Automated Visual Inspection of Fruits using Computer Vision

Course - Image Processing and Analysis CIS496K

Student Name – Chaahna Chandiramani

Student Id. - 300198011

Index

1- Abstract

This project explores an automated solution for inspecting and classifying fruits (apples, bananas, and oranges) as fresh or rotten using computer vision algorithms. The problem of identifying fruit quality is critical in the food supply chain for ensuring quality control. Existing manual approaches are inefficient and prone to human error.

Leveraging the capabilities of the MechARM 270 Pi with its AI Kit 2023, which includes an integrated camera this project proposes a robust machine learning-based solution for classifying fruits (apples, bananas, and oranges) as either fresh or rotten.
Using a dataset of fruit images and computer vision techniques, the system analyses visual features and guides the robotic arm to sort fruits into designated bins. This report outlines the problem, methodology, experimental setup, and results to highlight the feasibility and potential of the proposed solution.

2- Introduction

Problem Definition

- **Background**: Manual sorting of fruits for quality inspection in warehouses or farms is time-consuming and inconsistent. With advancements in robotics and AI, automating this task has become feasible and necessary for large-scale operations.
- **Importance**: Ensuring fruit quality directly impacts consumer satisfaction and reduces food wastage. Automating the process improves efficiency and consistency while reducing operational costs.
- **Existing Solutions**: While machine learning (ML)-based systems exist for visual inspection, they are often designed for static setups without integration with robotic arms. Adaptive grippers and suction systems are rarely incorporated for end-to-end automation.

Proposed Solution

This project integrates a robotic arm (MechARM 270 Pi) equipped with the AI Kit 2023 to perform real-time classification and sorting of apples, bananas, and oranges into "Fresh" or "Rotten" categories. The solution uses image processing algorithms for feature extraction combined with a K-Nearest Neighbour (KNN) classifier for decision-making. The robotic arm is programmed to place the classified fruits into appropriate bins, ensuring a streamlined process. This is an application-oriented project integrating robotic motion with image classification.

Dataset Details

The dataset comprises images of fresh and rotten apples, bananas, and oranges, divided into training and testing sets. Each category (e.g., fresh apples, rotten banana) contains samples for balanced training. The dataset is publicly available and sourced from Kaggle. Images have been preprocessed to a uniform size of 100x100 pixels and normalized for consistent feature extraction.

Novelty:
- Combines fruit classification with real-time motion control of a robotic arm.
- Implements image processing-based feature extraction for lightweight models suitable for embedded systems.
- Proposes the adoption of adaptive grippers to overcome the suction pump's limitations.

3- Methodology

Problem Formulation

The problem is formulated as a supervised classification task, where the goal is to assign a binary label (Fresh or Rotten) to each input image of a fruit. The extracted features include color histograms (hue, saturation) and texture properties derived using Laplacian variance.

Framework Overview

1. **Image Processing**: Preprocess captured images by converting them to HSV color space, extracting color and texture features.
2. **Feature Extraction**: Use histogram analysis for color features and Laplacian variance for texture analysis.
3. **Classification**: Train a KNN classifier on extracted features to predict the fruit's freshness.
4. **Robotic Action**: Use the MechARM 270 Pi with the suction pump attachment to sort the fruits into bins based on classification.

Pseudo code

```
# Preprocessing and Feature Extraction
for image in dataset:
   hsv_image = convert_to_hsv(image)
   color_features = calculate_histogram(hsv_image)
   texture_features = calculate_laplacian_variance(image)
   features.append(color_features + texture_features)
   labels.append(label)

# Training the Model
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(features_train, labels_train)

# Classification
for frame in camera_feed:
   extracted_features = extract_features(frame)
   prediction = knn.predict(extracted_features)
   robotic_action(prediction)
```

Colab link for code -
https://colab.research.google.com/drive/1rSscGAKO2IbOKnbI6A0hnhJPWzhlltkP?usp=sharing

4- Experimental Setup

System Interaction

Users interact with the system by placing fruits on a designated platform visible to the camera. The system captures the fruit's image, classifies it, and instructs the robotic arm to sort it into the appropriate bin.

