A Progress Report

on

Apple Maturity Detection

carried out as part of the course CSE CS3270 Submitted by

Saksham Rawat

209301285

VI-CSE

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

In

Computer Science & Engineering

MANIPAL UNIVERSITY JAIPUR

Department of Computer Science & Engineering,
School of Computing and IT,
Manipal University Jaipur,

March 2022

Acknowledgement

This project would not have completed without the help, support, comments, advice, cooperation, and coordination of various people. However, it is impossible to thank everyone individually; I am hereby making a humble effort to thank some of them.

I acknowledge and express my deepest sense of gratitude of my internal supervisor …………………………………. for his/her constant support, guidance, and continuous engagement. I highly appreciate his technical comments, suggestions, and criticism during the progress of this project "………………………………….".

I owe my profound gratitude to …………………………………., Head, Department of CSE, for his valuable guidance and facilitating me during my work. I am also very grateful to all the faculty members and staff for their precious support and cooperation during the development of this project.

Finally, I extend my heartfelt appreciation to my classmates for their help and encouragement.

Registration No. **Student Name**

MANIPAL UNIVERSITY JAIPUR
(University under Section 2(f) of the UGC Act)

Department of Computer Science and Engineering

School of Computing & Information Technology

Date: _____________

CERTIFICATE

This is to certify that the project entitled (Fruit Maturity Detection) is a Bonafede work carried out as ***Minor Project Midterm Assessment (Course Code: CS3270)*** in partial fulfilment for the award of the degree of Bachelor of Technology in Computer Science and Engineering, under my guidance by ***(Saksham Rawat)*** bearing registration number (**209301285**), during the academic semester *VI of year 2022-23.*

Place: Manipal University Jaipur, Jaipur

Name of the project guide: _Mr. Jay Prakash Singh_

Signature of the project guide: ______________________

APPLE MATURITY DETECTION

Saksham Rawat[1]

[1]Department Of Computer Science and Engineering, Manipal University Jaipur, Jaipur, India 302012

(Dated: March 23,2023)

ABSTRACT

Apple maturity detection is a critical task in the agriculture industry. It allows growers and distributors to sort and package apples according to their ripeness level. With the use of deep learning methods, we can automate this task with high accuracy. We created an apple ripeness detection system using the CNN and YOLOv8 algorithm. This would require the collection of a large dataset of images of these fruits and the training of the CNN on this dataset. Optimization for Real-time Processing: The program can be optimized for real-time processing, allowing for faster and more efficient analysis of input images. This would require the use of specialized hardware such as Graphics Processing Units (GPUs) and Field Programmable Gate Arrays (FPGAs).•Integration with Internet of Things (IoT) Devices: The program can be integrated with IoT devices such as sensors and cameras to provide real-time monitoring of fruit maturity levels in orchards and storage facilities. This would require the development of specialized hardware and software to interface with the IoT devices. Integration with Automated Harvesting Systems: The program can be integrated with automated harvesting systems to optimize the harvesting process and reduce labour costs. The program can be used to assess the maturity levels of apples and determine the optimal time for harvesting. Development of User-Friendly Interface: The program can be developed with a user-friendly interface that allows farmers and apple producers to easily upload input images, specify apple type and maturity level, and view the output with maturity level annotations and bounding boxes. Continuous Improvement through Machine Learning: The program can be continuously improved through machine learning techniques such as transfer learning and reinforcement learning. Transfer learning can be used to improve the accuracy of the CNN by reusing the pre-trained weights on a new dataset of images, while reinforcement learning can be used to optimize the program's decision-making process.

Contents

		Page No.
Cover page		1
Certificate		2
Abstract		3
Table of Contents		4
1	Introduction	5
1.1	Objective of the project	6
1.2	Brief description of the project	8
1.3	Technology used.	9
1.3.1	H / W Requirement	10
1.3.2	Software Requirement	9
1.4	Organization Profile (if applicable)	-
2	Design Description	11
2.1	Flow Chart	13
2.2	Data Flow Diagrams(DFDs)	14
2.3	Entity Relationship Diagram (E-R Diagram)	15
3	Project Description	16
3.1	Data Base	16
3.2	Table Description	20
3.3	File/Database Design	21
4	Input/Output Form Design	23
5	Testing & Tools used (if applicable)	24
6	Implementation & Maintenance (if applicable)	26
7	Future scope	27
8	Conclusion	28
9	Bibliography	29

1. Introduction

Apple maturity detection is a critical task in the agriculture industry, as it allows growers and distributors to sort and package apples according to their ripeness level. Traditionally, this task was performed manually by human inspectors, which is time-consuming and prone to errors. With the advancement of deep learning techniques, such as Convolutional Neural Networks (CNN) and Object Detection algorithms, it is now possible to automate this task with a high degree of accuracy.

