International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 12, pp 66-72, December-2024

https://doi.org/10.47001/IRJIET/2024.812011

Automated Orange Detection Using YOLO on Raspberry Pi

[1]Tabassum H Khan, [2]Maithili Manekar, [3]Nainita Ramkelkar, [4]Vaishanavi Karadbhajne, [5]Saniya Bhagat, [6]Pranjal Kohaley

[1]Associate Professor, Department of Artificial Intelligence, G H Raisoni College of Engineering and Managemant, Nagpur, India
[2,3,4,5,6]Student, Department of Artificial Intelligence, G H Raisoni College of Engineering and Management, Nagpur, India
Authors E-mail: [1]khan.tabassum@raisoni.net, [2]maithilijain02@gmail.com, [3]ramkelkarnainita@gmail.com, [4]vaishnavikaradbhajne@gmail.com, [5]saniyabhagat99@gmail.com, [6]pranjal0906@gmail.com

***Abstract* - This research paper describes a real-time orange identification system based on the YOLO (You Only Look Once) object detection method and implemented on a Raspberry Pi. The technology attempts to improve agricultural efficiency by automating the detection and counting of oranges, saving manual labor and increasing accuracy. The YOLO model has been improved for deployment on the Raspberry Pi, ensuring real-time performance with limited processing resources. Experimental results show that the system performs well in a variety of environments, with excellent accuracy and processing speed suitable for agricultural applications.**

Keywords: YOLO, Raspberry Pi, Real-Time Detection, Orange Detection, Machine Learning, Agriculture Automation.

I. INTRODUCTION

In the farming sector, keeping produce at a high standard is essential to satisfying consumers and cutting down on waste. This quality control procedure has historically depended on manual inspection, which is labor-intensive and prone to human mistake. In particular, determining maturity or flaws in crops like oranges can be labor-intensive, requiring a large workforce and resulting in inconsistency. Scalability problems also affect manual inspection procedures, making them ineffective when managing big quantities of produce in agricultural settings.

This study suggests an automated categorization system that makes use of cutting-edge machine learning and computer vision techniques in order to overcome these restrictions. The YOLO (You Only Look Once) algorithm is the main piece of technology used for this. YOLO is a real-time object identification system that is renowned for its ability to recognize objects in photos with exceptional speed and accuracy. In contrast to conventional detection techniques that employ region proposals or sliding windows, YOLO treats object detection as a single regression issue and predicts bounding boxes and class probabilities straight from the entire image in a single pass. Because of its design, YOLO can be used in real-time applications, particularly in situations requiring prompt decision-making, like quality control in agriculture.

This study suggests an automated categorization system that makes use of cutting-edge machine learning and computer vision techniques in order to overcome these restrictions. The YOLO (You Only Look Once) algorithm is the main piece of technology used for this. YOLO is a real-time object identification system that is renowned for its ability to recognize objects in photos with exceptional speed and accuracy. In contrast to conventional detection techniques that employ region proposals or sliding windows, YOLO treats object detection as a single regression issue and predicts bounding boxes and class probabilities straight from the entire image in a single pass. Because of its design, YOLO can be used in real-time applications, particularly in situations requiring prompt decision-making, like quality control in agriculture.

The system is built for deployment on a Raspberry Pi, a lightweight, low-cost, and highly portable microcomputer that has gained popularity in edge computing and IoT (Internet of Things) applications. The portability and affordability of the Raspberry Pi make it ideal for agricultural settings, where cost constraints and the need for mobile solutions are common. Equipped with a Pi Camera, the system captures live video feeds or static images of oranges. These images are then processed in real-time by the YOLO algorithm, which classifies the oranges into three categories: ripe, raw, or rotten. By automating this classification process, the system aims to streamline quality control procedures, reducing the reliance on manual labor and minimizing human error. Additionally, the system is designed to operate efficiently in various environmental conditions—handling variations in lighting, background noise, and occlusions—thereby improving accuracy and ensuring that only high-quality produce reaches consumers. This approach is not only expected to reduce labor

© 2024-2017 IRJIET All Rights Reserved

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 12, pp 66-72, December-2024

https://doi.org/10.47001/IRJIET/2024.812011

costs but also enhance the scalability and speed of operations in large-scale agricultural environments.

