

IDENTICAL FRUITS DETECTION USING FASTER R-CNN AND YOLO

BY

MD. ASHRAFUR RAHMAN BHUIYAN

ID: 191-15-2433

AND

AFRIDI HASAN SHADHIN

ID: 191-15-2699

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Tania Khatun

Assistant Professor

Department of CSE

Daffodil International University

Co-Supervised By

Md. Sabab Zulfiker

Lecturer

Department of CSE

Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

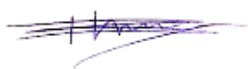
DHAKA, BANGLADESH

JANUARY 2023

APPROVAL

This Project/internship titled “**Identical Fruits Detection using Faster R-CNN And YOLO**”, submitted by Md. Ashrafur Rahman Bhuiyan, ID No: 191-15-2433 and Afridi Hasan Shadhin, ID No: 191-15-2699 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 23th January 2023.

BOARD OF EXAMINERS



Chairman

Dr. Touhid Bhuiyan

Professor and Head

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Dr. Md. Tarek Habib

Associate Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Tapasy Rabeya

Senior Lecturer

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

External Examiner



Dr. Dewan Md Farid

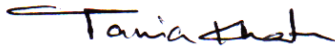
Professor

Department of Computer Science and Engineering
United International University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Tania Khatun, Assistant Professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

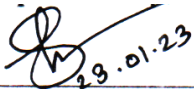
Supervised by:



Tania Khatun

Assistant Professor
Department of CSE
Daffodil International University

Co-Supervised by:



Md. Sabab Zulfiker


Sr. Lecturer
Department of CSE
Daffodil International University

Submitted by:



Md. Ashrafur Rahman Bhuiyan

ID: -191-15-2433
Department of CSE
Daffodil International University



Afridi Hasan Shadhin

ID: -191-15-2699
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible for us to complete the final year project/internship successfully.

We are really grateful and wish our profound indebtedness to **Tania Khatun, Assistant Professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “Data Mining, Machine Learning (ML), Internet of Things (IoT), Artificial Intelligence (AI), Image Processing, Deep learning, Natural language processing (NLP)” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. Touhid Bhuiyan**, Professor and Head, **Tania Khatun**, Assistant Professor and **MD. Sabab Zulfiker**, Sr. Lecturer, Department of CSE, for his kind help to finish our project and also to other faculty members and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

Fruits with the same color, shape, and size are called identical fruits. There are a few tiny distinctions between the two, despite their identical appearance. In an effort to discriminate between identical fruits, a method has been created. The "big challenge" is to create object-detection technology on par with human ability. Deep learning can be a revolutionary solution to this problem. Deep learning makes it easier and quicker to make sense of massive datasets and draw conclusions from them. We used deep learning algorithms to detect the difference between identical fruits. In this research, we used YOLO v5 and the Faster R-CNN algorithm. We conducted the study using data from four categories: apple, pear, orange, and lemon. For the implementation of this process, we collect image data from fruit stores and label them according to our algorithms' requirements. We have augmented the image and made a data set of 1680 images for the methods to gain more accuracy. In our quest to identify fruits with comparable properties, we have explored both the Faster R-CNN and Yolo v5 algorithms. Yolo v5 obtained 97% accuracy compared to 91% for Faster R-CNN. Among the two algorithms, Yolo v5 produced the highest success rate, 97%. Our research will contribute to the advancement of automated agricultural technology. It is also useful in the supermarket to identify identical fruits. Most importantly it is a critical step toward automating processes such as automated harvesting and packaging etc.

Keywords: Faster R-CNN, YOLO v5, Identical fruits, Detection, Automated farming.

TABLE OF CONTENTS

CONTENTS	PAGE
Approval	i
Declaration	ii
Acknowledgement	iii
Abstract	iv
List of Figures	viii
List of Tables	ix
CHAPTER	
CHAPTER 1: INTRODUCTION	01 - 06
1.1 Introduction	1
1.2 Motivation	2
1.3 Rationale of the Study	4
1.4 Research Questions	4
1.5 Expected Output	5
1.6 Report Layout	5
CHAPTER 2: BACKGROUND	07 - 14
2.1 Introduction	7
2.2 Related Works	8

2.3 Comparative Analysis	12
2.4 Scope of the Problem	13
2.5 Challenges	13
2.6 Summary	14
CHAPTER 3: RESEARCH METHODOLOGY	15 - 38
3.1 Introduction	15
3.2 Data Collection and Preprocessing	16
3.3 Statistical Analysis	22
3.4 Proposed Methodology	27
3.5 Classification Model	29
3.6 Implementation Requirements	35
CHAPTER 4: EXPERIMENTAL RESULTS AND DISCUSSION	39 - 44
4.1 Experimental Setup	39
4.2 Experimental Results & Analysis	39
4.3 Discussion	44

CHAPTER 5: IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY	45 - 47
5.1 Impact on Society	45
5.2 Impact on Environment	45
5.3 Ethical Aspects	46
5.4 Sustainability Plan	46
CHAPTER 6: SUMMARY, CONCLUSION, RECOMMENDATION AND IMPLICATION FOR FUTURE RESEARCH	48 - 49
6.1 Summary of the Study	48
6.2 Conclusions	48
6.3 Implication for Further Study	49
APPENDIX	49
REFERENCES	50 - 52
PLAGIARISM REPORT	53

LIST OF FIGURES

FIGURES	PAGE NO
Figure 2.1: Agricultural growth rate	7
Figure 3.1: Apple category	16
Figure 3.2: Pear category	17
Figure 3.3: Orange category	17
Figure 3.4: Lemon category	17
Figure 3.5: Label data visualization	19
Figure 3.6: Augmented data visualization	21
Figure 3.7: Average precision for Apple	22
Figure 3.8: Average precision for Lemon and Orange	22
Figure 3.9: Average precision for Pear	23
Figure 3.10: AP & AP50 visualization for Faster R-CNN	24
Figure 3.11: Precision for YOLO	24
Figure 3.12: Recall measurement visualization	25
Figure 3.13: False Negative ratio graph	26
Figure 3.14: Proposed model for research work	27
Figure 3.15: The Improved Faster R-CNN Network.	29
Figure 3.16: A look at the framework of Faster R-CNN	32
Figure 3.17: object detection architecture of YOLO v4	34
Figure 4.1: Process losses	41
Figure 4.2: Anchors visualization	42
Figure 4.3: Total time-lap	42

LIST OF TABLES

TABLES	PAGE NO
Table 3.1: Quantity explanation of used data set	17
Table 3.2: Data splitting configuration	17
Table 3.3: Data extenuation using augmentation	19
Table 3.4: Data configuration	27
Table 3.5: Research instrument explanation	34
Table 4.1: Algorithm accuracy and time	39
Table 4.2: Evaluate metrics accuracy for YOLO v5	39
Table 4.3: Average precision for individual classes in Faster R-CNN	40
Table 4.4: Evaluate metrics accuracy in Faster R-CNN	40

CHAPTER 1

Introduction

1.1 Introduction

In recent years, the field of study known as "agricultural robotics" has expanded quickly. In a moment, we'll take a closer look at how machine learning (ML) and AI-powered technologies are hastening the advent of fully automated farming. Automation of simpler, more routine activities is the main goal of agricultural technology. It is also known as "precision agriculture" or "smart farming", a farming technique that makes use of contemporary software and hardware to measure, track, and manage data in order to increase productivity, boost crop yields, and cut expenses.

The following are some of the most prominent technological advancements that are now being applied to farms. Sensors, drones, robots, and other technologies are used in automated farming to collect and analyze data, find the best levels of irrigation and fertilization, and check on the health of the crops. There are robots that plant seeds and pull weeds, machines that harvest crops, and drones that watch the conditions in the field. It helps farmers save money on labor costs and increase crop yields by making farming more accurate and efficient.

Precision agriculture has made significant progress in the area of fruit identification, which has several applications including yield estimation and robotic fruit and vegetable harvesting. Most of the time, machine vision and AI are used in farming to figure out what kinds of fruits are in a field. This technology can tell what kind of fruit it is based on its size, shape, color, and other features. The system can be used to find fruit in a certain spot or to look at a whole field or orchard. The system can also be used to find out if the fruit has any diseases.

Identical fruits and vegetables are those that have the same color, shape, and size. Despite the fact that they look exactly the same, there are a few subtle differences that may be detected between the two. In an effort to distinguish between identical fruits, we have developed a method. The "great challenge" is to develop technology that is as good at

object detection as a human being. It is important for both professional and academic research as well as for commercial applications. A framework like this can be used to develop a lot of beneficial programs.

To discover fruits that look the same, this can be utilized in automated farming and fruit harvesting. Automated fruit picking is accomplished via a self-operating system. As a result, less money must be spent on compensating employees for physical labor, and the fruits can be gathered swiftly without fear of wasting the important crop. Someone who owns a farm with various fruit trees that all produce the same type of fruit will benefit from this technology. To keep his farm functioning well, he only requires a small amount of labor and materials. Automated harvesting will improve and become simpler to obtain with the use of this technology.

1.2 Motivation

Bangladesh, like many other developing economies, has relied heavily on agriculture. Around half of the population is employed in agriculture, while roughly 70% of the population relies on agriculture in some way for subsistence. Agriculture is the primary or only source of income and employment for a large percentage of the world's impoverished. Employment in agro-based services and rural companies is an indirect reflection of reliance on agriculture. However, agricultural growth has averaged 3.2% annually over the past three decades. [33]

Identifying the species and variety of produce is essential for the cashier to set the appropriate price at the supermarket checkout. The use of barcodes has helped with this issue for packed goods, but when customers get to choose their own item, which is impossible, the item must be weighed. Assigning codes to similar types of fruit is a frequent option, but it has drawbacks including requiring memorized lists, which can lead to pricing mistakes.

One of the many valuable additions to this list in the twenty-first century is the ability to identify identical fruits. While the underlying principle is similar to that used in fruit

detection systems, there may be complications when trying to identify similar types of data. Just like the list-making problem, we can now solve the item-counting problem mechanically.

All of this has inspired us to finally take action in this area. The purpose of our system is to automate the process of identifying duplicate fruits, which will lead to more precise inventory counts, price estimates, and the timely harvesting of ripe fruit from orchards. And in addition, it will aid kids in learning the difference between similar-looking fruits.

What has happened has inspired us to take action in this area. Our goal is to develop a system that can identify duplicate fruits automatically, which will improve fruit inventory management, pricing setting, and harvesting. Along with that, it will aid the learning process by allowing kids to distinguish between fruits that look similar.

1.2.1 Aim & Objective

The application of image processing methods to agricultural problems is the focus of this thesis. The principal objectives are as follows:

- To identify different types and varieties of fruit based on photographs.
- To rapidly and precisely identify vegetables at grocery stores.
- To identify and confirm the ingredients used in dishes at restaurants.
- To guarantee that only the appropriate fruits are contained in packages in food processing and packaging facilities.
- Sorting and immediately identifying fruit in orchards.
- Accurately identifying and pricing products at farmer's markets.
- To assist in educating youngsters about various fruits in schools and daycare facilities.
- Quickly and precisely determine the country of origin of imported fruits at airports.
- To precisely weigh and price food in supermarkets.
- Last but not least, the detection of identical fruits provides farmers with more knowledge about the growth and health of their crops, enabling them to make wise decisions regarding crop management and land management.

1.3 Rationale of the Study

In agriculture, our research is very important. In this study, we are looking for the best way to tell if two fruits are the same, which will help with automated harvesting in supermarkets. Using this research, we can make an app that can find fruits and tell people about them, as well as help kids learn the names of the fruits.