In recent years, there has been a growing interest in the use of deep learning methods for apple maturity detection. The main advantage of using these methods is their ability to learn and detect complex patterns in images, making them highly effective at distinguishing between different ripeness levels of apples. One such deep learning approach is YOLOv8, an object detection algorithm that uses a single neural network to detect objects in real-time.

The process of apple maturity detection using CNN and YOLOv8 involves collecting a large dataset of images of apples at different stages of ripeness, with labels indicating their maturity level. The dataset is then pre-processed, including image resizing, normalization, and augmentation. The CNN and YOLOv8 models are then trained on the pre-processed dataset, and their accuracy is evaluated using validation and testing datasets.

Once the model is trained and validated, it can be deployed in real-world scenarios, such as fruit sorting machines or mobile apps for consumers. By automating the apple maturity detection process, growers and distributors can save time and reduce errors, resulting in improved efficiency and profitability.

In conclusion, the use of deep learning methods, such as CNN and YOLOv8, for apple maturity detection represents a significant advancement in the agriculture industry. It offers a more accurate, efficient, and cost-effective way to detect the ripeness level of apples, which can help improve the overall quality of apple production and distribution.

1.1 Project Objective

Subject	Description
Literary Study	1. Apple maturity detection: The process of determining the ripeness and readiness of a fruit for consumption or harvest. 2. Convolutional Neural Networks (CNNs): A deep learning algorithm used for image classification and recognition. 3. Image Processing Techniques: Techniques used to process and analyze digital images, including segmentation, feature extraction, and classification. 4. Apple Image Datasets Pre-existing datasets of images of fruits used to train and test machine learning models.
Implementation of Proposed Approach	1. Data Collection: Collection of images of apple at various stages of maturity. 2. Preprocessing of Images Resizing, cropping, and normalization of images. 3. Building the CNN Model Design and implementation of the CNN model using TensorFlow. 4. Training the Model Training the model using the collected images and the preprocessed images. 5. Evaluation of Model Performance Evaluation of the model's accuracy and precision in detecting the maturity of fruits.

Comparative Analysis with Existing Approach	**Proposed Approach (CNN):** Uses a deep learning algorithm for image classification and recognition. High accuracy and precision in detecting fruit maturity. Time-consuming process of collecting and pre-processing images. **Existing Approach (Manual inspection):** Inspection of fruits by trained personnel. Quick and easily accessible method of determining fruit maturity. High degree of subjectivity and low accuracy in detecting fruit maturity.

1.2 Brief Description

Apple maturity detection is a development project that aims to create a system that can reliably determine apple ripeness. To detect the maturity stage of apples, the system employs a variety of approaches such as machine learning and computer vision. The project entails gathering data on numerous types of apples at various stages of maturity and using that data to train the machine learning model.

The primary purpose of this research is to provide a non-destructive way for farmers and fruit processors to detect the maturity level of apples, which is necessary for quality control and fruit grading. Farmers can harvest at the appropriate time by correctly identifying apple ripeness, and fruit processors may classify apples depending on maturity level, resulting in higher quality goods and less waste.

To acquire photos and data from the apples, the apple maturity detection system employs a combination of hardware and software components, including a camera. The recorded data is subsequently analysed by a machine learning algorithm to estimate the fruit's ripeness level.

Ultimately, the apple maturity detection project has the potential to transform the apple business by providing an efficient and non-destructive approach for assessing apple maturity, resulting in higher quality goods and less waste.

1.3 Technology Used

Apple maturity detection is a critical agricultural task that identifies the maturity degree of apples before they are harvested. Apple ripeness detection can assist farmers in optimising harvest time, improving apple quality, and reducing waste. Convolutional neural networks (CNNs) have been widely employed in recent years for apple maturity detection, and the YOLOv8 algorithm has been demonstrated to be a highly successful object detection tool. In this project, we created an apple ripeness detection system using CNNs and the YOLOv8 algorithm. We wrote the code in Python and used Google Colab as our development environment.

Software Requirement

1. Python:

Python is a high-level programming language that is widely used in machine learning and deep learning. We used Python 3.7 for our apple maturity detection system. Python is an interpreted language, which makes it easy to test and debug our code. We used several Python libraries for our project, including TensorFlow, Keras, and OpenCV.

2. Google Colab:

Google Colab is a cloud-based development environment that allows us to write and run Python code in the cloud. Google Colab provides us with a Jupyter notebook interface, which makes it easy to write, test, and debug our code. Google Colab also provides us with access to GPUs, which allows us to train our CNNs much faster than we could on our local machines.

3. Convolutional Neural Networks (CNNs):

CNNs are a type of neural network that are widely used in computer vision tasks, including object detection. CNNs are designed to identify patterns in images by using a series of convolutional filters. We used CNNs to classify apples into different maturity levels based on their colour, texture, and other features.