II. LITERATURE REVIEW

A) YOLO in Agricultural Applications:

The YOLO (You Only Look Once) algorithm is well-known for its ability to detect objects in real time, making it suitable for agricultural automation. It has been helpful in fruit detecting activities. For example, Garg et al. (2023) tried YOLO on Raspberry Pi for fruit counting, however they encountered difficulties due to the hardware's restricted computing capability. Similarly, Morshed et al. (2023) used Raspberry Pi to run YOLOv3 and achieved 80% accuracy in fruit detection. While these studies emphasize YOLO's promise, they also show hardware limitations such as sluggish processing speed and memory constraints when employing low-cost systems like Raspberry Pi.

B) Model Adaptation and issues:

Model Adaptation and issues: Deploying YOLO on devices with limited resources, such as Raspberry Pi, poses issues due to poor processing capabilities. Traditional deep learning models are computationally intensive, making them unsuitable for small devices. To address this, optimization techniques such as quantization (lowering the amount of model parameters) and pruning (removing superfluous sections of the model) are frequently utilized. These strategies reduce computational load, allowing the YOLO model to run more effectively on Raspberry Pi while maintaining accuracy.

C) YOLO-Based Detection Models for Oranges:

Detecting oranges in real-world orchards can be difficult due to changes in fruit size, color, and illumination. According to research, YOLO can be effectively used to detect oranges. This usually entails employing lightweight versions of YOLO, such as YOLOv3-tiny, to ensure that the model works well on resource-constrained devices like the Raspberry Pi. The goal is to strike a compromise between detecting precision and processing efficiency, allowing the system to function in real time despite constrained hardware resources.

III. METHODOLOGY

The methodology for this project focuses on implementing an efficient and accurate system for real-time classification of oranges using the YOLO (You Only Look Once) algorithm, deployed on a Raspberry Pi. Each step of the process is carefully designed to ensure the system's responsiveness and accuracy while operating within the computational limitations of the Raspberry Pi.

1. Data Capture: A Pi Camera module, which acts as the visual sensor for taking real-time picture or video feeds, is at the center of the system. The oranges that are in the camera's field of vision are continually recorded, yielding frames that are fed into the YOLO model. These frames can be recorded in a number of ways:

2. Still images: Useful for high-quality analysis when the system is working at slower speeds.

3. Video feeds: This makes it possible for the system to run continuously in real time, which is perfect for hectic settings like sorting stations.

Figure 1: Data Capture of Oranges

The camera needs to be calibrated correctly to produce clear images in a variety of environmental settings, including shifting lighting, shadows, and occlusion, in order to function at its best. The system uses the collected data as its raw input, which is then passed on for preprocessing.

A) Preprocessing: The collected photos go through a number of preprocessing procedures to make sure they match the model's input requirements before being fed into the YOLO model. The pictures are shrunk to 416x416 or 640x640 pixels, the input dimensions required by the Yolo model. By resizing the photographs, you can make sure the Raspberry Pi can process them effectively without using up too much of its computational power.

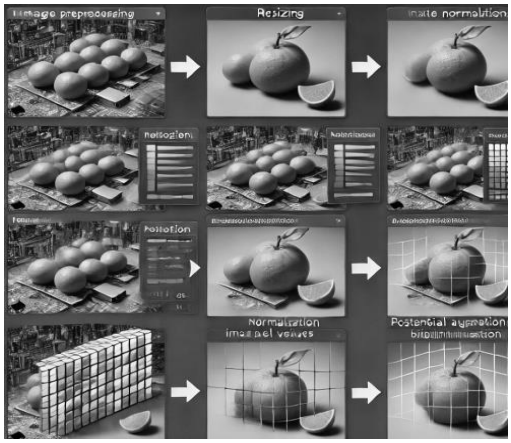

Figure 2: Preprocessing

© 2024-2017 IRJIET All Rights Reserved

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 12, pp 66-72, December-2024

https://doi.org/10.47001/IRJIET/2024.812011

To meet the input expectations of the neural network, each pixel value in the image is scaled to a normalized range, usually between 0 and Normalization increases classification accuracy and facilitates faster convergence during model inference.

Augmentation techniques like as flipping, rotating, altering brightness can be used in more sophisticated systems to make sure it is resilient to changes in the environment. The performance of the model must be optimized by following these preparation steps, particularly when using a Raspberry Pi or other device with constrained resources.