The application of image processing methods to agricultural problems is the focus of this thesis. The principal objectives are as follows:

- To identify different types and varieties of fruit based on photographs.
- To rapidly and precisely identify vegetables at grocery stores.
- To identify and confirm the ingredients used in dishes at restaurants.
- To guarantee that only the appropriate fruits are contained in packages in food processing and packaging facilities.
- Sorting and immediately identifying fruit in orchards.
- Accurately identifying and pricing products at farmer's markets.
- To assist in educating youngsters about various fruits in schools and daycare facilities.
- Quickly and precisely determine the country of origin of imported fruits at airports.
- To precisely weigh and price food in supermarkets.
- Last but not least, the detection of identical fruits provides farmers with more knowledge about the growth and health of their crops, enabling them to make wise decisions regarding crop management and land management.

1.4 Research Questions

The questions for the research are as follows:

- How can we get our hands on any fruit data for this research?
- How do we preprocess our data for this research?

- Which methodology will perform better in this research?
- How does the proposed methodology improve the detection of identical fruits?
- To what end will this study contribute to the development of automated farming techniques?

1.5 Expected Output

The findings of this study are anticipated to improve the precision with which identical fruits may be identified. which can recognize individual fruits as well as other fruits that look the same.

1.6 Report Layout

We introduced our research endeavor and briefly outlined its scope in chapter 1. We discuss why identical fruit detection, in particular, is so important in the present day. In this chapter, we also explain what drives us. What we've done and how we're going about doing it are also detailed. We have also briefly discussed other possible outcomes.

In chapter 2, we talk about some related research that has been done before. In which we split the review into two sections, one dealing with the identification and classification of fruits, and the other focusing on automated farming. Here, we detailed the methods and processes that each author used to perform their research. We have made an effort to quantify the most effective procedure. As a conclusion, we also explain in the summary what drives us to explore this topic of study.

In chapter 3, we have talked about the process and approach behind this thesis. In this section, we provide a synopsis of his thesis's methodology.

In chapter 4, we not only present the findings of our experiment, but also detail the research we conducted in preparation for it. We reviewed the mathematical methods and algorithms that were integral to this study. We have discussed these ideas, as well as the methods used to discover and describe experiments on them.

In chapter 5, we make an attempt to explain the impact that our works have had on the environment and the society around them. In addition to this, we discuss the ethical implications of our research as well as our long-term sustainability plan.

In chapter 6, we present a brief summary of our findings, draw some conclusions about the work we've done so far, make some suggestions about how we can go with this in the future, and speculate about the kinds of places this might be implemented and the results it would produce.

CHAPTER 2

Background

2.1 Introduction

If we see the previous year's agricultural growth, we can see that it is constantly decreasing its value which means we need to use more technology in the farming field to overcome that value. In this chapter, we have concentrated on the prior work done by a number of publishers in this research field. These domains include picture identification, fruit detection, automated farming, and fruit classification. Most of the researchers in this field detect fruits and vegetables and grade their quality, but no one works with identical fruit stems. There are some researchers working to develop automated farming systems and technologies. We have tried to gain knowledge from their research.

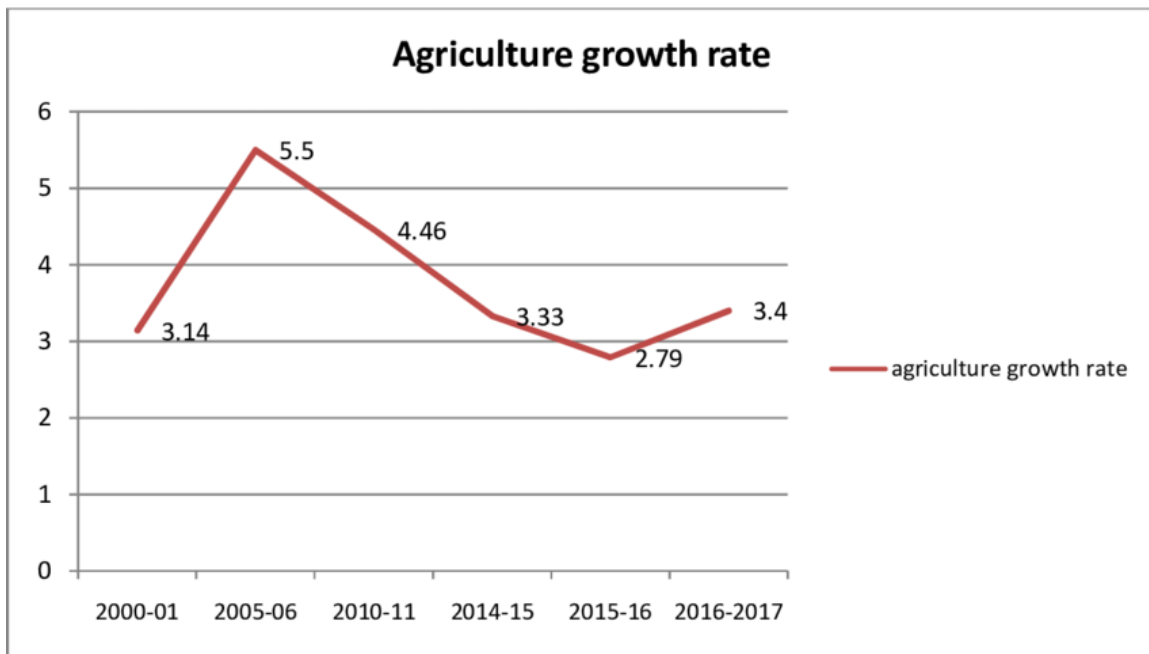


Figure 2.1: Agricultural growth rate [34]

2.2 Related Works

2.2.1 Fruit Identification or Classification

Recently, there has been a lot of progress in the field of classifying images. When it comes to the issue of categorizing fresh produce, Veggie-vision [1] was the first attempt at a fruit and vegetable recognition system. The system makes use of density, color, and texture. Because it was developed a while ago, this system does not benefit from recent advancements. Although it relies on the top four results to produce such an outcome, the stated accuracy in certain circumstances was near 95%. The data set collected in the supermarket has significant variations in lighting and color between distinct photographs, and there is no indicator for the suppression of secularities.

R. Zabih presents a method for comparing photos based on color coherence vectors. [2]. The extent to which picture pixels of that hue are constituents of a sizable area with homogenous color is how they define color consistency. These important areas are referred to as cohesive regions. In contrast to incoherent pixels, coherent pixels are a component of a significant continuous area. The approach blurs and discretizes the image's color space over 15,000 images to reduce subtle changes between adjacent pixels before computing the CCVs. The next step is to find the image's related parts so that the pixels of a certain color type can be put into coherent or incoherent groups. CCV makes two color histograms, one for coherent pixels and one for incoherent pixels, based on the types of pixels in the image. One histogram is kept, which is made up of the two.

Using computer vision, this system [3] makes it possible for self-service devices in a grocery store to identify fruits and vegetables. Two Convolutional Neural Network designs, Inception and MobileNet, are used to sort 10 different kinds of fruits or vegetables into groups. The images used by the classifier come from a Raspberry Pi Camera Module v2 that is connected to a Raspberry Pi. Even though this result is based on the top three results, the accuracy was said to be 97% in some situations.

In a broad sense, the difficulty in classifying different types of produce, fruit, and vegetables can be understood as an example of the categorization of an item. The author of referenced work [4] used Principal Component Analysis (PCA) to determine the

reconstruction error caused by first projecting the entire picture to a subspace and then moving back to the space containing the original image. However, this is very dependent on the stance, the form, and the light.

Images created with computer graphics rendering software may be so lifelike that it's difficult to tell them apart from actual photographs. With the increasing ease with which digital pictures may be modified, the distinctive status of photographs as definitive recordings is dwindling. As a solution, [5] outlines a technique for detecting fake photos from real ones. It is demonstrated that a statistical model can distinguish between photorealistic and photographic pictures based on first-order and higher-order wavelet statistics. The problem with this is that it does not provide any rendering-specific guidance on how to get more lifelike results when using these models.

Weber [6] has considered geographic limitations by means of a generative constellation model. The method is able to deal with occlusion, albeit at a high price (exponential with the number of parts). Someone creates something entirely new.

Bags of features, as introduced by Marszalek and Schmid [7], describe a picture as a disorganized collection of features. They've provided a strategy for making use of object boundaries during supervised training to capitalize on spatial linkages between characteristics. They provide more credit to features that are in agreement with the objects' shapes and locations and give less credit to features that are utilized only for background information.

When comparing photos, it is helpful to employ collections of local characteristics that are robust against modifications. An effective method for comparing pictures without grouping descriptors based on their discrete distributions of distinctive local invariant characteristics has been developed by Grauman and Darrel [8]. Images are compared using a fast and cheap approximation of Earth Mover's Distance, which determines the degree of similarity between two sets of characteristics in a bag. An EMD embedding with minimal distortion is used to translate the feature distribution of each picture onto a standardized coordinate system. Using an estimated closest neighbor search in the number of instances in the

embedded space, the most comparable examples to a new query image may be recovered in sub-linear time.

In [9], a different intriguing method is presented. This article's author has used the gradient image to derive feature points. Connecting the points along a connecting path, and determining whether or not the obtained contour is sufficiently similar to the one in the database, the match is made. The approach depends too much on silhouette cues, which are not particularly useful signals for fruits like lemons, melons, and oranges; and it necessitates a nonlinear optimization phase to obtain the ideal shape.

In [10], we included prior knowledge for distribution estimation, which allowed us to decrease the number of training examples to about 10 photos while maintaining a high recognition rate. Despite this progress, categorization remains impractical due to the exponential rise in the number of components. concerns raised in this work.

In [11], we learn how to identify the item classes represented in an unlabeled picture collection. Probabilistic models are developed in statistics textbooks. Analysis of Latent Meaning with a Custom Model Bag-of-words document representation in a corpus is employed for this purpose in text analysis, allowing for the discovery of subjects within the text. Assertion of Objectivity in the Present Since subjects and categories are interchangeable, a picture with occurrences of both is considered to include Blended concepts are used to build categories. Applying the model to photos, a SIFT-like area is created using vector quantization using a visual representation of the word. Despite not attempting to describe spatial limitations between features, these "bags of features" methods nevertheless performed better.

In their work [12], Rocha et al. provide a unified method that can integrate several features and classifiers. According to the author, the multi-class classification issue may be broken down into a series of binary classification problems, each of which can be tackled with a different collection of features and classifiers. They refer to the binary classifier as a "base learner" and describe class binarization as "a mapping of a multi-class problem into two-class problems" (divide and conquer). Specifically, if N is the number of classes, then we require $N(N-1)/2$ binary classifiers. Class I patterns are treated as positive, whereas class

j patterns are treated as negative in the ij th binary classifier, as stated by the author. To determine the final result, they determine the smallest distance between the created vector (binary outcomes) and the binary pattern (ID) for each class. They've separated the test cases into groups according to the shortest possible gap between their ID and the binary results.

When compared to paintings, photographs have distinct advantages in terms of their sharper edges, more vivid colors, and smoother textures. Identifying paintings in pictures and real-world settings automatically is a challenge that is tackled in [13]. The issue was illustrated by the writers through the use of color, edge, and texture characteristics. The accuracy of their discrimination has improved from 70–80% when using a single feature to more than 90% when employing a classifier with many features. Our pictures were compared to those of paintings and photographs, but no distinctions in edge structure were found.

Progressive Randomization (PR) [14] is a novel image descriptor that employs perturbations on the values of the least significant bits (LSB) of pictures for wide image classification, and it is described in the literature by Rocha and Goldenstein. The authors demonstrated that their technique yields diverse results for various types of photos. In order to capture the dynamic nature of the artifacts created by a perturbation process in each of the broad-image classes, they have developed a new approach. The PR descriptor's low dimensionality and uniform approach across applications (e.g., picture class, object class within a confined domain) make it particularly useful, even when using several cameras and lighting conditions. Although PR has strong separability with a small training set, its accuracy improves as the training set grows in size.

2.2.2 Automated farming

In this work [15], the author discusses how technologies like AI and the IoT are being used to mechanize and digitize farming operations. They discuss the extent to which machinery is used in rice production. They presented a number of algorithms and Internet of Things (IoT) technologies for automating the farming process. The author perceives five

weaknesses in this thesis: security, IT integration, unclear RIO, technical expertise, and interoperability. This thesis paper will help developers develop automated farming technology.