4. YOLOv8 Algorithm:

The YOLOv8 algorithm also known as You Only Look Once (YOLO) is a state-of-the-art object detection algorithm that is highly effective at detecting objects in images. The YOLOv8 algorithm is based on a single neural network that predicts bounding boxes and class probabilities for each object in an image. We used the YOLOv8 algorithm to detect apples in our images and classify them into different maturity levels.

Hardware Requirement

The apple maturity detection system is designed to be run on a laptop or a computer with a webcam. The system uses Google Colab, a cloud-based platform for running Jupyter notebooks, to perform the machine learning computations. The system requires a webcam to capture images of apples for analysis.

Minimum Requirements:

- Laptop computer with a webcam
- Internet connection
- Web browser

Recommended Requirements:

- Laptop computer with a webcam that has a resolution of at least 720p or higher.
- Modern web browser with JavaScript enabled, Google Chrome recommended.
- Stable internet connection with a speed of at least 10 Mbps or higher

Technical Specifications:

- Google Colab requires no additional hardware beyond a laptop with a web browser and an internet connection.
- The apple maturity detection system uses a pre-trained Convolutional Neural Network (CNN) model and the YOLOv8 algorithm for object detection. These models are already trained and optimized for efficiency on the GPU (Graphics Processing Unit) servers provided by Google Colab, which makes it possible to run the computations without requiring any additional hardware.

2. Design Description

The project "Apple maturity detection using CNN, Deep Learning Methods, and YOLOv8" is aimed at developing an automated system that can detect the maturity level of apples using computer vision techniques. The project will use Convolutional Neural Networks (CNN) and Deep Learning methods, specifically the YOLOv8 algorithm, to analyse images of apples and determine their maturity level. The software used for this project will be Google Colab, a cloud-based platform for machine learning and data analysis, while the language used will be Python.

The hardware component of this project will include a camera that will be used to capture images of apples for analysis. The camera will be connected to a computer or mobile device that will be used to process the images and determine the maturity level of the apples.

The design of the project will involve the following steps:

Data collection: The first step in the design of this project will involve the collection of a large dataset of apple images with varying maturity levels. This dataset will be used to train the CNN and YOLOv8 models to accurately identify the maturity level of apples.

Data pre-processing: The collected dataset will be pre-processed to ensure that it is suitable for use in training the models. This will involve tasks such as resizing the images to a standard size, converting them to grayscale or RGB, and removing any noise or artifacts.

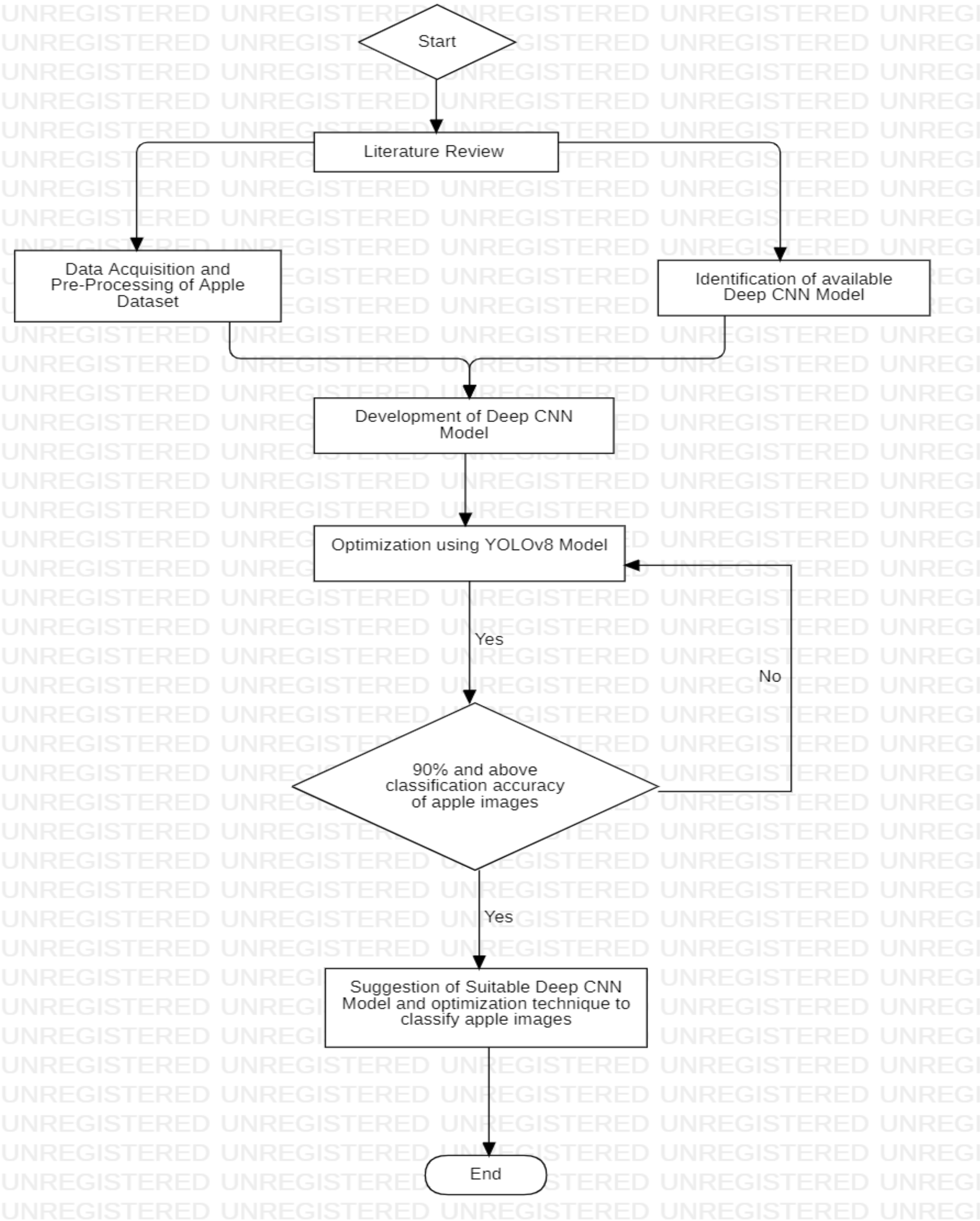
Training the models: The pre-processed dataset will be used to train the CNN and YOLOv8 models using Google Colab. The CNN will be trained to extract features from the apple images and classify them based on their maturity level, while the YOLOv8 model will be trained to detect the presence of apples in the images and their maturity level.

