

Smart Automated Fruit Freshness Recognition System Using Image Processing and Deep Learning

Submitted by

Prantha Shil
18301219

Zisanur Rahman
18301025

Jawad Bin Jalil
22141044

Kazi Rishad Bin Sakib
18301274

Md. Tamim Hossain
19101417

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
September 2022

© 2022. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis presented is our own remarkable work completed while studying at Brac University.
2. The thesis contains no previously published or written by a third party content unless properly cited through comprehensive and comprehensive referencing.
3. The thesis contains no material that has been approved or submitted for any other degree or certificate at a university or other institution.
4. We have honored all major sources of assistance.

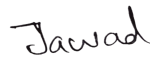
Student's Full Name & Signature:



Prantha Shil
18301219



Zisanur Rahman
18301025



Jawad Bin Jalil
22141044



Kazi Rishad Bin Sakib
18301274



Md. Tamim Hossain
19101417

Approval

The thesis titled “Smart Automated Fruit Freshness Recognition System Using Image Processing and Deep Learning ” submitted by

1. Prantha Shil (18301219)
2. Zisanur Rahman (18301025)
3. Jawad Bin Jalil (22141044)
4. Kazi Rishad Bin Sakib (18301274)
5. Md. Tamim Hossain (19101417)

Of Summer, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September,20 2022.

Examining Committee:

Supervisor:



(Member)

Dr. Jia Uddin
Associate Professor (On leave)
Department of Computer Science and Engineering
Brac University
Assistant Professor (Research Track)
AI and Big Data Department, Endicott College
Woosong University

Co-Supervisor:



(Member)

Faisal Bin Ashraf
Lecturer
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

MD. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Bangladesh is one such country with a tropical monsoon climate typified by significant seasonal rainfall, high temperatures, and high humidity. A wide range of tropical and subtropical fruits are abundant in Bangladesh. The fruits that are most frequently grown are mango, jackfruit, pineapple, banana, litchi, lemon, guava, wood apple, papaya, tamarind, watermelon, pomegranate, plum, etc. Automated fruit recognition is essential since fruits in Bangladesh's markets come in a variety of types and qualities. This thesis presents a deep learning-based automated fruit recognition model that uses image processing and deep learning architecture to identify fruits and grade their quality. We will make use of our dataset of Bangladeshi fruits for the experimental evaluation.

This thesis aims to provide a novel Convolution Neural Network (CNN) structure, called VGG19, for identifying, classifying, and evaluating fruit objects according to their freshness. An application for Keras called VGG19 has a high degree of accuracy in object detection. The outcomes show that our method works better than the linear predictive model and demonstrate its particular merit.

Keywords: CNN, VGG19, Deep Learning, Fruit Freshness, Regression, Image Recognition, Keras application.

Dedication

We would like to honor our beloved parents with this thesis. Including all of the amazing professor and instructors we met and acquired knowledge from while obtaining our Bachelor's degree. It was a gratifying experience.

Acknowledgement

To begin with, we would like to express our deepest gratitude to the Almighty, the most forgiving and generous Creator of everything in the universe, for giving us the strength and will to start and successfully complete our research.

Next, we would like to thank Dr. Jia Uddin, who supervised our research, for his insightful and constructive guidance during the planning and execution of this particular research. We are grateful for his willingness to give up his time so generously. We would also like to convey our appreciation to the teachers and employees of BRAC University's computer science and engineering department. They have continually led us throughout our stay at BRAC University, particularly in establishing and strengthening our foundation in education and knowledge.

We also want to convey our indebtedness to our dearest family members and friends for all they have done for us. We owe a debt of appreciation to everyone who took part in our study and helped us, whether it was personally or through another person. A special thanks to Md. Tanzim Reza, who helped us with technical ideas. We sincerely appreciate his guidance and assistance.

Finally, and this should go without saying, we would like to acknowledge BRAC University for facilitating us to demeanour this research and for providing us with the tools and support we needed.