2.3 Comparative Analysis

We've read countless articles and combed through countless websites to ensure that Faster R-CNN and YOLO are the best tools for the job.

- In 2015, k He et al. presented a faster version of the R-CNN algorithm. The slowest part of the architecture is the selective search, and that comes after the Fast R-CNN. owing to the fact that it must produce 2000 proposals for each image. It accounts for a sizable portion of the total training time for the architecture as a whole. It was changed to the region proposal network in the newer version of Faster R-CNN. In the first place, the image was sent into the backbone network of this particular network. One type of feature map that can be produced by this backbone network is a convolutional one. After that, these feature maps are uploaded into the network that proposes new regions. The region proposal network begins with a feature map and then creates anchors for the map (the center of the sliding window with a unique size and scale). After that, the classification layer, which decides whether or not there is an object, and the regression layer receive these anchors and use them to make their determinations.
- The YOLO algorithm is significant for a number of reasons, including the
 - Because it can make accurate predictions in real time, this method increases the rate at which items are found.
 - YOLO is a prediction method that delivers precise findings with relatively few errors caused by the surrounding environment.
 - The method possesses great learning capabilities, which allow it to learn the representations of things and then apply them when object detection is being performed.

2.4 Scope of the Problem

There are several problems that can be created if we did not look after this study. Like-

- Similarly categorize fruit listing.
- Finding problems in multi-fruit harvesting orchards.
- In grocery stores, automated price tagging for identical fruits.
- In digital learning for children learning stage. etc.

And all these problems inspired us to work or study more about identical fruit detection. To our knowledge, this is going to be the first work on this topic. For this, we have many opportunities in front of us to do deep research on it. Our goal is to develop a method that makes it simple to tell whether or not fruit is spoiled.

The Faster Region-based Convolutional Neural Networks [31] (Faster R-CNN) and the You Only Look Once [32] (YOLO) algorithm both perform exceptionally well in the kind of work that we are doing, as our research has shown. In the not-too-distant future, we will be able to put such technologies to use in a variety of settings, including large factories, supermarkets, and farms.

Our goal is to make our work easily accessible to everyone by making its source code public and allowing anyone to view it. With our technology, magistrates can simply carry out their responsibilities in the numerous regions of our nation's wholesale markets where the wrong fruit is sold due to a lack of awareness regarding the fruit that is brought in from other countries.

2.5 Challenges

Data collecting is the most crucial step in this portion. You must utilize clear images to achieve the highest detecting results. We encountered difficulties while working on this project, including

- data collecting,
- dataset selection,
- data preparation,
- technique selection,

- best algorithm selection,
- searching for data-gathering sources,
- sources for building a model,
- and data validation.

The objective of this study is to develop a model that, when applied to images from the dataset, accurately identifies the same fruits. For example, when a variety of fruits are gathered in one location, the system may automatically determine the fruit's category and separate the fruit list.

2.6 Summary

In this chapter, we conduct a comprehensive literature review to provide context for the current state of research into picture classification issues. We have discussed the numerous methods provided by the researchers for detecting and categorizing fruits. In Section 2.2.1, we explain how to find fruits and put them into groups. In Section 2.2.2, we talk about automated farming technologies that were useful for this study.

CHAPTER 3

Research Methodology

3.1 Introduction

The number of classes and features that must be taken into account in modern vision and pattern recognition issues, such as face recognition, fingerprint identification, picture classification, and DNA sequencing, is frequently arbitrarily huge. To detect fruits using computer vision that is close to human levels of recognition is a huge task. Numerous studies have been conducted on fruit categorization, but none have addressed the issue of similar fruit. But we are trying to cover that work. To achieve that, we'll utilize well-known techniques like the YOLO and Faster R-CNN. As this is the first study on this topic, we are using some widely accessible fruits from our own nation that have a similar appearance like apple, pear, orange, and lemon. Using this system, the store can utilize fruit identification to generate pricing for fruit consumer purchases automatically. The automated sorting of fruits from a group of several fruit types with similar appearances may also be done using fruit identification in computer vision.

To conduct automatic fruit and vegetable recognition utilizing the images from the camera, there are a number of difficulties that must be overcome. Depending on how ripe they are, a variety of fruits can exhibit substantial differences in color and texture. The primary elements of natural pictures are color and texture, which are crucial for visual perception. This section provided a texture feature generated from the color photos rather than discussing color and texture features individually.

The foundation for the automated categorization of fruits is covered in this chapter. It uses a picture of a fruit as input to determine the type of fruit it is. The input pictures can include single or multiple kinds of fruit in any quantity and location.

3.2 Data Collection Procedure and Preprocessing

The procedure of gathering data is largely self-explanatory. Even so, there are several ways to collect data. When collecting data, a lot of requirements had to be completed. such as the type of data, the date it was created, if it fulfills our job expectations, and so on.

For the purpose of our research project, we collect the data on our own. We explored several other internet data sources, but the majority of them had low-resolution photographs in their data sets. However, in order to achieve a higher level of accuracy, high-quality images are required. Because of this, we collect all of our data on our own. We started by buying several fresh fruits like apples, pears, oranges, and lemons from the supermarket. Then, we unified our backgrounds so that all of our photographs of the fruit would have the same context. Then, we took the photos using our Redmi note 8 smartphones. We ended up with around 900 photos. We've narrowed it down to just 700 photos for use in the study. There were 150 images of apples, 150 of pears, 150 of oranges, 150 of lemons, and 100 images of mix fruits.

Moreover, we make our dataset available on kaggle for research by anyone. Our dataset is freely available to the scholarly community so that it may be used in a variety of ways. Figure 3.1, 3.2, 3.3, and 3.4 depicts the classes of the data set.



Figure 3.1: Apple category



Figure 3.2: Pear category



Figure 3.3: Orange category



Figure 3.4: Lemon category

Table 3.1: Quantity explanation of used data set

Class	Quantity	Total
Apple	150	700
Pear	150	
Orange	150	
Lemon	150	
Mix	100	

All images were several pixels and had a silver background. Because with this dataset, if we pre-process the pixels of the dataset, the accuracy looks so disappointing. The fruits were shot while they rotated around a fixed axis on two or three distinct axes. Next, photos are captured, and subsets are used for various purposes. The annotated dataset is divided into training, validation, and testing sets for each individual fruit.

Table 3.2: Data splitting configuration

Fruit	Training (70%)	Validation (20%)	Testing (10%)
Apple Pear Orange Lemon	490	140	70

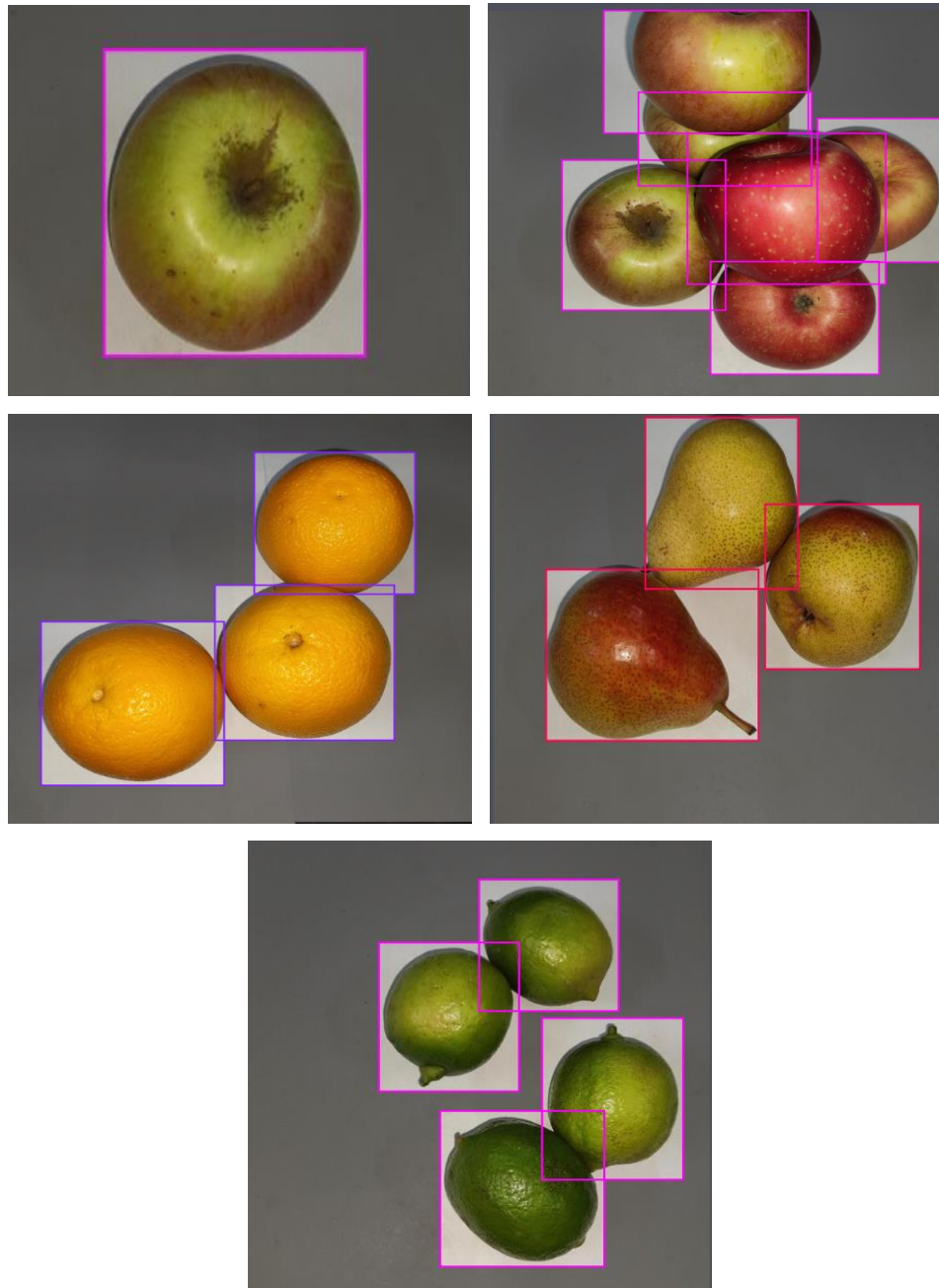


Figure 3.5: Label data visualization

Image labeling refers to the process of identifying and tagging elements within an image. Automatically synthesizing metadata or making product recommendations to consumers based on the specifics in their images both benefit from accurate and detailed image labeling. Later, when we're ready to train the algorithm, we'll use the label segmentation of various classes, as shown in Figure 3.5.

3.2.1 Feature Extraction

Feature extraction is a part of the dimensionality reduction strategy, which reduces the size of a large dataset by breaking it down into smaller, more manageable chunks. This will make processing much easier. The most important aspect of these massive data sets is the abundance of variables they contain. It takes a lot of computing resources to process these variables. With the use of feature extraction, the most useful feature may be extracted from massive data sets by selecting and combining variables into features. These features are easy to implement while also properly and uniquely representing the underlying data.

3.2.2 Data Augmentation

The quality and diversity of the sample images can be increased through image enhancement, which also helps to increase the accuracy of Faster R-CNN detection. When an illustration, the "Stern" increases samples as brightness changes to increase the accuracy of detection. When the orchard is illuminated by natural light, particularly strong light, the external shadows cast by the mangos' partial coverage or photographs taken in the opposite direction to the light source cause a significant difference in color between areas with diffuse lighting and those with normal lighting. It has an impact on the fruit sample photos' quality, which has an impact on the model's detection.