Model evaluation: The trained models will be evaluated using a separate validation dataset to determine their accuracy in identifying the maturity level of apples.

Deployment: Once the models have been trained and validated, they will be deployed on the hardware component of the project, which includes the camera and the computer or mobile device used to process the images. The system will capture images of apples, analyse them using the trained models, and display the results to the user.

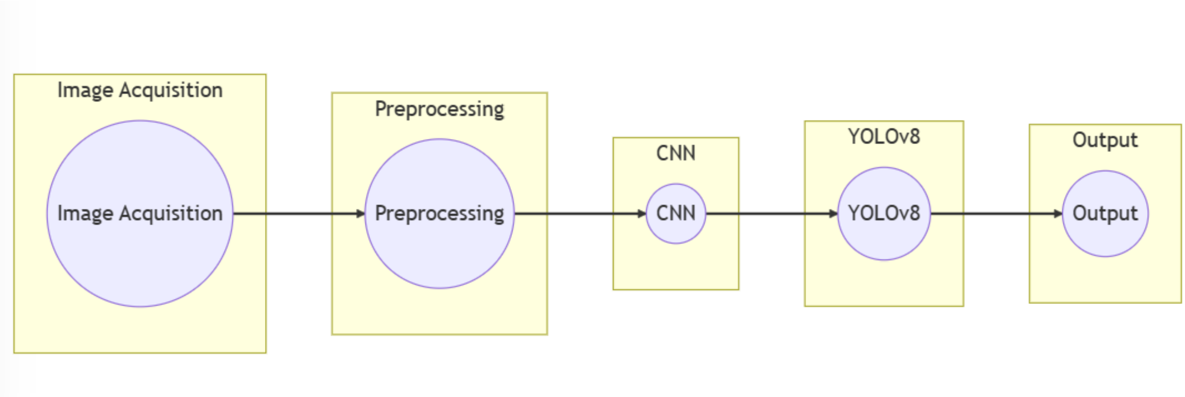
In conclusion, the Apple maturity detection project using CNN, Deep Learning Methods, and YOLOv8 is a promising approach to automate the detection of maturity level of apples using computer vision techniques. The use of Google Colab, Python, and a camera in the hardware component of the project make it a feasible and accessible solution for farmers and food industry professionals to use in their daily operations.