List of Figures

3.1	The teacher-student structure.	9
3.2	Knowledge categories in a teacher model.	10
3.3	Knowledge distillation in training schemes.	11
3.4	Transfer Learning Strategies	11
3.5	Types of Transfer Learning Strategies and their SettingsWorkflow	12
3.6	Transfer Learning Using Pre-Trained Deep Learning Model.	12
3.7	Transfer Learning Process.	13
3.8	Layers of VGG-19.	15
4.1	YOLOv5 Performance Comparison	16
4.2	YOLOv5 Architecture	17
4.3	Concept of TP, TN, FP, FN for YOLO Model	18
4.4	VGG-16 Architecture	19
4.5	Layers of VGG-16	20
5.1	Work Plan.	22
5.2	Model Distribution.	22
7.1	Training and Validation Accuracy of Teacher Model	26
7.2	Training and Validation Loss of Teacher Model	27
7.3	Student Training and Validation Accuracy	27
7.4	Training and Validation Loss of Student Model	28
7.5	Scratch Student Training and Validation Accuracy	28
7.6	Training Loss and validation Loss of Scratch Student Model.	29
7.7	Comparison Between Constructed Models	29
7.8	Comparison Table	30
7.9	YOLOv5 Model Comparison	30
7.10	Inference test of The YOLOv5s	31

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ANN Artificial Neural Network

BiFPN Bi-directional Feature Pyramid Network

CNN Convolutional Neural Network

CSPNet Cross Stage Partial Network

FLOPs Floating Point Operations Per Second

FN False negative

FP False Positive

FPN Feature Pyramid Network

FPS Frame Per Second

GPUs Graphics Processing Units

ILSVRC ImageNet Large Scale Visual Recognition Challenge

KL Kullback-Leibler

NumPy Numerical Python

PANet Path Aggregation Network

ReLU Rectified Linear Unit

RGB Red, Green, and Blue

tanh tangent Hyperbolic

TN True Negative

TP True Positive

VGG Visual Geometry Group

YOLO You Only Look Once

Chapter 1

Introduction

Bangladesh prospered thanks to the large variety of tropical and subtropical fruits that are widely grown throughout much of the country. All major and minor fruits develop and become available in May, June, and July, which is why these months are referred to as fruit festival months. In addition, summer is the best time of year to grow fruit because of the climate and topography. Bangladesh is classified as the eighth-largest producer of mangoes, with 24 lakh tons produced a year, and the second-largest producer of jackfruit, with 10 lakh tons produced annually. However, Bangladesh's recent decline in fruit production means that there is currently insufficient fruit to meet demand. Fruit marketing, processing, and storage are significant problems in our nation. Storage, processing, and marketing of fruit are substantial challenges in our country. However, it may be claimed that people eat enough fruit to meet their nutritional demands on a daily basis. These fruits have a life cycle; after a certain amount of time, a fruit loses its freshness or shelf life. After that, bacterial and fungal problems cause it to deteriorate. Fruits' exterior shells exhibit deterioration-like visual characteristics.

This article uses deep learning to classify fruits and grade their freshness by analyzing diverse fruit photos. We have evaluated the areas that we have looked into for this project, including VGG-19 for identifying the area of interest with considerations of high-resolution images, YOLO, VGG-16, MobileNet, and ResNet as the virtual platforms for feature extraction for freshness grading.

1.1 Research Problem

Fruits are essential for a nutritious diet since they are stuffed with vitamins, minerals, and nutrients like potassium, folic acid, polyphenols, and antioxidants. Fresh fruits are essential since they are rich in vitamins and minerals and are known for their ability to prevent vitamin C and vitamin A deficits. To help fruit purchasers decide whether a fruit is fresh or rotting in the first half, a fruit freshness detecting system is being developed to assess the freshness of various fruits. We researched many strategies for reaching the system's goals.

In the majority of developing nations, fresh fruits constitute a vital element of a healthy lifestyle. Our immune systems depend on fresh fruits, so every country

should provide fresh fruits to its citizens so that everyone has a robust immune system. Fruits are high in potassium, which helps lower your risk of heart attack and stroke. Both kidney stones and bone loss can be avoided with potassium. The body's production of red blood cells is also aided by the nutrient folate (folic acid). A deficiency of folate in the neural tube results in a congenital disability known as spina bifida. Both women of childbearing age who might get pregnant and expectant mothers in the first trimester need enough folate. Antioxidants called polyphenols have been shown to alter gut microecology or the ratio of good to harmful bacteria. Additionally, including fresh fruit in a balanced diet offers antioxidants that aid in repairing free radical damage and may offer protection from some cancers. It might be advantageous for intestinal health. Our method focuses on automatically determining whether a fruit is fresh. The objective is to develop a fruit freshness detection image processing system.