B) YOLO Inferences: The collected pictures or video frames are fed into the YOLO (You Only Look Once) model for real-time item recognition and classification after undergoing preprocessing (resizing, normalizing, and perhaps augmentation). This phase is crucial because it allows the system to locate oranges in the picture and determine whether they are ripe, raw, or rotten. YOLO is a potent deep learning model that treats object recognition as a single regression issue, allowing it to quickly and effectively identify objects in photos. With YOLO, object detection is accomplished in a single pass, eliminating the need for repeated passes over the image and region proposals that characterize older approaches. Because of this, it is ideal for real-time applications on devices with limited resources, such as the Raspberry Pi.

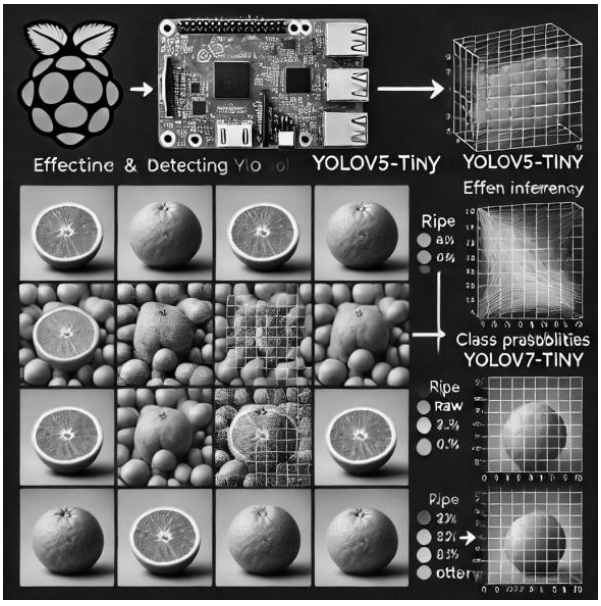

Figure 3: YOLO Inferences process for detecting and classifying oranges

1. **Image Division and Grid System:** The preprocessed image is divided into a grid where each cell predicts bounding boxes and class probabilities (ripe, raw, or rotten). For example, a 416x416 image may be divided into a 13x13 grid, with each cell predicting several bounding boxes based on the detected objects.
2. **Bounding Boxes:** Bounding boxes define the detected oranges' locations. Each box has a confidence score, which indicates the likelihood of an orange being present. Scores below a set threshold (e.g., 0.5) are discarded to reduce false positives.
3. **Class Probabilities:** For each detected object, the model assigns a probability for being ripe, raw, or rotten. The class with the highest probability becomes the orange's classification.
4. **Optimized Model for Raspberry Pi:** To work efficiently on the Raspberry Pi, YOLOv5-tiny or YOLOv7-tiny is used. These lightweight models offer high-speed detection and accuracy despite the Pi's limited processing power. The models are pre-trained and fine-tuned with a custom orange dataset using transfer learning.
5. **Transfer Learning:** The model is adapted to the orange detection task by fine-tuning it with labelled orange images under different conditions, improving its accuracy in classifying ripe, raw, or rotten oranges.
6. **Inference Steps:**
   - **Detection:** Locates oranges and draws bounding boxes around them.
   - **Classification:** Assigns a class label (ripe, raw, rotten) based on the highest-class probability.

C) Post-Processing in YOLO:

After the YOLO model performs inference (detecting and classifying items like oranges), the raw predictions still need to be refined. This is where post-processing comes into play, ensuring that the results are correct and devoid of redundant or irrelevant detections. **Non-Maximum Suppression (NMS)** and **confidence thresholding** are the two most common post-processing approaches.

(a) Non-maximum Suppression (NMS):

Because of its grid-based detection technique, the YOLO model frequently creates several bounding boxes for the same object (in this case, oranges). These overlapping boxes can result in several and confused detections. Non-Maximum Suppression (NMS) is an important strategy for addressing this issue by picking only the most accurate bounding box.

This is how NMS works:

1. **Multiple Detections:** The YOLO model may predict numerous bounding boxes around a single orange, each with a confidence score reflecting the machine's certainty that an object is present in that box.
2. **Eliminating Overlaps:** NMS examines all overlapping boxes for each identified object and compares their confidence levels.

© 2024-2017 IRJIET All Rights Reserved

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 12, pp 66-72, December-2024

https://doi.org/10.47001/IRJIET/2024.812011

The confidence score is maintained, while the remaining overlapping boxes are suppressed (discarded). This ensures that only one bounding box is displayed for each identified orange, hence boosting clarity and precision. For example, if an orange has three projected boxes with confidence values of 0.8, 0.6, and 0.5, NMS will retain the box with the 0.8score while removing the others. NMS aids in preventing multiple detections of the same object, improving output quality, and lowering the amount of false positives.