Table 3.3: Data extenuation using augmentation

Raw data	Augmented training data
700	$490 \times 3 = 1470$

New pictures may be made using the image data augmentation method. You may achieve this by making a few tweaks, such as increasing or decreasing the image's brightness, rotating it, or moving the subject horizontally or vertically. The training set size may be artificially increased through image augmentation techniques, giving the model access to a larger pool of data with which to work. Your model's accuracy can be raised in this way because it will be better able to spot new variations in the original training data. Common ways to improve an image are to move it horizontally or vertically, flip it vertically or horizontally, rotate it, change the brightness, or zoom in or out.

By using adaptive equalization technology, which is equivalent to altering the photos' brightness and broadening the variety of samples' lighting conditions, the paper improves the appearance of the fruit and lessens the impact of light on its quality. As a result, this technique is utilized to increase samples and improve image quality. In this study, we used horizontal flip, vertical flip, rotation 90 degrees up and down, rotation between -24 and +24 degrees, saturation between -25% and +25%, and vertical shear 15 degrees.



Figure 3.6: Augmented data visualization

Figure 3.6 is the visualization of augmented data where we move it horizontally or vertically, flip it vertically or horizontally, rotate it, change the brightness, or zoom in or out.

3.3 Statistical Analysis

Accuracy: One measure of a model's generalizability is its accuracy across different types of data. This is essential in cases where all classes are weighted equally. It is calculated by dividing the number of correct guesses by the total number of predictions made.

$$Accuracy = \frac{(TP+TN)}{(TP+FP+TN+FN)} \dots \dots \dots (1)$$

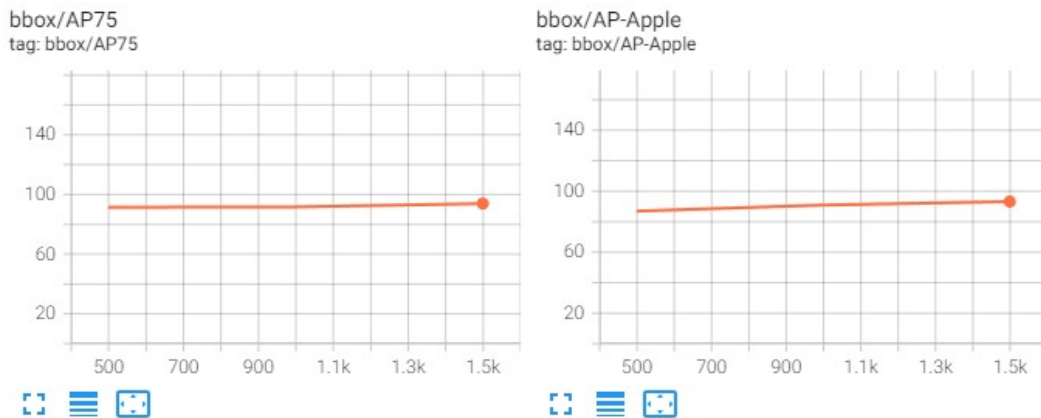


Figure 3.7: Average precision for Apple

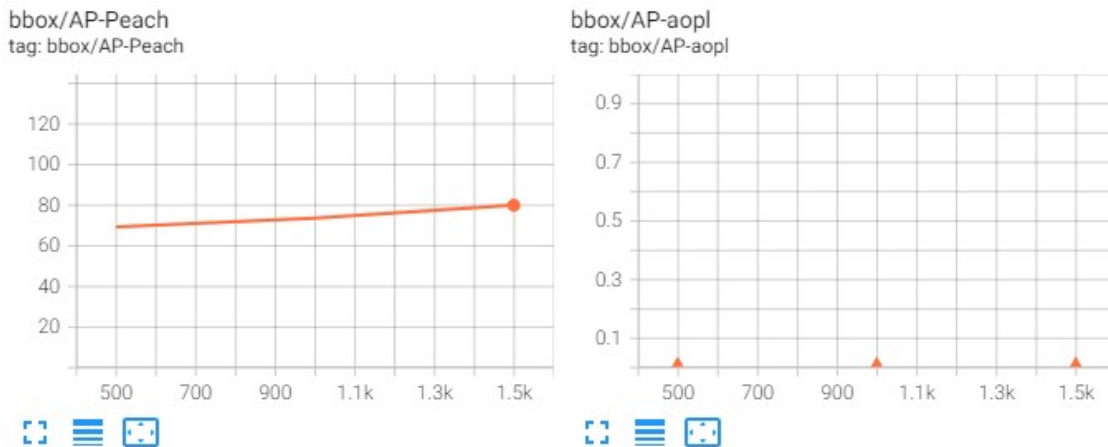


Figure 3.8: Average precision for Lemon and Orange

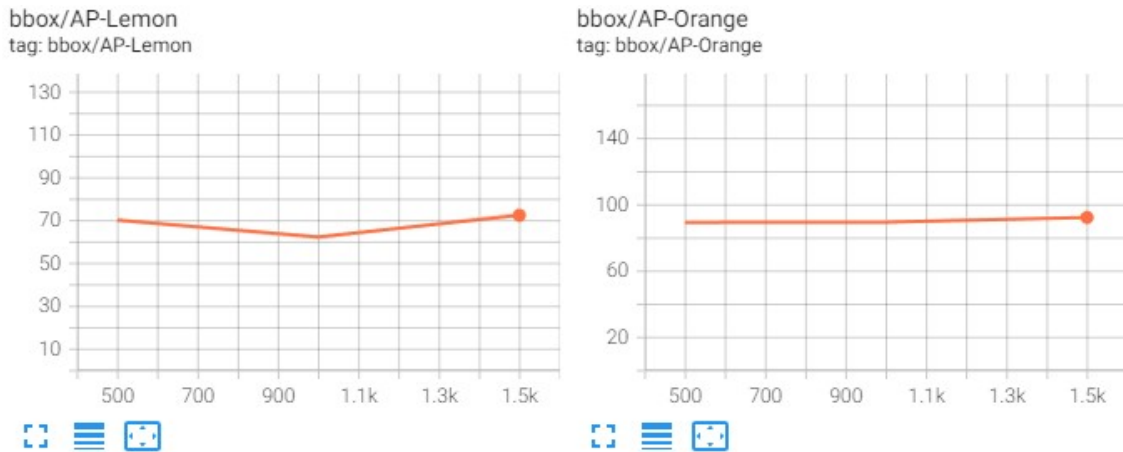


Figure 3.9: Average precision for Pear

F1-Score: To calculate the F1 score, the consonant normal of review and accuracy is used. Remember that the harmonic average is utilized instead of the more prevalent arithmetic mean. It typically comes in handy when calculating an average rate. The F1 score is derived by averaging the precision and recall values. Applying the symmetrical average makes it reasonable given that they are both rates. Here is an example of the F1 score formula:

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \dots\dots\dots(2)$$

Precision: One indication of a deep learning model's efficacy is the accuracy of its successful predictions. To calculate precision, divide the total number of accurate positive forecasts by the total number of positive predictions that were correct. Precision, for example, is the proportion of all consumers whose cancellation of subscriptions the model correctly predicted would occur as opposed to the actual consumers who did so.

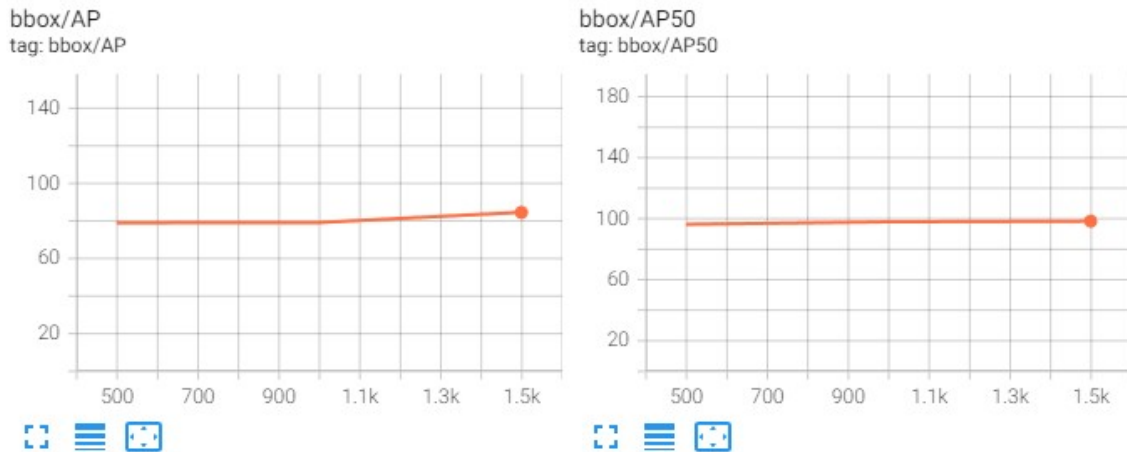


Figure 3.10: AP & AP50 visualization for Faster R-CNN

Precision is measured as the ratio of accurately identified positive occurrences to all samples that were categorized as positive, whether rightly or incorrectly. How well the model decides whether a sample is positive is its level of precision.

$$Precision = \frac{TP}{TP + FP} \dots\dots\dots(3)$$

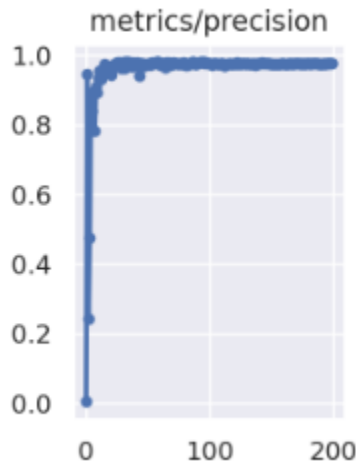


Figure 3.11: Precision for YOLO

Recall: The number of instances remembered is based on how many positive cases were correctly found out of all positive cases. Recall measures how well a model can find

examples that are good. As more good examples are found, the ability to remember grows. The recall is calculated by dividing the total number of true positives by the number of true positives that were correctly labeled as such. Recall is a measure of how well a model can pick out samples that are good. When the recall is high, it is easier to find positive samples.

$$Recall = \frac{TP}{TP + FN} \dots\dots\dots(4)$$

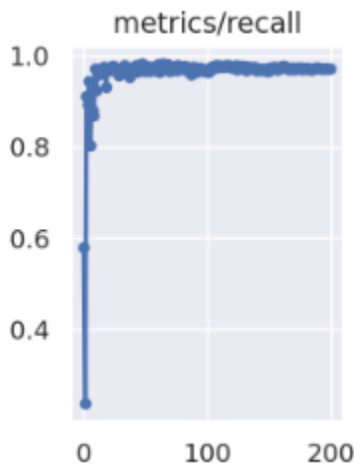


Figure 3.12: Recall measurement visualization

Specificity: Specificity refers to an algorithm's or a model's capacity to correctly forecast a true negative for each available category. In literature, it is also known as the genuine negative rate. Formally, it can be calculated using the equation below.

$$Specificity = \frac{TN}{TN + FP} \dots\dots\dots(5)$$

False Positive Rate: To determine it, divide the total number of false positives by the total number of true negatives (FP/FP+TN). If a positive result is returned even though the actual value is negative, there is a chance that a false alarm will go off.

$$FPR = \frac{FP}{Actual\ Negative} = \frac{FP}{TN + FP} \dots\dots\dots(6)$$

False Negative Rate: Whether or not the flawed speculative claim was true, the false negative rate refers to the important tests that failed to reject it. A false negative is a result where the model mistakenly anticipated the negative class.

$$FNR = \frac{FN}{Actual\ Positive} = \frac{FN}{TP + FN} \dots \dots \dots (7)$$

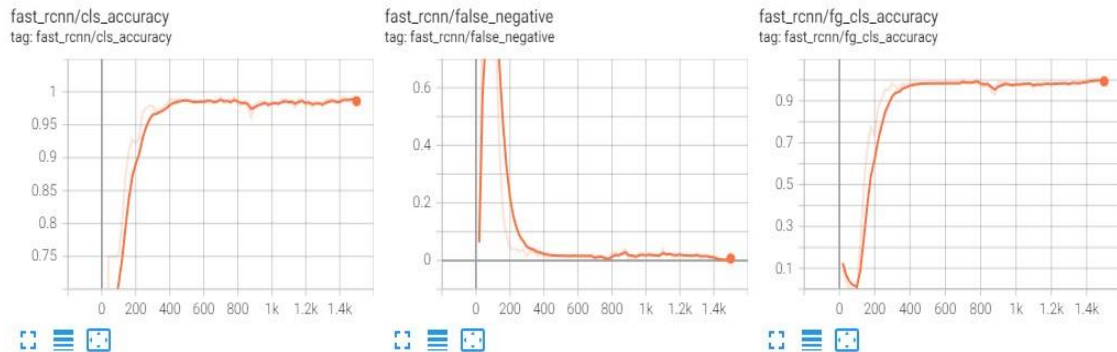


Figure 3.13: False Negative ratio graph

Root Mean Square Error: The mean percentage error of this model with respect to the test set is the average of each prediction error's actual values over all instances of the test set. It demonstrates how the Euclidean distance separates predictions from actual values. Before you can obtain the square root error, you must first establish the leftover, norm, and mean for each data point (RMSE). RMSE is often used in supervised learning approaches because it demands and uses real measurements at each predicted data point.

$$RMSE = \sqrt{\frac{\sum(y_i - y_p)^2}{n}} \dots \dots \dots (8)$$

y_i = actual value

y_p = predicted value

n = number of observation / rows

3.4 Proposed Methodology

Figure 3.14 shows how the proposed framework for the fruit recognition system works. It has two parts: training and testing. The four classes in our work model are apple, Pear, orange, and lemon.