The functioning of the body's systems and overall health depend on fruit nutrients. Eating raw fresh fruit lower the chances of having heart disease, type-2 diabetes, and strokes. These projects' methodologies were justified. The operation of each subsystem is then described. In this study, we aim to identify fruit freshness, so we collected datasets from various fruits. We collected fresh and decaying apples separately in order to distinguish between them. One problem we ran across when evaluating the dataset was the variety of the images we had taken. The array of fruit images is necessary to carry out this inquiry and provide reliable findings.

Additionally, we just analyzed the exterior layer of the fruits to decide if they were fresh or not. However, we are still working on developing freshness detection. Therefore, we can tell whether or not a fruit is rotten by sensing the fruit itself. To help the buyer buy fresh fruits, for instance, our technology will identify the fruit and then immediately inform whether it is rotten.

1.2 Research Objectives

Fruits are a prominent source of dietary nutrients and integral to our everyday lives. Additionally, it is crucial for protecting against illnesses, including high blood pressure, cancer, and heart disease. However, fruits are also influential hosts for bacteria, fungi, etc. Fresh fruit is also necessary to have. Furthermore, certain fruits are imported from other nations. Fruits cause a number of issues during export and import, including the potential for food quality to suffer during packaging. It is difficult to determine whether the fruits are safe for consumers.

In this study, we will implement a fruit freshness detection system using image processing. However, using an optical instrument designed for the fruit freshness detection system and an algorithm based on the data gathered, this system was built to identify fruits and the freshness of fruits.

The research objective:

1. For the classification of fruit images, three transfer learning models— VGG19, VGG16, and YOLOv5—have been modified.
2. To determine which of these three models provides us with the best accuracy and uses the least amount of memory.
3. To modify the model and transfer data from an enormous model to a smaller one.
4. Examining fruits for freshness to prevent purchasing rotting fruit from the market.

Chapter 2

Literature Review

Our thesis, which focuses on the identification of fruit freshness, is based on the VGG-19 Architecture, a comprehensive Convolutional Neural Network with pre-trained layers and a solid insight of what separates an image in terms of appearance, color, and pattern. CNNs have a number of layers that combine to convert an image into an output the model can comprehend. The VGG-19 model is frequently used for transfer learning, and deep learning training makes it possible to employ it more effectively the second time around. To precisely determine the solution, numerous studies have been carried out in this domain. These studies significantly impacted our conclusions and our modeling of the detection of fruit freshness.

2.1 Related articles

In “A Design of Deep Learning Experimentation for Fruit Freshness Detection,” the authors suggested a concept of a computer vision-based method employing deep learning using the CNN model to identify fruit freshness [17]. Then, the custom-created CNN model is assessed for classification using publicly available datasets of both fresh and rotting fruits. 10,901 photos of three different types of fruit in six classes—fresh fruit and rotting fruit—make up the dataset utilized in this study. This research adopts the open-source TensorFlow framework, Python 3.6, and a PC with requirements for the CNN design training procedure.

A unique neural network structure, YOLO + Regression CNNs, was suggested by author Yuhang Fu in, “Fruit Freshness Grading Using Deep Learning” for fruit object localization, categorization, and freshness grading [11]. Fruits are treated as an object, and images of each one are fed into YOLO for segmentation and regression before being graded for freshness. Six fruits—an apple, a banana, a dragon fruit, an orange, a pear, and a kiwi—from varied places with various ambient noises, irrelevant adjacent objects, and lighting conditions help compensate for the retrieved dataset. Each variety of fruit 700 has 4,000 images total that have been assembled. The collection of images was divided in a 9:1 ratio into sets for training and validation. All fruits showed good identification scores with a maximum accuracy of up to 99% for both the training and validation sets. For both training and validation, the lowest percentages are 85% and 82%, respectively. Accuracy, precision, and recall metrics show the YOLO classifier’s average performance. The accuracy and

precision scores are above 90%, while the recall and precision scores are above 80%.