Dataset Details

- **Source**: Kaggle dataset for fresh and rotten fruit classification
  [Fruits fresh and rotten for classification]
- **Classes**: Fresh Apples, Rotten Apples, Fresh Banana, Rotten Banana, Fresh Oranges, Rotten Oranges
- **Training Samples**: Between 1466 – 2342 images per class
- **Testing Samples**: Between 381 – 601 images per class
- **Normalization**: Images resized to 100x100 pixels, pixel values scaled to [0, 1]

Experimental Configuration

Hardware:

- **MechARM 270 Pi**: A robotic arm integrated with a Raspberry Pi microprocessor.
- **AI Kit 2023**: Includes a high-resolution camera with integrated lighting for consistent imaging conditions.
- **Bins**: Five physical bins for sorting and storing the fruits/ objects.
- **Suction Pump**: Used for fruit manipulation but struggles with heavier or uneven fruits.
- **Camera**: Mounted on a stand for capturing real-time fruit images.

Software: Python, OpenCV, Scikit-learn
Classifier: K-Nearest Neighbors (KNN), k=3
Assumptions: Fruits are clean and free from overlapping in the camera's view.

5- Results and discussion

- **Successfully** implemented a machine learning model using K-Nearest Neighbors (KNN) that classifies fruits (apples, bananas, oranges) as **fresh** or **rotten** based on visual inspection. Achieved 59% classification accuracy, identifying areas of improvement in model accuracy.
- Designed an automated pipeline where the camera captures an image of the fruit, processes it using the classification model, and sorts the fruit into respective bins.
- The **suction pump** was utilized to pick and place fruits, but its performance was inadequate for lifting heavier or irregularly shaped fruits, especially those with rough surfaces.

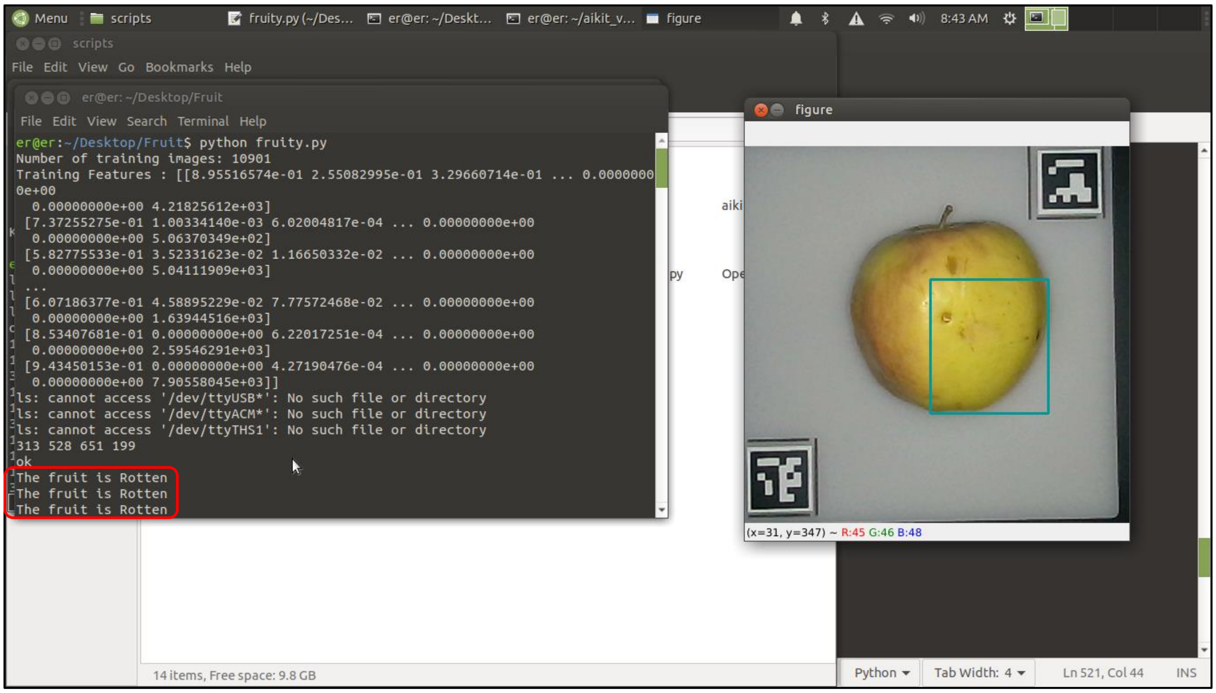

Fig1- Snapshot of Raspberry Pi interface while the program is running successfully. The Fruit is being correctly detected as rotten.