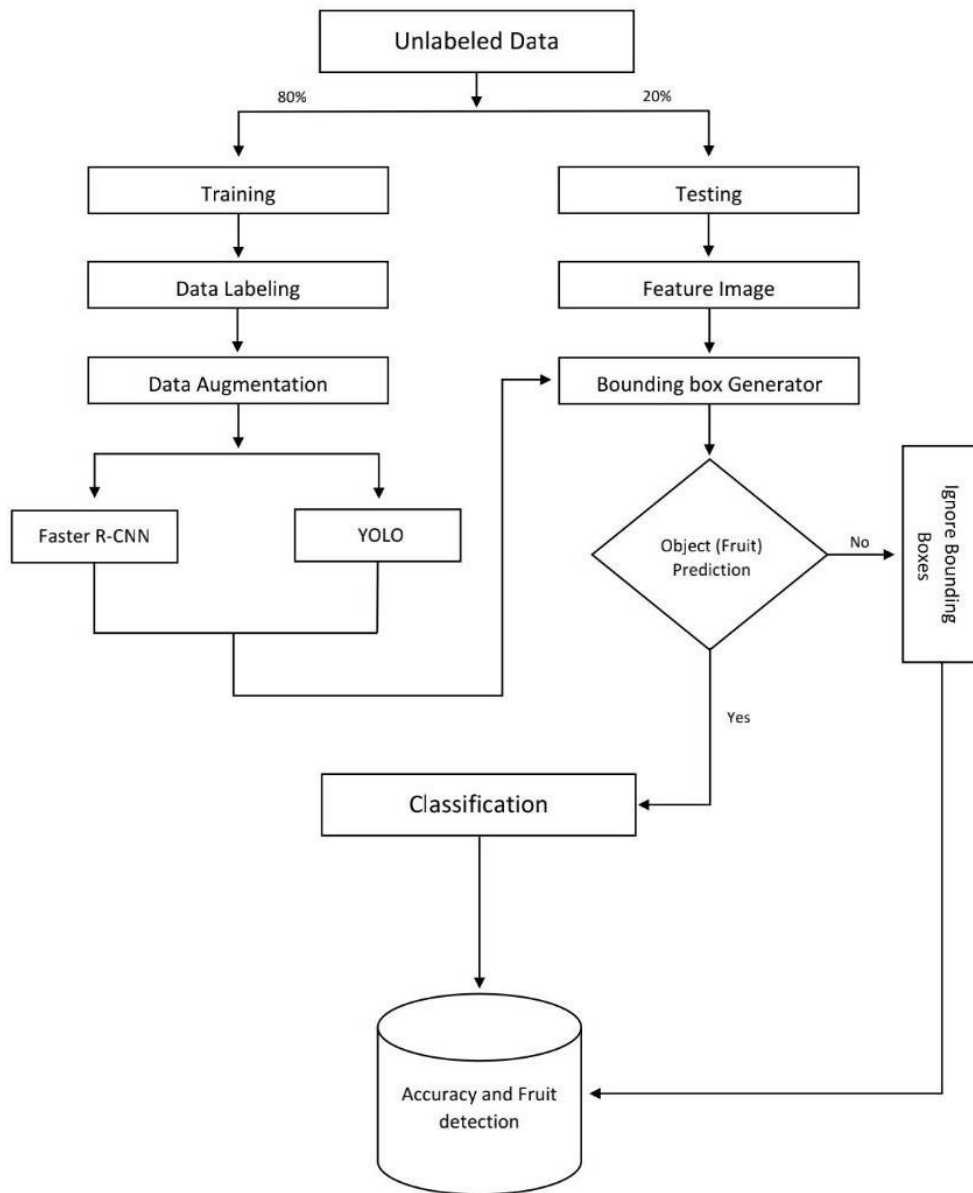


Figure 3.14: Proposed model for research work

Figure shown in 3.14 is the basic model that we have used on this research project. The first step is to look for image data from various grocery stores or from fruit orchards. The first data must be unlevelled once it has been obtained. That's why after collecting images we need to label the data. The machine will then be able to utilize this data to be fed. After that, we may utilize this data to train and test our chosen ML techniques on new data. In the case of machine learning, the more data you can train with, the better accuracy you will get, and that requires a large amount of data. That's why we will pass the images from an augmented process to increase the data. The machine will then be able to utilize this data to be fed.

After that, we may utilize this data to train and test our chosen ML techniques on new data. We will divide the data after the augmentation procedure in a ratio of 80% for training and 20% for testing. The 80% label data will be subjected to machining learning (ML) approaches using Faster R-CNN and YOLO, where it will be trained using certain pre-trained Faster R-CNN and YOLO models. It will then move on to the classification process.

In the testing part first, we will feature the testing image dataset for making the data more reliable for the test part. After that, it will generate the bounding box and search for the predicted object (in this case fruit). If the system did not find the object it goes for the accuracy calculation but if it finds any object then the classification module will start labeling the class for the finding object. Then it goes for the accuracy calculation and prediction visualization.

Table 3.4: Data configuration

Fruit	Train (80%)	Validation (20%)	Test (10%)
Apple, Pear, Orange, Lemon	1470	140	70

3.5 Classification Model

3.5.1 Faster R-CNN

Microsoft's research team developed the Faster R-CNN model. The faster R-CNN deep convolutional network performs as a single, integrated system when used for object detection. The network's prediction speed and accuracy for object locations are unparalleled. Understanding the foundational networks, R-CNN and Fast R-CNN, is crucial to understanding the inner workings of Faster R-CNN. Faster R-CNN is an enhanced variant of Fast R-CNN. To achieve this speed increase over Fast R-CNN, Faster R-CNN makes use of the region proposal network (RPN). Although the model itself is a congruent whole, Faster R-architecture CNN's is split in two. Convolutional neural network Region Proposal Network (RPN) proposes regions and the class of items to be considered within those regions.

As shown in Figure 3.15,[21] input fruit photographs from the training sets are used by the framework for fruit detection based on faster R-CNN to generate classification results and bounding box coordinates of electrical components.

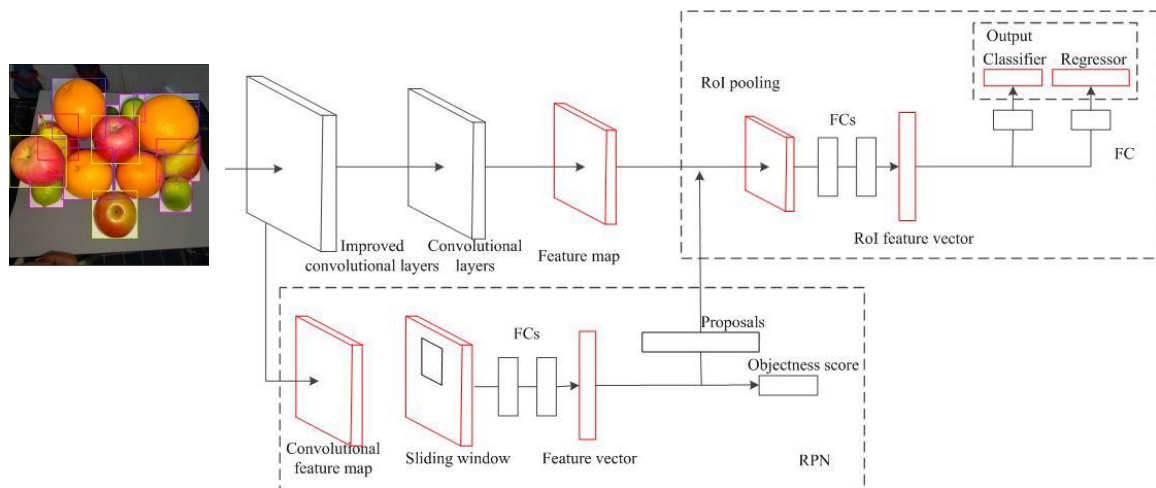


Figure 3.15: The Improved Faster R-CNN Network.

The feature extraction procedure makes use of the CNN model VGG-16. This model has four pooling layers, 13 ReLu layers, and 13 convolutional layers. The original inspection

photos are the sources of the inputs. The output feature vector of the l th layer, assuming it is a convolutional layer, is:

$$x_j^l = f_l(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l) \dots \dots \dots (9)$$

where x_j is the i^{th} output of this layer, k_{ij}^l is the convolutional kernel, $*$ means the convolution multiplication, b_j^l is the bias of this layer and $f_l(g)$ is the ReLu function. Assume the M^{th} layer is a pooling layer, the output feature vector of this layer is:

$$x_j^m = f_2(\beta_j^m S(x_j^{m-1}) + b_j^m) \dots \dots \dots (10)$$

where β_j is the connecting coefficient, x_j^{m-1} is the input for this pooling layer, $S(g)$ is the summation of the whole matrix, b_j^m is the bias and $f_2(g)$ is the softmax function.

There is a risk of overfitting during training if there are insufficient training samples. To further improve detection accuracy, balance model complexity, and collect more data, a regularization technique is used to the weights for several high dimensional characteristics, causing them to decay. Two loss functions are added to the convolutional and pooling layers to make them more effective. The different filming angles can automatically adjust the size of each convolutional layer and the kernel parameters to keep them within a good range. As such, the loss function of the convolutional layer is defined as follows:

$$L_{cv}(W, x^k) = \frac{1}{2m} [\sum_{i=1}^m (y_k - W * x_{ik}^2 + \lambda \sum_{j=1}^n w_j^2)] \dots \dots \dots (11)$$

where $W = [\omega_1, \omega_2, \dots, \omega_n]^T$ are the parameters for the kernel, n is the number of the kernels, $*$ means the convolution multiplication, $x^{(k)} = [x_{1k}, x_{2k}, \dots, x_{mk}]^T$ is the k^{th} input, m is the dimension of the database, y_k is the actual label of the k^{th} sample, and λ is the regularization penalty factor. In the proposed loss function, a likelihood function and a regularization function. The outputs are as close as the inputs through these two functions.

The loss function in the pooling layer is defined as

$$L_{pl}(W, x^k) = \frac{1}{2m} [\sum_{i=1}^m (y_k - U * x_{ik}^2 + \lambda \sum_{j=1}^n \mu_j^2)] \dots \dots \dots (12)$$

where the first expression is showing the reconstruction loss and the second 165 expression is presenting the model complexity. $U = [\mu_1, \mu_2, \dots, \mu_t]$ are the pooling parameters and t is the total number of the parameters. λ is usually determined by expert knowledge, and it will be chosen based on testing results. Then a convolutional feature map can be obtained for each inspection image.

