Through experimentation, we found that the grape localization system performs well when implemented on a real robot platform. With 36 successful grasps out of 42 attempts, the Grasp Experiment found a **grasp success rate of 85.71%** which would satisfy the requirements of a real world robotic harvester. Fig. 5 shows the results of the Gripper Pose Error Experiment, which found an **average gripper pose error of 6.08 cm, 3.80 cm, and 3.11 cm** in the x, y and z directions respectively. These errors are on the order of hundredths of a meter, indicating that the localization pipeline is performing accurately and the pose outputs are close to the actual location of the grapes.

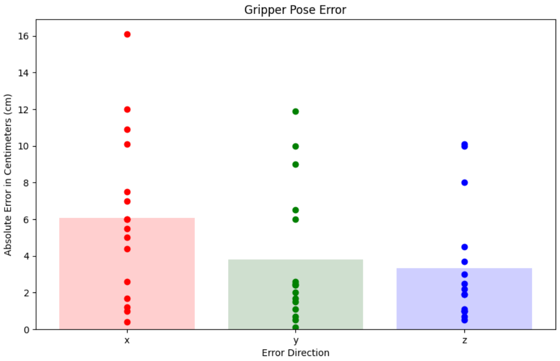

Fig. 5: Gripper pose over 16 trials. The average gripper pose error was 6.08 cm, 3.80 cm and 3.11 cm in the x, y and z directions.

V. Conclusions & Future Work

Our research presents a robust grape localization pipeline tailored for robotic harvesting applications. Addressing the limitations of existing systems, our approach integrates a novel stem segmentation branch into the vision pipeline, enabling precise localization of both grape bunches and their stems. Through experimentation on the Fetch mobile manipulation platform, we demonstrated the performance of our localization pipeline in real-world scenarios. The **grasp success rate of 85.71%** obtained from our experiments underscores the practical viability of our approach for autonomous harvesting tasks. Furthermore, a low average **gripper pose error of 6.08 cm, 3.80 cm, and 3.11 cm** in the x, y, and z directions respectively, reaffirms the accuracy of our pose estimation system. These results demonstrate the potential of robotics and deep learning-based perception to revolutionize harvesting in agriculture.

In the future we plan to expand the capabilities of the localization system by training the masking model on bunch and stem data from a broader set of environments. This would help the vision model generalize better and output even better mask results. Another key area to expand this study would be to incorporate labeled pose data for the grapes. This could open up the possibility of implementing learning based pose estimation methods. The last area for improvement would be improving runtime by parallelizing the localization pipeline on GPUs. This would drastically reduce pose inference time, and also open the model to being augmented with a probabilistic filter at the end, such as a Kalman Filter. These adjustments would greatly improve the applicability of this system to a real world application.

References

[1] T. Santos, “Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association,” *Computers and Electronics in Agriculture*, vol. 170, 2020.

[2] C. Gao, “Improved binocular localization of kiwifruit in orchard based on fruit and calyx detection using yolov5x for robotic picking,” *Computers and Electronics in Agriculture*, vol. 217, 2024.

[3] W. Yin, “Fruit detection and pose estimation for grape cluster–harvesting robot using binocular imagery based on deep neural networks,” *Frontiers in Robotics and AI*, vol. 8, 2021.

[4] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, “Depth anything: Unleashing the power of large-scale unlabeled data,” in *CVPR*, 2024.