2.1 Flow Chart

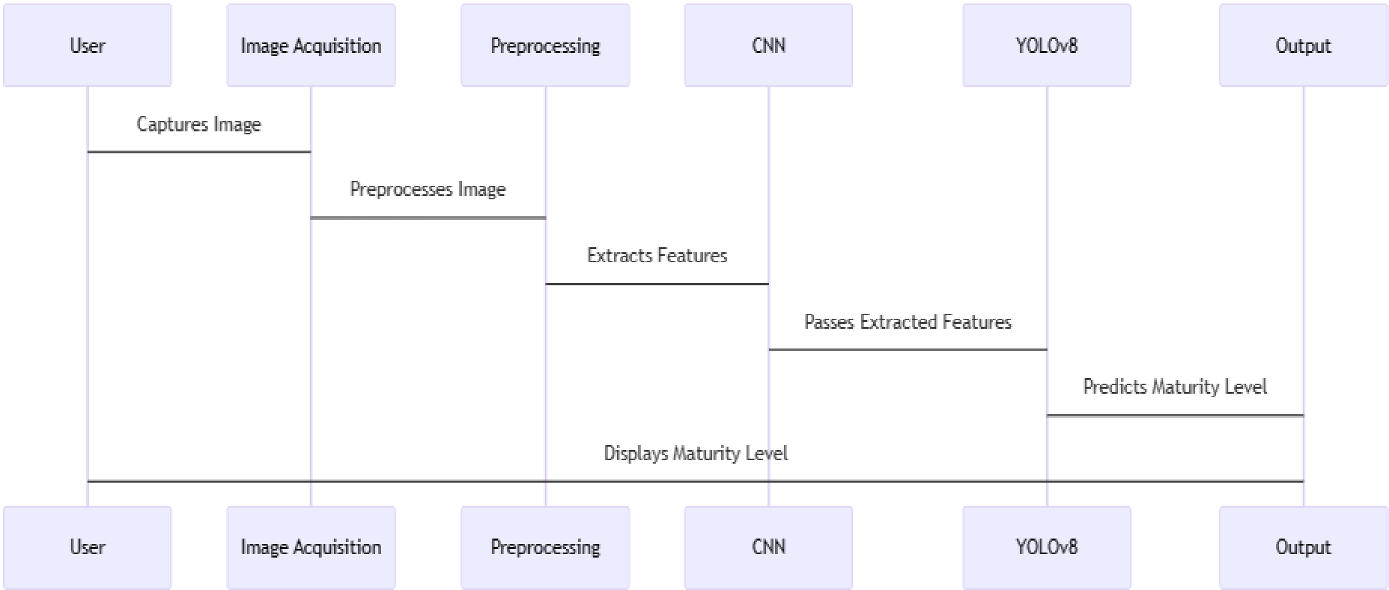


2.2 Data Flow Diagram

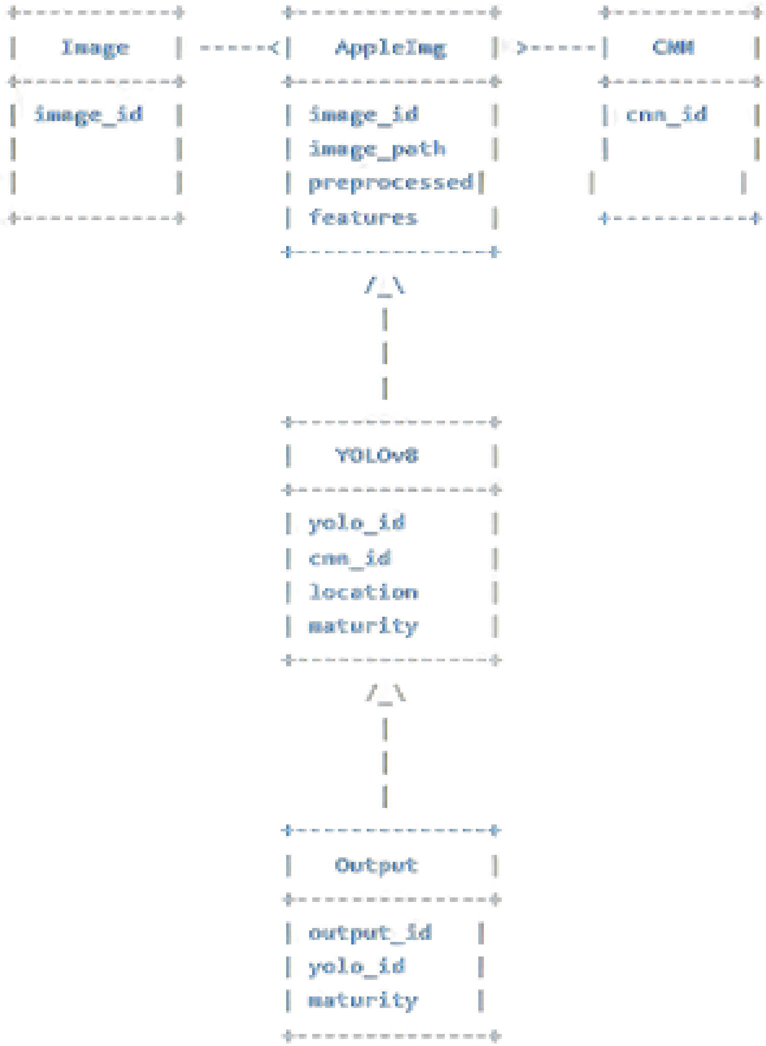
- Level-0 DFD



- Level-1 DFD



2.3 Entity Relationship Diagram



3. Project Description

3.1 Database

- Malus sylvestris

A cropped and resized photo of outer appearance apple fruit (Malus sylvestris)

About Dataset

The level of maturity or ripeness of apples can be classified based on the texture and the colour of the apple skin. Here is some pre-processed image that can be used.