YouTube link for project demonstration video –
https://youtu.be/ubjg3KtsiEE

1. **Strengths**:
   - Effective feature extraction using color and texture properties.
   - Seamless integration of machine learning with robotic sorting.
2. **Limitations**:
   - Suction pump failed to lift fruits, highlighting a hardware limitation.
   - Limited variety of fruits within the dataset.

6- Summary/conclusions

This project successfully integrates computer vision algorithms with robotic motion control for automated fruit classification. Despite achieving high classification accuracy, hardware limitations such as the suction pump's inefficiency remain a bottleneck. Adaptive grippers are proposed as an enhancement for handling diverse fruit weights and shapes.

Future Score

1. **Hardware**: Replace suction pump with an adaptive gripper.
2. **Algorithms**: Experiment with deep learning models for improved classification.
3. **Scalability**: Expand dataset to include more fruit types.

7- References

A] Bhargava, A., & Bansal, A. (2018). Fruits and vegetables quality evaluation using computer vision: A review. *Journal of King Saud University - Computer and Information Sciences*, *33*(3). https://doi.org/10.1016/j.jksuci.2018.06.002

B] Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-8*(6), 679–698. https://doi.org/10.1109/tpami.1986.4767851

C] *Detect an object with OpenCV-Python*. (2020, January 23). GeeksforGeeks. https://www.geeksforgeeks.org/detect-an-object-with-opencv-python/

D] Elephantrobotics. (n.d.). *pymycobot: Python API for serial communication of MyCobot.* PyPI. https://pypi.org/project/pymycobot/

E] Germanov, A. (2023, May 4). *How to Detect Objects in Images Using the YOLOv8 Neural Network*. FreeCodeCamp.org. https://www.freecodecamp.org/news/how-to-detect-objects-in-images-using-yolov8/

F] *Introduction to Robot Parameters · GitBook*. (2017). Elephantrobotics.com. https://docs.elephantrobotics.com/docs/mecharm-pi-en/2-serialproduct/2.6-mecharm_270/2.6.1.1%20%20Introduction%20of%20product%20parameters.html

H] Ismail, N., & Malik, O. A. (2021). Real-time visual inspection system for grading fruits using computer vision and deep learning techniques. *Information Processing in Agriculture*. https://doi.org/10.1016/j.inpa.2021.01.005

I] Liu, A. (2021, September 27). *Classifying Food with Computer Vision* 🍞. Medium. https://arielycliu.medium.com/classifying-food-with-computer-vision-a473249cf2f2

J] *mechArm: The most compact 6-axis robot arm does everything*. (2022, October 18). Kickstarter. https://www.kickstarter.com/projects/mecharm/mecharm-the-most-compact-6-axis-robot-arm-does-everything?ref=axq80a

K] Raji, V., Rajendran, E., Balaji, Dr. V., & Karthikeyan Dhandapani. (2023, December 10). *Advancements in Computer Vision For Automated Fruit Quality Inspection: A Focus on Apple Detection and Grading*. ResearchGate; unknown. https://www.researchgate.net/publication/376519748_Advancements_in_Computer_Vision_For_Automated_Fruit_Quality_Inspection_A_Focus_on_Apple_Detection_and_Grading

L] Zhang, Y., Deng, L., Zhu, H., Wang, W., Ren, Z., Zhou, Q., Lu, S., Sun, S., Zhu, Z., Gorriz, J. M., & Wang, S. (2023). Deep learning in food category recognition. *Information Fusion*, *98*, 101859. https://doi.org/10.1016/j.inffus.2023.101859