After taking an inspection image of any size as input, an RPN will recommend a sequence of rectangular regions. Obtaining region suggestions involves dragging a small 3x3 spatial window from the convolutional feature map of the network across the convolutional feature map generated by the final shared convolutional layer. A sliding window is mapped to a lower dimensional feature, creating an intermediary layer. To save as much of the information from the input feature vector as feasible, a region of interest (ROI) pooling layer is utilized to reduce the size of the feature maps while maintaining their shape. Certain conversion vectors, such as 44, 22, and 11, are employed in the pooling procedure. To begin, we utilize the 4×4 vector to divide the world into 16 halves. Every area executes a maximum pooling strategy. The identical method is then utilized on the 1×1 and 2×2 vectors. As a result, the dimension reduction for every proposed region of varied sizes is 21. Two of its offspring, the box classification and box regression layers, are inextricably linked to it. Multiple region proposals are projected for each sliding-window site, with k representing the maximum number of possible ideas at that site. Each sliding location in this analysis employs a total of $k = 9$ anchors. The classifier and regressor function as follow:

1. **Classifier:** The chances that the proposed region actually includes an object are computed at this layer. Once the feature map has been properly analyzed, the softmax function is used to calculate the probability P_i that each element in the feature map I comprises the target item. The regions having the highest P_i in the aggregate rankings make up the 300 regions proposed.
2. **Regressor:** The regressor uses the Intersection over Union (IoU) index to measure the accuracy of the bounding box and, by extension, to pinpoint the location (x, y) of the anchor box's origin as well as its width and length. Considerations that go into calculating IoU include:

$$IoU = \frac{A \cap B}{A \cup B} \dots\dots\dots(13)$$

where the proposed areas for the two regions A and B are. Assume that the suggested region's width and height are (w, h) and that (x, y) are the coordinates of the region's center point. Then a proposed region can be described using a four-dimensional index (x, y, w, h). (Px, Py, Pw, Ph), (Gx, Gy, Gw, Gh) are representing the anchor box, predicted box, and ground truth box respectively. In a regression procedure, the window of IoU can reposition the margins of the initially suggested object region.

The error between the originally proposed region and the actual region is $t_k = (t_x, t_y, t_w, t_h)$, where:

$$t_x = \frac{G_x - P_x}{P_w}, t_y = \frac{G_y - P_y}{P_h} \dots\dots\dots(14)$$

$$t_w = \log\left(\frac{G_w}{P_w}\right), t_h = \log\left(\frac{G_h}{P_h}\right) \dots\dots\dots(15)$$

Therefore, the objective function can be written as:

$$L_e = \sum_{i=1}^N (t_k^i - \omega_k^T \phi(P_i))^2 \dots\dots\dots(16)$$

where N is the number of samples, ω_k is the coefficient learned through training, and P_i is the feature vector used to train the model. The objective function can be solved using the least squares technique. Thus, the regressor produces 4k outputs that reflect the coordinates of k boxes, and the classifier produces 2k scores that indicate estimates of the likelihood that each proposition is an item or background. Anchor boxes in the number k reference set are connected to proposal sets for that many regions. However, the same CNN model can be used to build an RPN without training a separate model, which significantly impedes processing speed.

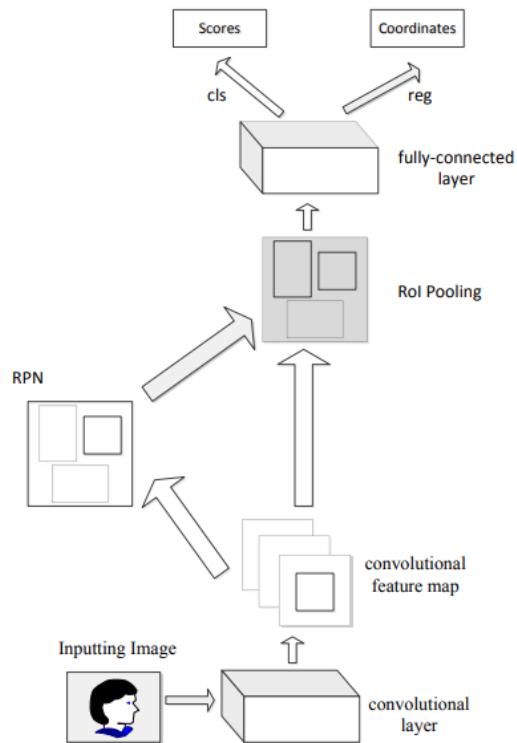


Figure 3.16: A look at the framework of Faster R-CNN

3.5.2 YOLO v5

One of the most well-known and cutting-edge object detectors available today, Yolo is a member of the one-stage detector family. The original version (YOLO) was introduced by Redmond et al. [17] in 2016, and it was heralded as a major advancement in real-time object detection/tracking. Previous object detectors, such as R-CNN and Fast R-CNN, employed region proposals to suggest potential item bounding boxes in an image, then bounding box classification to categorize the objects within, followed by bounding box refining and duplicate removal.

While the efficacy of such frameworks is undeniable, the complexity of their underlying design can be off-putting. For instance, the inference is still sluggish, and each of the aforementioned components must be trained individually. YOLO takes a regression-based approach to the problem of object detection, wherein the aforementioned steps are all

performed by a single network, in order to eliminate these bottlenecks. The number of false positives and inference bottlenecks has been drastically reduced. And YOLO showed excellent generalization abilities.

Twenty-four convolutional layers and two fully linked layers make up Yolo's architecture. In [17], the YOLO framework is described. Unfortunately, YOLO ran into generalization problems when the items in the image were too tiny or when the image proportions were off. A number of changes were made to YOLOv2 [18] to address the problems that were present in the previous version. For instance, we now support larger picture sizes, have done batch normalization on all convolutional layers, and have swapped out all fully connected layers with anchor boxes for bounding box prediction. More so, multiscale training was included.

When compared to YOLOv1, YOLOv2 proved to be both quicker and more accurate. The network has a total of 11 levels dedicated to finding objects inside a larger 19-layer network (Darknet-19). As a result, the performance of small objects suffers as a result of the layers' down sampling of the input, which eliminates fine-grained signals. The third iteration, YOLOv3, compromises precision for speed in order to increase the detection rate, thus resolving the earlier problem. In particular, we assign a Darknet-53 variation that has 106 layers. Additionally, many techniques were implemented, including skip connections, residual blocks, and up sampling. The performance with small objects has greatly improved, and the up-sampling attribute was responsible for preserving the fine-grained details. This, however, results in higher processing costs [19].

In particular, CSPDarknet53 is the backbone upon which YOLOv4 is built. In addition, CSPDarknet53 incorporates a spatial pyramid pooling strategy to enhance the receptive field and differentiate contextual data. Training may also be run on a single GPU in YOLOv4 [20], which is another great feature. The process is shown in Figure 3.17.

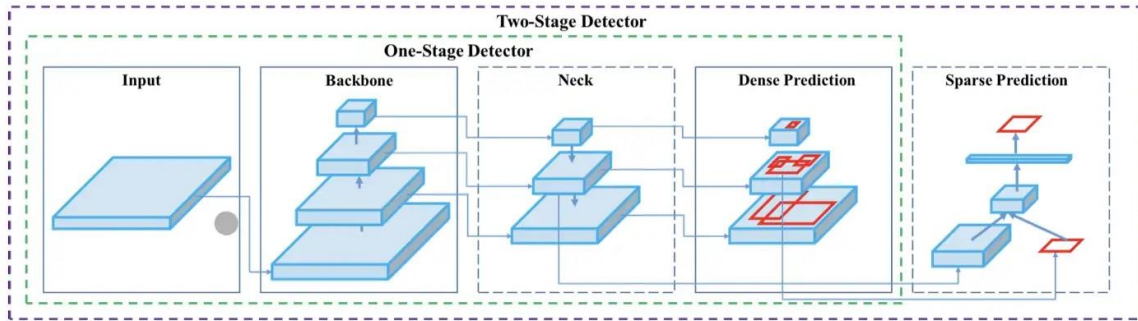


Figure 3.17: object detection architecture of YOLO v4

Recently, ultralytics [21] released the last version of YOLO. Even though its name has caused some debate, most people call it "YOLO V5". CSPNet is chosen as the backbone because it takes less time to process. PANet is used as a model neck to build feature pyramids so that they can handle changes in the size of objects. Anchor boxes are used to figure out class probabilities, bounding boxes, and objectiveness scores, just like they were in earlier versions. In YOLOv5, they reduced the number of parameters. The YOLOv5 network will be trained with four different versions: YOLOv5l, YOLOv5m, YOLOvx, and YOLOvs. In this letter, we choose the YOLOv5s, which is the lightest model. In conclusion, one of the most important things about YOLOv5 is that it looks at object detection from a regression point of view, while DETR uses an encoder decoder architecture and EfficientDet takes advantage of a feature pyramid network.

3.6 Implementation Requirements

We had to utilize a variety of software and tools, the majority of which were open-source technologies, for the aim of our research on fruit identification. Because we mostly used open-source software, there were limitations. This work was done mostly in the Python programming language on our home computer, which runs Windows 11 operating system. The Table 3.1 was placed here to describe about those instruments.

Table 3.5: Research instrument explanation

Package Name	Description	URL Link
Python [24]	Python [24] is a pretty sophisticated programming language, to put it simply. It is an application development language with a focus	www.python.org

	<p>on object orientation. For quick application development, it provides a great and appealing built-in data structure that mixes dynamic typing and binding. It is a fantastic programming language for ML, as well. This kind of machine learning project. It is also a commonly used language.</p>	
<p>Image Annotation Lab [25]</p>	<p>The process of classifying photographs in order to develop AI and machine learning models is known as image annotation [25]. Human annotators frequently use image annotation tools to label pictures or tag pertinent information, for example by giving separate items in an image-appropriate class name. After that, the output data, sometimes called structured data, is supplied to a machine learning algorithm for the process commonly known as training a model.</p>	<p>ial.4smartmachines.com</p>
<p>JupyterLab[26]</p>	<p>Jupyter[26] is an interactive web-based development environment for writing code, keeping notes and manipulating data. Users may easily change and reorganize their data science, scientific reporting, machine learning, and scientific computing processes because of its adaptive user interface. Because of its modular design, it may be customized by adding new functions and enhancing existing ones.</p>	<p>jupyter.org</p>
<p>Google Colab [27]</p>	<p>The Google Research team created the Colaboratory, or "Colab" for short. Colab is top-notch in the domains of ML/DL/EDU. With just</p>	<p>colab.research.google.com</p>

	<p>a web browser, you can develop and run Python code. Colab, which is a hosted Jupyter notebook service, makes it simple and cost-effective for customers to gain access to powerful computing tools (including GPUs).</p>	
--	---	--

Used Libraries/Module:

- **Matplotlib:** Among the tools for plotting data in Matplotlib is a series of functions known as Pyplot. It allows you to do things like identify lines in a plot, set the bounds of a plot during form generation, etc.
- **NumPy:** The NumPy Python module is a well-liked option for manipulating arrays. Its scope includes matrix operations, linear algebra, and the Fourier transform. The NumPy library for Python provides a variety of tools and methods for working with arrays of different sizes. Arrays may be executed mathematically and logically with NumPy. NumPy is a Python library for numerical calculation, to put it simply. Additionally, the phrase "numerical Python" is used.
- **TORCH:** A tensor library with extensive GPU support, similar to NumPy. The torch package implements mathematical operations on multidimensional tensors and provides data structures for these tensors. Additionally, it offers a variety of tools for effective serializing of arbitrary types and tensors, as well as other helpful tools.
- **YAML:** Anything written in JSON can be converted to YAML since YAML is a tight superset of JSON. It's mostly used for configuration files in projects, and the code is quite simple to read and comprehend.