Nazrul Ismail and Owais A. Malik, the authors of "Real-time visual inspection system for grading fruits using computer vision and deep learning techniques," developed a superior machine visionary system relying on state-of-the-art deep learning methods. It strives to provide a safe, and cost-effective method for automating the visual examination of fruits' freshness and beauty by stacking ensemble approaches [19]. To determine which deep learning model would be the most effective in classifying fruits, researchers trained, evaluated, and compared the performance of MobileNetV2, ResNet, EfficientNet, NASNet, and DenseNet. Additionally, the recommended solution delivers real-time visual inspection utilizing a low-cost Raspberry Pi module outfitted with a camera and a touchscreen display that is necessary for human interaction. The algorithm accurately evaluates each object after successfully separating many examples of fruits from an image. Two datasets were used to train and test the system. The EfficientNet model's average accuracy for the test sets for bananas and apples was 98.6% and 99.2%, respectively. When using the stacking ensemble deep learning methods, a little increase in the recognition rate such as 0.06% for bananas and 0.03% for apples was also noticed. The performance of the created system has outperformed that of previously used approaches on the same datasets. Additionally, during real-time test execution on experimental samples, the accuracy was found to be 93.8% for bananas, and 96.7% for apples, demonstrating the usefulness of the proposed solution.

In "Fruit Freshness Detection Using Raspberry PI," the authors illustrate how to use Raspberry Pi to identify the freshness of fruit and provide a system for doing so [8]. When the fruit is picked, it should be positioned on a conveyor belt so that it may traverse through a sensor device that can determine the fruit's cumulative freshness status and display it. Numerous parts were employed, including digital image processing, a Raspberry Pi 0W, a proximity sensor, a gas sensor, and a load cell. The visual of the fruit is first captured using the suggested system. Then, the picture is passed through the open CV processing stage, where the fruit characteristics, such as their coloring, design, and shape, of fruit samples, are recovered. To spot the damaged fruit, edge detection techniques are used to segment the acquired image. The load cell is used to measure the weight of the samples that are put and to average the weights in order to assess the fruit's freshness. To determine if gas has been sprayed upon or is present in the fruits, a gas sensor is employed. With an efficiency of 80% in freshness detection, this technique is prospering.

The authors of "A Novel Model to Detect and Classify Fresh and Damaged Fruits to Reduce Food Waste Using a Deep Learning Technique" suggest a way to utilize conventional refrigerators by adding gadgets like cameras to the compartments [20]. The three fruit varieties—oranges, apples, and bananas—are to be found both in their fresh and damaged stages. Additionally, using deep learning techniques, the models are constructed to determine how to recognize the fruit in the refrigerator most precisely. Convolutional Neural Networks (CNNs) function incredibly well for image detection and fruit item classification. This paper's main objective is to determine if fruits are fresh or damaged using CNN. To train the datasets in identifying fruit images, they have pre-trained models like AlexNet, Google Net,

ImageNet, VGG16, and VGG19 available. The proposed model has been tested on 840 photos and trained on 7560 images. The test accuracy was 97.1% , while the training accuracy was 98.4%.

Chapter 3

Methodology

3.1 Models

To conduct this study, we applied transfer learning of CNN models and Knowledge Distillation for efficient model abridgement. The goal is to apply knowledge distillation of the base model to the student model and compare the results. We used one of the Transfer learning Models named VGG19 Architecture as our base model, and the student model we have created is a modified version of the VGG-16 network with a slight modification. This modus operandi and architecture are exemplified below.

3.1.1 VGGNet

In 2014, Karen Simonyan and Andrew Zisserman from the University of Oxford devised the CNN architecture referred as VGGNet [3]. VGG19 is a sophisticated CNN with pre-trained layers and keen knowledge of how form, color, and structure create an image. With millions of different pictures and challenging classification problems, VGG19 has undergone extensive training [3].