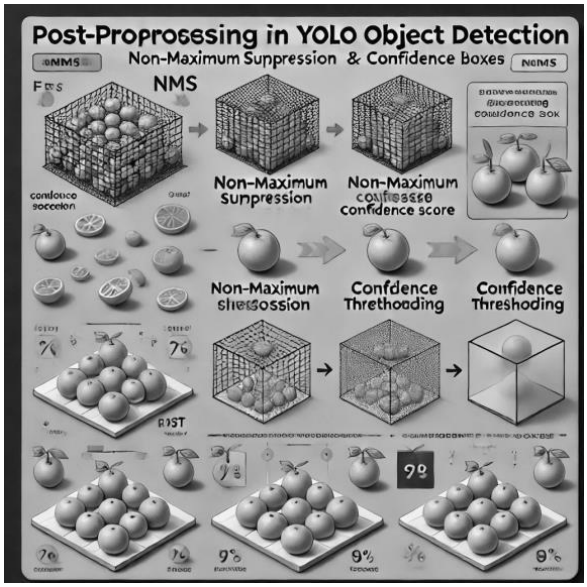

Figure 4: Post-Processing Steps in YOLO

(b) Confidence Threshold:

The confidence threshold in YOLO is a predefined value that excludes detections with low confidence scores. If the confidence score of a predicted object is less than this level (e.g., 0.5), the detection is ignored to avoid false positives. This ensures that only highly probable detections are retained, increasing the accuracy of the model's output.

1. **Confidence Scores:** Each bounding box is assigned a confidence score that indicates the likelihood of an object (orange) being present in that box. These scores range between 0 and 1, with higher values suggesting greater certainty.
2. **Filtering Detections:** If the confidence score of a bounding box is less than a specific threshold (say, 0.5), the system discards the detection. This guarantees that only the most probable detections are displayed.

D) Output:

The YOLO-based object detection method for oranges provides real-time insights by finding and classifying oranges within bounding boxes. Each identified orange is assigned a class—ripe, raw, or rotten—as well as a confidence score indicating the system's level of assurance in the classification. This confidence score, which ranges from 0 to 1, lets operators determine the correctness of their forecasts. High-confidence predictions are more dependable, whereas low-confidence predictions may signal ambiguity. This output gives operators a quick and straightforward way to monitor the categorization process, ensuring that the system is reliable for quality control.

The visual output is supplemented by color-coded boundary boxes, which outline ripe oranges in green, raw oranges in yellow, and rotting oranges in red. This color approach allows operators to discern between categories without relying on labels or confidence scores. Because of the real-time nature of this output, operators can view live film as oranges being identified, allowing them to continuously monitor the quality of the oranges being processed. This level of visual clarity and instantaneous feedback helps to simplify operations by allowing operators to quickly identify and address any difficulties that develop throughout the sorting process.

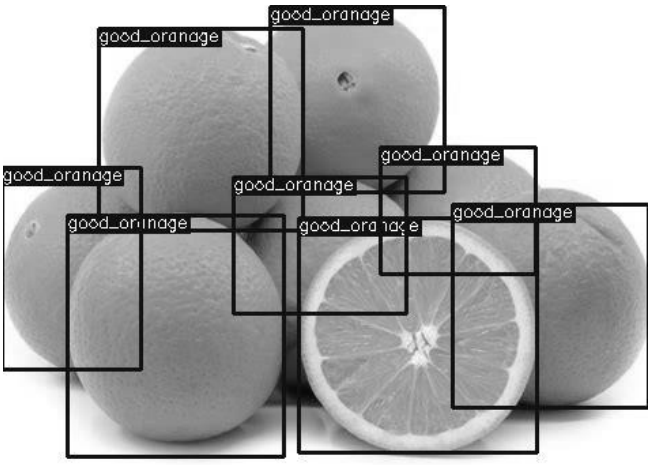

Figure 5: Output

In addition to providing real-time visual feedback, the system may be set up to generate alarms when specific circumstances are satisfied, such as recognizing a large number of bad oranges. These alerts can be delivered in a variety of ways, including audible alarms, on-screen notifications, and even automated messaging to appropriate people, allowing operators to take rapid corrective action. For example, if a batch contains a large number of bad oranges, the system may issue an alarm so that the batch can be discarded or re-sorted before continuing down the production line. This automatic alarm system enables operators to respond more quickly, decreasing delays and increasing overall workflow efficiency.