- **Detectron2:** A complete overhaul of Detectron that began with Mask R-CNN benchmark is called Detectron2. PyTorch has now been used to implement the platform. Detectron2 now has a new, more modular design that makes it adaptable, expandable, and capable of offering quick training on one or more GPU servers.
- **CV2:** OpenCV is a handy open-source toolkit that may be used for a wide variety of computer vision tasks, including video analysis, CCTV footage analysis, and image analysis. More than 2,500 efficient algorithms make up OpenCV, which is written in C++. We may utilize this library to begin concentrating on real-world issues while developing computer vision applications that we don't want to design from scratch.
- **Random:** A Python built-in module called Python Random is used to create random integers. These numbers are not actually random since they are pseudo-random. This module may be used to generate random numbers, print a random value for a list or string, and do other random operations.

CHAPTER 4

Experimental Results and Discussion

4.1 Experimental Setup

In this part of the report, we will typically do an analysis of the outcome of our work based on the proposed model of our work, and we will explain the outcome of our work using appropriate logic. Our model was trained using the Detectron2 framework for the Faster R-CNN algorithm and the PyTorch framework for the YOLO v5 algorithm. Also, make use of the epoch 200 for improved training purposes. The percentage of our data that comes from training, validation, and testing looks like this: 70%, 20%, and 10%.

4.2 Experimental Results & Analysis

It was to be expected to encounter difficulties while attempting to use a new algorithm for detecting identical types of fruit. This is why we strove to diversify the procedure by utilizing a variety of techniques. We examined and tried a variety of methods before settling on what we believe to be the optimal one for this experiment. To better our work and the results it produced, we experimented with a variety of methodologies.

We picked up knowledge of various image processing and classification strategies in the course of our work. We used the colab's python library, which we found under the name function, to perform the image classification. We discovered how transfer learning may be used to make a model more precise. We discovered that deep learning works well on tiny datasets by reviewing the literature on the topic. More specifically, the newer versions of YOLO and Faster R-CNN. In order to improve the accuracy of both Faster R-CNN and YOLO, we applied the same efficient strategy for both.

On the dataset that had been used before, we tried out two distinct algorithms, each of which was distinct from the algorithm that had been used before. The results of our experiment can be broken down into a few distinct categories. Following the completion of our research on deep learning algorithms, we utilized two of the available algorithms. We are able to construct our very own model using the assistance of deep learning in order

to locate better output. Faster R-CNN and YOLO are the names of the two deep learning algorithms that we utilized. Python libraries and functions were utilized in our study to determine the relative amounts of accuracy possessed by each technique when applied to the same fruit dataset that was used in the evaluation of the models. That indicates which algorithm is the more suitable one for our research.

Table 4.1: Algorithm accuracy and time

Algorithm	Speed (ms/image)	Accuracy (%)	Average Accuracy (%)
YOLO v5	30	97	94.45
Faster R-CNN	20	91.91	

We can see from Table 1 that the accuracy that we achieve while using the YOLO algorithm and the Faster R-CNN method respectively is 97% and 91.91%. The combined accuracy of both algorithms comes to an average of 94.45%.

Additional metrics for the distinct classes supported by both techniques are presented in Tables 2, 3, and 4. Recall, mean average precision, average precision small, average precision medium, average precision huge.

Table 4.2: Evaluate metrics accuracy for YOLO v5

Evaluate metrics	Classes			
	Apple (%)	Pear (%)	Orange (%)	Lemon (%)
AP	99	99	98	98
mAP@50	99	97	98	99
mAP@95	94	97	90	90
Recall	99	91	97	99

Table 4.3: Average precision for individual classes in Faster R-CNN

Evaluate metrics	Classes			
	Apple (%)	Pear (%)	Orange (%)	Lemon (%)
AP	91.69	94.31	93.57	88.06

Table 4.4: Evaluate metrics accuracy in Faster R-CNN

Evaluate metrics	Accuracy (%)
AP50	98.82
AP75	98.29
APs	Nan
APm	75.54
API	92.06

4.2.1 Graphical Visualization

There are a lot of outputs generated by the model, and it can be difficult to make sense of them all without some type of visual representation. We've included a handy chart to help with that.

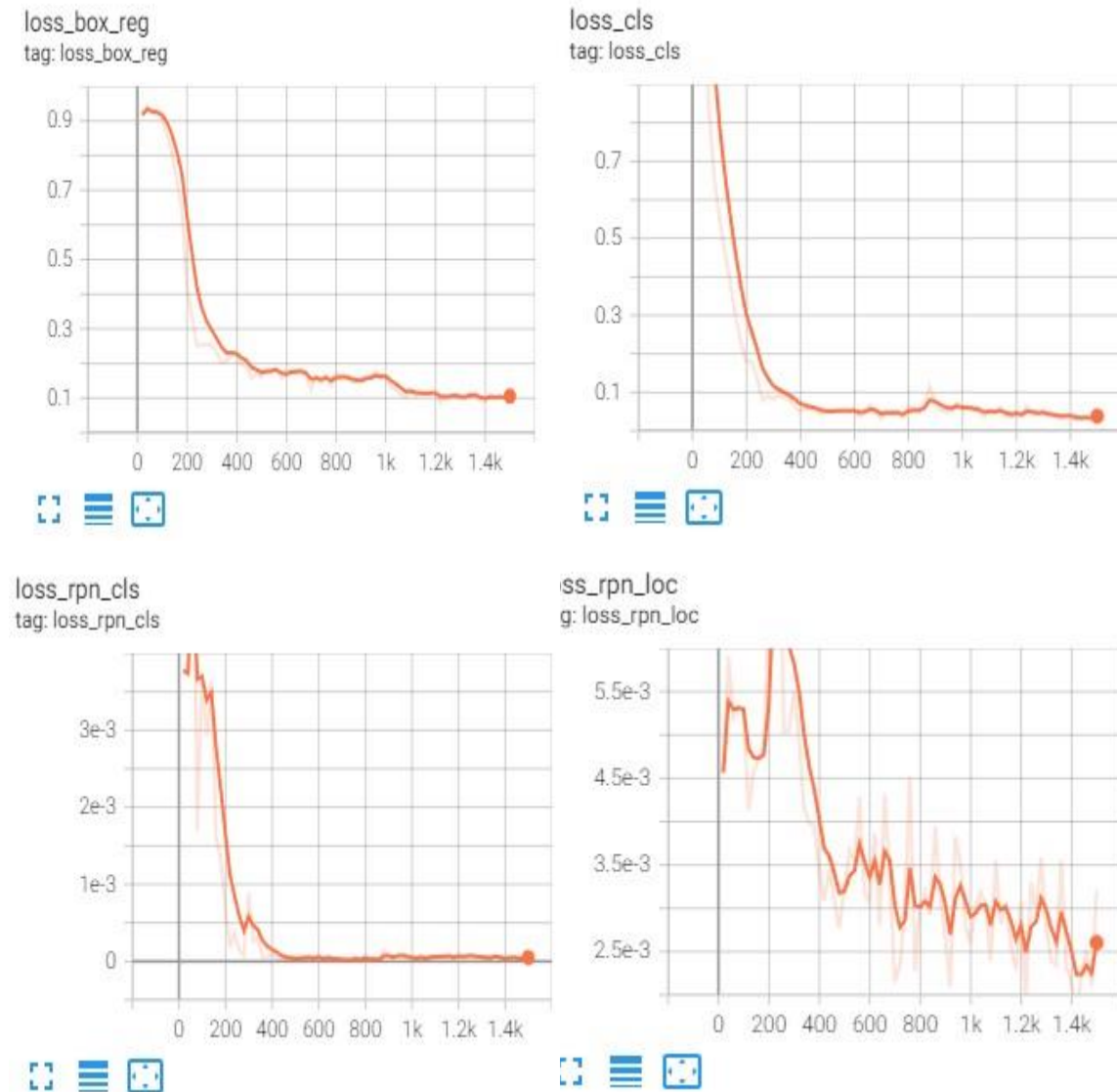


Figure 4.1: Process losses

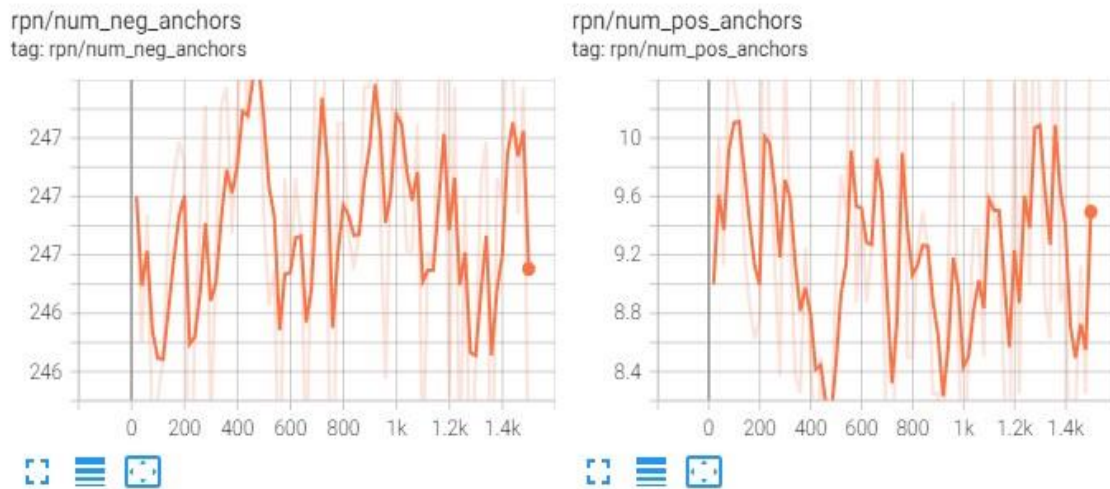


Figure 4.2: Anchors visualization

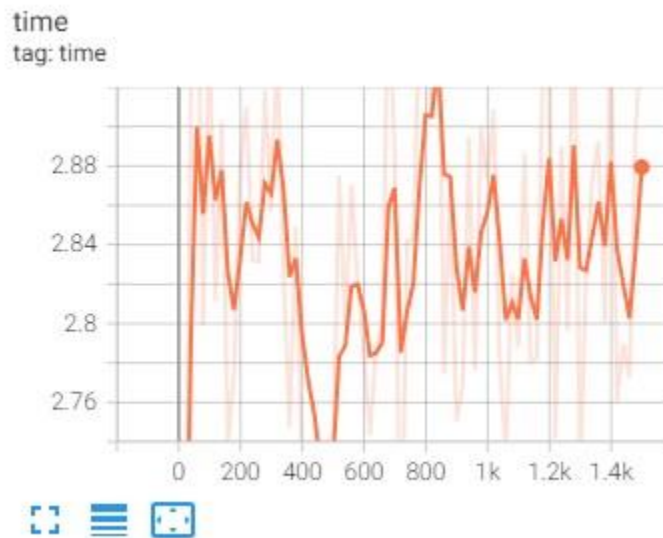


Figure 4.3: Total time-lap

The precision loss is displayed in Figure 4.1 for box reg, loss cls, loss rpn cls, and loss rpn loc. The representations of both negative and positive anchors could be found in Figure 4.2. In addition, Figure 4.3 illustrates how the total amount of time spent training decreased.

4.3 Discussion

In this chapter, a strategy for detecting identical fruits that is based on image processing is proposed as well as assessed. The strategy that has been suggested will primarily consist of three steps. In the initial step of the process, acquire the reliable dataset that will be used for the study. The second stage consists of the extraction of characteristics. Training and classification are carried out on the Faster R-CNN and YOLO models during the third step of the process. In order to conduct an analysis of our program, we selected a case study consisting of four different sorts of classes, namely Apple, Pear, Orange, and Lemon.

Through experimentation with several Deep Learning techniques, we were able to successfully complete our aim. We have utilized a total of two algorithms in this project. Before beginning our work, we had to look around for a bit for a few various things. As soon as we decided on an algorithm, we got to work on it. Next, we learned how precise these algorithms actually are. As was previously indicated, we had already self-labeled our data. And improved upon them by adding to them.