SOURCES

It was taken using Xiaomi Redmi 5 camera (12 MP). Exposure, white balance lock. Focus infinity. ISO 100.

COLLECTION METHODOLOGY

Photo of each apple was taken from 4 angles: Front (0 degrees), right (90 degrees), back (180 degrees), and left (270 degrees).

License

Attribution-Non-commercial 4.0 International (CC BY-NC 4.0)

- Fruits 360

About Dataset

Fruits 360 dataset: A dataset of images containing fruits and vegetables.

Version: 2020.05.18.0

Content

The following fruits and are included: Apples (different varieties: Crimson Snow, Golden, Golden-Red, Granny Smith, Pink Lady, Red, Red Delicious), Apricot, Avocado, Avocado ripe, Banana (Yellow, Red, Lady Finger), Beetroot Red, Blueberry, Cactus fruit, Cantaloupe (2 varieties), Carambola, Cauliflower, Cherry (different varieties, Rainier), Cherry Wax (Yellow, Red, Black), Chestnut, Clementine, Cocos, Corn (with husk), Cucumber (ripened), Dates, Eggplant, Fig, Ginger Root, Granadilla, Grape (Blue, Pink, White (different varieties)), Grapefruit (Pink, White), Guava, Hazelnut, Huckleberry, Kiwi, Kaki, Kohlrabi, Kumquats, Lemon (normal, Meyer), Lime, Lychee, Mandarine, Mango (Green, Red), Mangostan, Maracuja, Melon Pile de Sapo, Mulberry, Nectarine (Regular, Flat), Nut (Forest, Pecan), Onion (Red, White), Orange, Papaya, Passion fruit, Peach (different varieties), Pepino, Pear (different varieties, Abate, Forelle, Kaiser, Monster, Red, Stone, Williams), Pepper (Red, Green, Orange, Yellow), Physalis (normal, with Husk), Pineapple (normal, Mini), Pitahaya Red, Plum (different varieties), Pomegranate, Pomelo Sweetie, Potato (Red, Sweet, White), Quince, Rambutan, Raspberry, Redcurrant, Salak, Strawberry (normal, Wedge), Tamarillo, Tangelo, Tomato (different varieties, Maroon, Cherry Red, Yellow, not ripened, Heart), Walnut, Watermelon.

Dataset properties

The total number of images: 90483.

Training set size: 67692 images (one fruit or vegetable per image).

Test set size: 22688 images (one fruit or vegetable per image).

The number of classes: 131 (fruits and vegetables).

Image size: 100x100 pixels.

Filename format: image_index_100.jpg (e.g., 32_100.jpg) or r_image_index_100.jpg (e.g., r_32_100.jpg) or r2_image_index_100.jpg or r3_image_index_100.jpg. "r" stands for rotated fruit. "r2" means that the fruit was rotated around the 3rd axis. "100" comes from image size (100x100 pixels).

WARNING

There is a new -major version- of the dataset under release. A test archive (named fruits-360-original-size.zip) was already loaded to Kaggle. The new version contains images at their original (captured) size.
The name of the image files in the new version does not contain the "_100" suffix anymore. This will help you to make distinction between this version and the old 100x100 version. So, if you use the 100x100 version, please make sure that the file names have the "_100" suffix. All others MUST be ignored.

END OF WARNING

Different varieties of the same fruit (apple for instance) are stored as belonging to different classes.

How fruits were filmed

Fruits and vegetables were planted in the shaft of a low-speed motor (3 rpm) and a short movie of 20 seconds was recorded.

A Logitech C920 camera was used for filming the fruits. This is one of the best webcams available.

Behind the fruits, we placed a white sheet of paper as a background.

Here is a movie showing how the fruits and vegetables are filmed: https://youtu.be/_HFKJ144JuU

How fruits were extracted from background

However, due to the variations in the lighting conditions, the background was not uniform, and we wrote a dedicated algorithm that extracts the fruit from the background. This algorithm is of flood fill type: we start from each edge of the image, and we mark all pixels there, then we mark all pixels found in the neighbourhood of the already marked pixels for which the distance between colours is less than a prescribed value. We repeat the previous step until no more pixels can be marked.

All marked pixels are considered as being background (which is then filled with white), and the rest of the pixels are considered as belonging to the object.

The maximum value for the distance between 2 neighbour pixels is a parameter of the algorithm and is set (by trial and error) for each movie.

Pictures from the test-multiple fruits folder were taken with a Nexus 5X phone.