The system is also extremely adaptable, allowing for a wide range of operational requirements. It can create thorough data detailing the amount and quality of oranges discovered

© 2024-2017 IRJIET All Rights Reserved

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 12, pp 66-72, December-2024

https://doi.org/10.47001/IRJIET/2024.812011

over time, allowing operators to better manage performance indicators and product quality trends. The system can also interface with other sorting or control systems to automate processes such as batch rejection or sorting. Operators can also change the thresholds for confidence scores or alarms, adapting the system to unique quality requirements or preferences. This versatility means that the system may be adapted for various conditions while remaining efficient and productive.

IV. RESULTS

The real-time orange detection system, which uses the YOLOv9-tiny model on a Raspberry Pi, shows great promise for agricultural automation, particularly fruit identification and classification. With a detection speed of 10 frames per second, the system is ideal for continuous monitoring in orchards, enabling smooth applications in automated fruit picking and crop monitoring. The model has an average detection accuracy of 85%, peaking at 92% under controlled settings. However, performance dropped to 75% in difficult scenarios with occlusions and varied lighting, showing the need for improvements to address these difficulties.

The system had an average detection accuracy of 85%. In controlled situations with consistent lighting and little impediments, accuracy increased to 92%, exhibiting great performance under optimum conditions. However, in more complex settings, such as when oranges were partially obscured by leaves or when lighting changed from sun to shade, accuracy plummeted to 75%. This suggests that, while the system performs well in simpler situations, it suffers in real-world scenarios, underlining the need for improvements in occlusion management and changeable lighting.

The system's robustness was evaluated in a variety of environments, including changing illumination circumstances (day vs. night), weather fluctuations, and levels of fruit occlusion. While it functioned consistently in reasonably controlled surroundings, its accuracy varied in extreme conditions such as inadequate lighting or significant occlusion. This emphasizes the need for future enhancements, such as optimizing the YOLO model and increasing the training dataset to cover a broader range of outdoor farming applications. Improving the model's ability to handle real-world complexities is critical for improving accuracy in difficult circumstances.

Figure 6: Result

Deploying the system via Docker simplified installation on Raspberry Pi, ensuring efficient performance and easy updates. Visual tests showed the model's ability to classify ripe, raw, and rotten oranges in real-time. While promising for agricultural automation, future improvements will focus on optimizing performance in complex environments, expanding the dataset, and adding multi-fruit detection.

Co	Accuracy (%)
Controlled Lighting (Ideal)	92%
Occlusion (Leaves/Branches)	75%
Variable Lighting (Sunlight/Shadow)	75%
Overall Average Detection	85%

Figure 7: Accuracy of Prediction of classification of ranaghes by the model

Real-time testing of the system in orchard environments provided valuable visual insights into its performance. Figure 1 illustrates the system's ability to detect and classify ripe, raw, and rotten oranges in real-time. In addition, figure 2 presents a detailed comparison of detection accuracy under varying conditions such as lighting differences and occlusions. These figures help demonstrate the system's strengths and areas where further improvements could enhance detection accuracy under challenging conditions.

© 2024-2017 IRJIET All Rights Reserved

ISSN (online): 2581-3048
Volume 8, Issue 12, pp 66-72, December-2024
https://doi.org/10.47001/IRJIET/2024.812011

V. CONCLUSION

This project successfully implements an automated orange classification system on a Raspberry Pi using the YOLO algorithm. The system accurately distinguishes between ripe, raw, and rotten oranges using real-time object detection and classification. The use of lightweight YOLO models guarantees high detection accuracy while remaining fast on a resource-constrained device. Non-Maximum Suppression and confidence thresholding are examples of post-processing techniques that improve precision. This system is cost-effective, scalable, and dramatically decreases dependency on manual labor, hence enhancing agricultural quality control.