We routinely achieve more than 97% and 91% success rate with both algorithms, respectively. which enhances the prior work mentioned [22][29] in the 4.1 portion of 4th chapter. When comparing the results of the Faster R-CNN and YOLO algorithms, YOLO comes out on top in terms of accuracy. However, despite their similarity, apples, pears, oranges, and lemons all constantly acquire 99%, 99%,97%, and 98% accuracy using YOLO and 91.69%, 94.31%, 93.57%, and 88.06% accuracy using Faster R-CNN. When we used the YOLO v5 algorithm, we got 97% accuracy, which was the best we could do.

CHAPTER 5

Impact on Society, Environment and Sustainability

5.1 Impact on Society

The introduction of a system that can identify identical fruits has the potential to have a significant and beneficial effect on society. It is possible for this technology to assist in lowering the rate at which food is wasted while simultaneously raising the level of food security achieved. Additionally, it has the potential to assist in the reduction of labor expenses involved with the manual sorting and grading of fruit. Additionally, the increased precision of fruit sorting and grading can help bring down prices for customers, giving them greater access to fruits and other types of products that are within their price range. Last but not least, the increased precision of fruit sorting and grading can assist farmers and other agricultural employees in obtaining higher prices for their produce, which in turn leads to an enhanced quality of life as well as increased economic opportunities.

5.2 Impact on Environment

The influence on the environment of a fruit detection system that was otherwise identical would be primarily determined by the components of the system, such as the variety of sensors applied, the supply of electricity, and the materials that were utilized in its construction. Although it is expected that the system would be fueled by a renewable energy source, such as solar or wind, the sensors and materials that are utilized may have an adverse effect on the natural environment. Sensors with a higher energy consumption would have a greater need for electricity, while materials such as plastic might be challenging or even impossible to recycle or dispose of in a responsible manner. In addition, there is a possibility that the system will have a negative impact on the local flora and fauna. Some of the sensors may emit sound or light, which may cause the animals to become distracted or anxious.

5.3 Ethical Aspects

Identical fruit detecting systems' environmental impact would be determined by factors like their sensors' sensitivity, their power supply, and the materials they were built from, even if they were otherwise similar. The sensors and materials employed may have an adverse effect on the environment, despite the fact that the system would presumably be fueled by a renewable energy source like solar or wind. More electricity is needed to power sensors that use a lot of power, and materials like plastic may be challenging to recycle or dispose of in an environmentally responsible manner. Some sensors may produce noise or light, which may be distracting or even stressful to neighboring species. While the only thing we're doing to affect the environment is developing an algorithm for recognizing duplicate fruits, this might change if somebody decides to utilize the method to create a system. Since we aren't participating, we won't be contributing to any negative ecological effects.

5.4 Sustainability Plan

We intend to continue working on this system in the future and ultimately attempt to materialize it. For that, we've got several long-term strategies like-

- Create a structure for the system to consume as little energy and as few resources as possible.
- Replace conventional fuel sources with environmentally friendly ones like solar or wind energy.
- Make use of energy-saving measures including low-power processors, efficient cooling systems, and LED lighting.
- Create a plan to reuse and recycle system parts and resources.
- Produce and fabricate the system's components using only environmentally friendly materials.
- Invest in cloud computing and energy-efficient data centers to lessen the planet's overall energy burden caused by storing and processing information.
- Develop a carbon offset program to offset the energy used in the system.

- Utilize green logistics practices such as using efficient shipping methods and packaging materials.
- Invest in green IT solutions such as virtualization and server consolidation to reduce energy usage.
- Implement a monitoring system to track the energy usage of the system and identify areas for improvement.

CHAPTER 6

Summary, Conclusion, Recommendation and Implication for Future Research

6.1 Summary of the Study

In this thesis, we analyzed the difficulties associated with both the automatic recognition of fruits from images and the detection of visually similar fruits in images. Since automated fruit harvesting technologies and authenticating fruits in grocery stores rely on being able to tell one type of fruit apart from another, this study aims to design a method for doing just that. We have experimented with both the Faster R-CNN and Yolo v5 algorithms in an effort to find similar results. Yolo v5 achieved 97% accuracy, whereas Faster R-CNN achieved 91%. Among the two algorithms, Yolo v5 gave the best result, which is 97%.

6.2 Conclusions

For identical image detection, we used the two methods Faster R-CNN and Yolo v5. Here, we make use of the pear, apple, lemon, and orange fruit classes. Faster RCNN provides a region of interest for convolution, whereas YOLO simultaneously does detection and classification. Due to its end-to-end training, we would suggest that YOLO seems to be a more efficient method for object detection.

In chapter 3, two frameworks were proposed for the automatic detection of identical fruits from the images. The two frameworks use images for detection. Among the two algorithms, Yolo v5 worked better than Faster R-CNN, and Yolo v5 gave better results. Our current system works to find fruits that look the same. R-CNN, the speedier option, achieved 91% accuracy, while Yolo v5 achieved 97%.

6.3 Implication for Further Study

In this thesis, we have developed the fruit recognition problem one deep learning approach and one feature at a time. All of the fruits we've used have been exact duplicates. Eventually, researchers hope to examine many features and multiple algorithms simultaneously. Multiple features can be combined to improve the system's performance and output. We plan to enhance its grading functionality and implement a system for identifying fruits and vegetables that look the same in the future.

APPENDIX

A research project with the working title "Identical fruit recognition using Faster R-CNN and YOLO v5" was where we had begun our adventure. During the course of this effort, we faced a number of challenges. The most important one is the collecting of data. Because collecting data locally is a challenge that requires a lot of effort and time from students. People in a country like Bangladesh did not have a favorable reaction to it. Therefore, we need to gather all of the necessary data from various web sources. And finally, after a significant amount of time, effort, and a number of failed tries, we were successful.

REFERENCE

- [1] Bolle, Ruud M., Jonathan H. Connell, Norman Haas, Rakesh Mohan, and Gabriel Taubin. "Veggievision: A produce recognition system." In *Proceedings Third IEEE Workshop on Applications of Computer Vision. WACV'96*, pp. 244-251. IEEE, 1996.
- [2] Pass, G., Zabih, R. and Miller, J., 1997, February. Comparing images using color coherence vectors. In *Proceedings of the fourth ACM international conference on Multimedia* (pp. 65-73).
- [3] Femling, F., Olsson, A. and Alonso-Fernandez, F., 2018, November. Fruit and vegetable identification using machine learning for retail applications. In *2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)* (pp. 9-15). IEEE.
- [4] Turk, M.A. and Pentland, A.P., 1991, January. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition* (pp. 586-587). IEEE Computer Society.
- [5] Lyu S, Farid H. How realistic is photorealistic? *IEEE Transactions on Signal Processing*. 2005 Jan 17;53(2):845-50.
- [6] M. Weber, "Unsupervised Learning of Models for Object Recognition," PhD Thesis, Caltech, Pasadena, US, May 2000.
- [7] Schmid, C., 2006, June. Spatial Weighting for Bag-of-Features. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) (Vol. 2, pp. 2118-2125). IEEE Computer Society.
- [8] Grauman, K. and Darrell, T., 2005, June. Efficient image matching with distributions of local invariant features. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (Vol. 2, pp. 627-634). IEEE.
- [9] Berg, A.C., Berg, T.L. and Malik, J., 2005, June. Shape matching and object recognition using low distortion correspondences. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) (Vol. 1, pp. 26-33). IEEE.
- [10] Fei-Fei, L., Fergus, R. and Perona, P., 2006. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4), pp.594-611.
- [11] Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A. and Freeman, W.T., 2005, October. Discovering objects and their location in images. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1* (Vol. 1, pp. 370-377). IEEE.
- [12] Rocha, A., Hauagge, D.C., Wainer, J. and Goldenstein, S., 2010. Automatic fruit and vegetable classification from images. *Computers and Electronics in Agriculture*, 70(1), pp.96-104.
- [13] Cutzu, F., Hammoud, R. and Leykin, A., 2005. Distinguishing paintings from photographs. *Computer Vision and Image Understanding*, 100(3), pp.249-273.
- [14] Dubey, S.R., 2012. Automatic recognition of fruits and vegetables and detection of fruit diseases. *Master's theses*.

- [15] Subeesh, A. and Mehta, C.R., 2021. Automation and digitization of agriculture using artificial intelligence and internet of things. *Artificial Intelligence in Agriculture*, 5, pp.278-291.
- [16] Edan, Y., Han, S. and Kondo, N., 2009. Automation in agriculture. Springer handbook of automation, pp.1095-1128.
- [17] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
- [18] Redmon, J. and Farhadi, A., 2017. YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).
- [19] Redmon, J. and Farhadi, A., 2017. YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).
- [20] Bochkovskiy, A., Wang, C.Y. and Liao, H.Y.M., 2020. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.
- [21] Wan, S. and Goudos, S., 2020. Faster R-CNN for multi-class fruit detection using a robotic vision system. *Computer Networks*, 168, p.107036.
- [22] Sachin, C., Manasa, N., Sharma, V. and AA, N.K., 2019, November. Vegetable Classification Using You Only Look Once Algorithm. In *2019 International Conference on Cutting-edge Technologies in Engineering (ICoN-CuTE)* (pp. 101-107). IEEE.
- [23] Liu, B., Zhao, W. and Sun, Q., 2017, October. Study of object detection based on Faster R-CNN. In 2017 Chinese Automation Congress (CAC) (pp. 6233-6236). IEEE.
- [24] Python, available at <<<http://www.python.org/>>> last accessed on December 20, 2022 at 08:00 PM.
- [25] SmartMachines, available at <<<https://ial.4smartmachines.com/>>> last accessed on December 23, 2022 at 08:00 PM.
- [26] Jupyter, available at <<<http://jupyter.org>>> last accessed on December 20, 2022 at 09:00 PM.
- [27] Colaboratory, available at <<<https://colab.research.google.com/>>> last accessed on December 12, 2022 at 09:00 AM.
- [28] Liu, B., Zhao, W. and Sun, Q., 2017, October. Study of object detection based on Faster R-CNN. In 2017 Chinese Automation Congress (CAC) (pp. 6233-6236). IEEE.
- [29] Kaggle, available at <<www.kaggle.com/datasets/chrisfilo/fruit-recognition?resource=download&select=Pear>> last accessed on November 12, 2022 at 09:00 PM.
- [30] YOLO v5, available at <<<https://github.com/ultralytics/yolov5>>> last accessed on December 20, 2022 at 10:00 AM.
- [31] Wikipedia, available at <<https://en.wikipedia.org/wiki/Region_Based_Convolutional_Neural_Networks>> last accessed on December 25, 2022 at 10:00 AM.
- [32] Section, available at <<<http://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/>>> last accessed on December 25, 2022 at 10:00 AM.

- [33] The Daily Star, available at <<<https://www.thedailystar.net/supplements/30th-anniversary-supplements/news/revitalising-bangladeshs-agriculture-sector-2042629>>> last accessed on December 25, 2022 at 10:00 AM.
- [34] Rezvi, M.R., 2018. The factors of declining agricultural growth in Bangladesh and its impact on food security. South Asian Journal of Social Studies and Economics, 1(1), pp.1-9.

Identical Fruit Detection Using Faster R-CNN & YOLO

ORIGINALITY REPORT

19%

SIMILARITY INDEX

15%

INTERNET SOURCES

10%

PUBLICATIONS

9%

STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	5%
2	coek.info Internet Source	3%
3	Submitted to Daffodil International University Student Paper	3%
4	ikee.lib.auth.gr Internet Source	1%
5	www.inderscience.com Internet Source	1%
6	www.ic.unicamp.br Internet Source	1%
7	Submitted to Jacksonville University Student Paper	1%
8	Submitted to University of Northumbria at Newcastle Student Paper	<1%
9	Shaohua Wan, Sotirios Goudos. "Faster R-CNN for multi-class fruit detection using a	<1%