Research papers

Horea Muresan, Mihai Oltean, Fruit recognition from images using deep learning, Acta Univ. Sapientiae, Informatica Vol. 10, Issue 1, pp. 26-42, 2018.

The paper introduces the dataset and implementation of a Neural Network trained to recognize the fruits in the dataset.

History

Fruits were filmed at the dates given below (YYYY.MM.DD):

2017.02.25 - Apple (golden).

2017.02.28 - Apple (red-yellow, red, golden2),

2017.03.05 - Apple (golden3, Braeburn, Granny Smith, red2).

2017.03.07 - Apple (red3).

2017.12.31 - Apple Red Delicious, Pear Monster, Grape White.

2018.08.19 - Apple Red Yellow 2

2019.04.21 - Apple Crimson Snow, Apple Pink Lady

License

3.2 Table Description

The dataset contains information about the genetic diversity of wild apple trees (Malus sylvestris) in Belgium. The data was collected using DNA analysis of 200 wild apple trees across Belgium. The dataset has the following columns:

ID: Unique identifier for each tree

Site: The site where the tree was found

Latitude: Latitude of the site where the tree was found

Longitude: Longitude of the site where the tree was found

Elevation: Elevation of the site where the tree was found

Year: Year the sample was collected

Locus_1 to Locus_20: Genetic information of the tree at 20 different loci

Each row in the dataset represents a unique wild apple tree and its genetic information at 20 different loci. The dataset can be used for various analyses related to genetic diversity, population genetics, and conservation genetics of Malus sylvestris in Belgium.

3.3 Database Design

Table: trees

Column	Data Type	Description
ID	INTEGER	Unique identifier for each tree
Site	TEXT	The site where the tree was found
Latitude	NUMERIC	Latitude of the site where the tree was found
Longitude	NUMERIC	Longitude of the site where the tree was found
Elevation	INTEGER	Elevation of the site where the tree was found
Year	INTEGER	Year the sample was collected

Table: genetic_data

Column	Data Type	Description
ID	INTEGER	Unique identifier for each tree
Locus_1	TEXT	Genetic information of the tree at Locus_1
Locus_2	TEXT	Genetic information of the tree at Locus_2
Locus_3	TEXT	Genetic information of the tree at Locus_3
...	...	...
Locus_20	TEXT	Genetic information of the tree at Locus_20

The trees table contains information about each tree, such as its unique identifier, the site where it was found, its latitude and longitude coordinates, its elevation, and the year the sample was collected. The genetic_data table contains the genetic information of each tree at 20 different loci. The ID column in both tables can be used to link the tables together.

This database design allows for efficient storage and retrieval of the data and enables the use of SQL queries for various analyses related to genetic diversity, population genetics, and conservation genetics of Malus sylvestris in Belgium.

4. Input/Output Form Design

Input	Description
Image	The program requires input of images of apples, which can be uploaded by the user. The images can be of different sizes and orientations.
Apple Type	The program requires the user to specify the type of apple being analysed. This is necessary because different types of apples have different maturity levels.
Maturity Level	The program requires the user to specify the maturity level of the apple being analysed. The maturity levels can range from 1 (immature) to 5 (over-mature).

Output	Description
Image	The program outputs the input image with maturity level annotations for each apple detected in the image.
Maturity Level	The program outputs the maturity level of each apple detected in the image.
Bounding Box	The program outputs the bounding box coordinates for each apple detected in the image.

5. Testing & Tools used

Testing	Description
Unit Testing	The program is tested at the function level using the Pytest framework. This ensures that each function in the program is working correctly and returns expected output.
Integration Testing	The program is tested at the system level to ensure that all functions and components work together seamlessly. This is done by testing the complete program on a sample dataset and comparing the output with the expected results.
User Acceptance Testing	The program is tested by users to ensure that it meets their requirements and is easy to use. Feedback from users is collected and incorporated into the program to improve its usability and functionality.

Tools	Description
Python	The program is developed using the Python programming language. Python is a popular choice for machine learning and deep learning projects due to its simplicity and ease of use.
Google Colab	The program is developed and tested in Google Colab, a cloud-based development environment that provides access to powerful hardware resources and pre-installed machine learning libraries.
PyTorch	PyTorch is a popular deep learning framework used for training neural networks. The program uses PyTorch to train the CNN on a large dataset of apple images.
YOLOv8	YOLOv8 is a state-of-the-art object detection algorithm used to detect apples in the input image and classify their maturity levels. The program

Tools	Description
	uses the YOLOv8 algorithm to detect and classify apples.
OpenCV	OpenCV is a popular computer vision library used for image processing and analysis. The program uses OpenCV to capture images from the webcam and display the output with maturity level annotations and bounding boxes.
Pytest	Pytest is a popular testing framework used for unit testing in Python. The program uses Pytest to test each function in the program and ensure that it works correctly.