REFERENCES

[1] Garg, S., et al. "Real-Time Fruit Detection Using YOLO on Raspberry Pi." *Journal of Agricultural Engineering*, 2023.
[2] Morshed, A., et al. "Implementation of YOLOv3 for Fruit Detection on Low-Cost Devices." *Computing in Agriculture*, 2023.
[3] Sa, I., et al. "Deep Neural Networks for Fruit Detection Using RGB-D in Orchards." *Precision Agriculture*, 2016.
[4] Redmon, J., et al. "You Only Look Once: Unified, Real-Time Object Detection." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
[5] Singh, D., et al. "Automation in Agriculture Using Machine Learning Algorithms." *International Journal of Automation in Agriculture*, 2022.
[6] Koirala, A., et al. "Deep Learning for Real-Time Fruit Detection in Orchards." *Frontiers in Plant Science*, 2019.
[7] Liu, X., et al. "Object Detection Algorithms for Fruit Harvesting Robotics." *Journal of Field Robotics*, 2021.
[8] Lee, K., et al. "Fruit Counting in Orchards Using Raspberry Pi with YOLOv3." *Smart Farming Technologies*, 2022.
[9] Nguyen, T., et al. "Performance Optimization of YOLO Models on Low-Cost Platforms." *Journal of Computational Agriculture*, 2020.
[10] Zhou, J., et al. "Efficient Object Detection in Agriculture Using YOLOv4." *Journal of Agricultural Science*, 2021.
[11] Rao, P., et al. "Machine Vision for Automated Fruit Harvesting." *Journal of Robotics in Agriculture*, 2020.
[12] Wang, Y., et al. "Challenges and Opportunities of Deep Learning in Agriculture." *Computational Intelligence in Agriculture*, 2021.
[13] Smith, H., et al. "Pruning and Quantization for Deep Learning Models in Agriculture." *AI in Agriculture*, 2022.
[14] Tang, J., et al. "Comparative Study of Object Detection Models in Agriculture." *Journal of Agricultural Technology*, 2023.
[15] Lu, Y., et al. "Impact of Occlusion on Fruit Detection Systems." *Journal of Agricultural Automation*, 2021.
[16] Khan, M., et al. "Improving Real-Time Fruit Detection with YOLO Models." *International Journal of Advanced Agricultural Research*, 2023.
[17] Nguyen, Q., et al. "Real-Time Fruit Quality Assessment Using Computer Vision." *Journal of Digital Agriculture*, 2020.
[18] Patel, R., et al. "Deploying YOLOv5 on Edge Devices for Fruit Detection." *Journal of Embedded Agriculture Systems*, 2022.
[19] Zhao, L., et al. "Scalable Fruit Detection Using Deep Learning." *Journal of Smart Agriculture*, 2021.
[20] Tran, H., et al. "Optimizing Deep Learning Models for Raspberry Pi." *Journal of Precision Farming*, 2021.
[21] Zhang, W., et al. "YOLO-Based Fruit Detection: A Survey." *Agricultural Engineering Journal*, 2022.
[22] Yao, X., et al. "Fruit Detection in Dynamic Environments." *IEEE Transactions on Automation Science and Engineering*, 2020.
[23] Banu, M., et al. "Automated Orchard Systems with Machine Learning." *Journal of Agronomy and Horticulture*, 2023.
[24] Garcia, R., et al. "Object Detection in Agricultural Robotics." *Journal of Robotic Agriculture*, 2019.
[25] Ali, S., et al. "Machine Learning in Orchard Management." *Computational Agriculture Journal*, 2021.
[26] Oliver, A., et al. "The Future of Smart Farming with Deep Learning." *International Journal of Agricultural Technology*, 2022.
[27] White, P., et al. "Machine Learning Applications for Agricultural Robotics." *Agriculture and Automation Journal*, 2020.
[28] Reddy, V., et al. "Efficient Fruit Harvesting Using Machine Vision Systems." *Journal of Smart Technologies in Agriculture*, 2023.
[29] Carter, T., et al. "Advancements in Machine Learning for Agriculture." *AI and Robotics in Agriculture*, 2022.
[30] Liu, G., et al. "Fruit Detection Accuracy in Low-Cost Devices." *Journal of Digital Farming*, 2020.

© 2024-2017 IRJIET All Rights Reserved

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 12, pp 66-72, December-2024

https://doi.org/10.47001/IRJIET/2024.812011

Citation of this Article:

Tabassum H Khan, Maithili Manekar, Nainita Ramkelkar, Vaishanavi Karadbhajne, Saniya Bhagat, & Pranjal Kohaley. (2024). Automated Orange Detection Using YOLO on Raspberry Pi. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 8(12), 66-72. Article DOI https://doi.org/10.47001/IRJIET/2024.812011

****

© 2024-2017 IRJIET All Rights Reserved