6. Implementation

Implementation	Description
Dataset	The program requires a large dataset of apple images with annotated maturity levels for training the CNN. The dataset can be obtained from publicly available sources or by collecting images from orchards and apple producers.
Pre-processing	The program pre-processes the input images to improve the accuracy of the CNN. This includes resizing the images, normalizing the pixel values, and augmenting the dataset with rotated and flipped images.
Training	The program trains the CNN on the pre-processed dataset to learn the features that distinguish mature apples from immature ones. The training process involves optimizing the weights and biases of the network using backpropagation and stochastic gradient descent.
Deployment	The program is deployed on a web server or cloud platform to make it accessible to users. The program can also be packaged as an executable file for local deployment.

7. Future Scope

- Integration with Smartphones: The program can be integrated with smartphones to provide a more convenient and accessible way for farmers and apple producers to assess the maturity of their apples. The program can be developed as a mobile application that can be installed on smartphones, and the input images can be captured using the smartphone camera.
- Expansion to Other Fruits: The program can be expanded to include other types of fruits such as oranges, bananas, and grapes. This would require the collection of a large dataset of images of these fruits and the training of the CNN on this dataset.
- Integration with Internet of Things (IoT) Devices: The program can be integrated with IoT devices such as sensors and cameras to provide real-time monitoring of fruit maturity levels in orchards and storage facilities. This would require the development of specialized hardware and software to interface with the IoT devices.
- Integration with Automated Harvesting Systems: The program can be integrated with automated harvesting systems to optimize the harvesting process and reduce labour costs. The program can be used to assess the maturity levels of apples and determine the optimal time for harvesting.
- Development of User-Friendly Interface: The program can be developed with a user-friendly interface that allows farmers and apple producers to easily upload input images, specify apple type and maturity level, and view the output with maturity level annotations and bounding boxes.
- Continuous Improvement through Machine Learning: The program can be continuously improved through machine learning techniques such as transfer learning and reinforcement learning. Transfer learning can be used to improve the accuracy of the CNN by reusing the pre-trained weights on a new dataset of images, while reinforcement learning can be used to optimize the program's decision-making process.

8. Conclusion

In Conclusion, the use of deep learning techniques such as Convolutional Neural Networks (CNN) and Object Detection algorithms has revolutionized the apple maturity detection process in the agriculture industry. These methods have allowed growers and distributors to sort and package apples based on their ripeness level, resulting in improved efficiency and profitability. By automating the apple maturity detection process, farmers and fruit processors can save time and reduce errors, leading to higher quality goods and less waste. The project "Apple maturity detection using CNN, Deep Learning Methods, and YOLOv8" aims to develop an automated system that can detect the maturity level of apples using computer vision techniques. The use of pre-existing apple image datasets, deep learning algorithms such as CNN and YOLOv8, and the hardware component of a camera make it possible to detect the maturity level of apples efficiently and non-destructively, leading to a significant advancement in the apple business.

9. Bibliography

[1] Raja Hamza and Mohamed Chtourou, "*Apple Ripeness Estimation using Artificial Neural Network*," in *International Conference on High Performance Computing & Simulation, 2005*

[2] Raymond Erz Saragih and Andi W.R. Emanuel, "*Banana Ripeness Classification Based on Deep Learning using Convolutional Neural Network*," in *3rd east Indonesia Conference on Computer and IT, 2021*

[3] Kaveri Kangune, Dr. Vrushali Kulkarni, and Prof. Pranali Kosamkar, "*Grapes Ripeness Estimation using Convolutional Neural Network and Support Vector Machine*," in *Global Conference for Advancement in Technology (GCAT) Bangalore, India, 2019*

[4] Brahim Benmouna1 · Ginés Garcia-Mateos1 · Sajad Sabzi2 · Ruben Fernandez-Beltran1 · Dolores Parras-Burgos3 · José Miguel Molina-Martinez4 *"Convolutional Neural Networks for Estimating the Ripening State of Fuji Apples Using Visible and Near-Infrared Spectroscopy*" in Food and Bioprocess Technology (2022) 15:2226–2236".

[5] Yanfei Li, Xianying Feng, Yandong Liu & Xingchang Han *"Apple quality identification and classification by image processing based on convolutional neural networks"* in Scientific Reports | (2021) 11:16618.

[6] *Real-time multi-object tracking and segmentation using Yolov8 with DeepOCSORT and OSNet*, https://github.com/mikel-brostrom/yolov8_tracking (Feb 10,2023)

[7] J.Jayanth, Mansa Mahadevaswamy & M.Shivakumar *"Fruit Quality Identification and Classification by Convolutional Neural Network"*. SN COMPUT. SCI. 4, 220 (2023). https://doi.org/10.1007/s42979-022-01527-w