3.2 Knowledge Distillation

A technique known as knowledge distillation means transporting details from a broad, onerous model or group of models to a single, more comprehensible model that may be used in real-world applications [10]. In essence, it's a kind of model compression that Bucilua and colleagues initially examined with success in 2006.

Large deep neural network deployment challenges are particularly relevant for edge devices with constrained memory and processing power. A model compression strategy was initially put up to address this problem by transferring the information from a big model into training a more petite model without suffering a significant performance loss [10]. This process of distilling knowledge from a voluminous model into a pocket-sized one was defined by Hinton and colleagues as the "Knowledge Distillation" methodology.

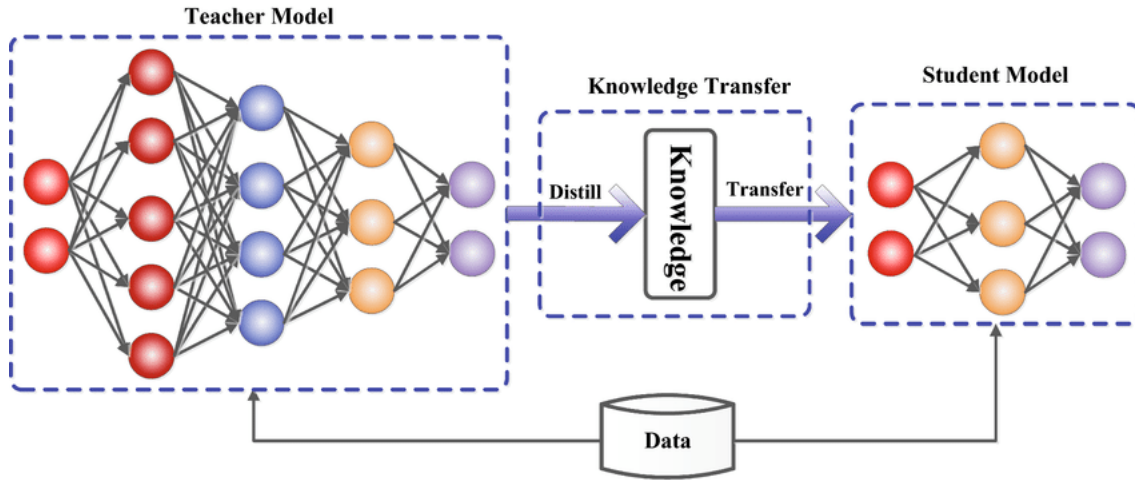


Figure 3.1: The teacher-student structure.

As shown in Figure 3.1, a smaller student model pursue to mimic a larger teacher model and uses the teacher’s expertise to consummate parity or greater accuracy in knowledge distillation [10]. We will go into more detail about the design and workings of the knowledge distillation framework. Generally speaking, Three components make up a knowledge distillation system.

1. Knowledge
2. Distillation algorithm
3. Teacher-student architecture

Knowledge: Knowledge in a neural network often refers to the learned biases and weights. In addition, a large deep neural network has a wide variety of information sources. The different forms of knowledge are categorized into three different types:

1. Response-based knowledge
2. Feature-based knowledge
3. Relation-based knowledge

These three distinct knowledge categories are shown in Figure 3.2 and are derived from the teacher model [4].

Training: For developing student and teacher models, there are three main groups of techniques: offline, online, and self-distillation [4]. The classification of the distillation training approaches depends on whichever the teacher model is modified simultaneously with the student model such as Figure 3.3.

Architecture: :Deep neural networks’ depth and breadth make it difficult to transfer knowledge from them [10]. One of the most popular models for information transmission is the student model, which are:

1. A teacher model that is more simplified, has fewer layers, and has neurons per layer

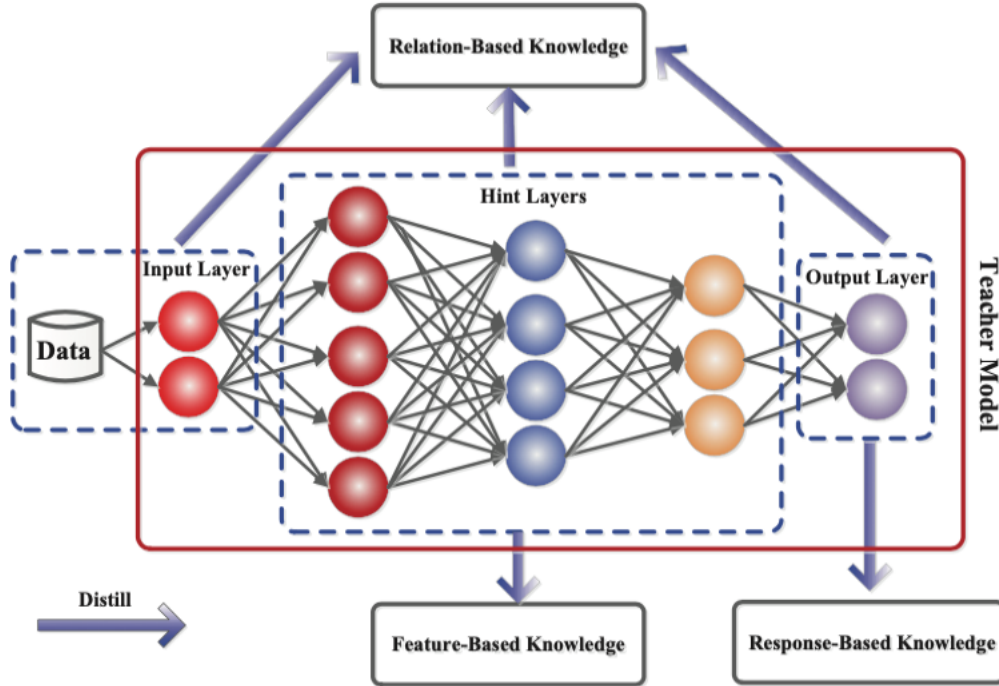


Figure 3.2: Knowledge categories in a teacher model.

2. Response-based knowledge
3. A quantized version of the model
4. A more compact network with effectual fundamental operations

Along with the methods mentioned above, more recent innovations like neural architecture search may be used to create the best student model architecture, given a certain teacher model.

3.3 Transfer Learning

When we attempt to instruct a kid how to recognize fruits, we approach by exhibiting apples of various hues, such as red apples, green apples, golden or yellow apples, etc., as well as showing a scattered of apples, such as lemon, apples, mangoes, etc. The kid acquires the ability to select the appropriate environment by modeling throughout many circumstances.

A model that has been trained for one job is repurposed for a different, related task using the machine learning approach known as transfer learning. Although it is not just a topic for deep learning research, it is related to issues like idea drift and multi-task learning [9]. Transfer learning necessitates training basis connectivity on the target and core collections first, after which the learned attributes are reused or relocated to a second target network that will be trained on the targeted collection. If somehow the qualities are generic, that is, applicable to both the source task and the destination task, This process is more likely to be successful.

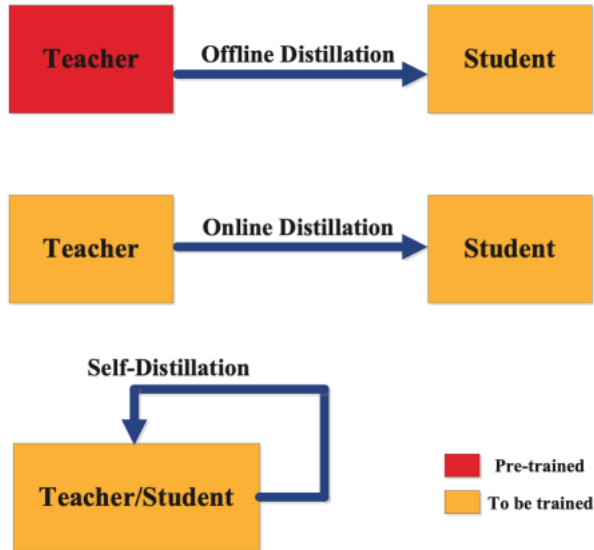


Figure 3.3: Knowledge distillation in training schemes.

Inductive Transfer Learning: The form of transfer learning used in deep learning is inductive transfer [4]. Here, using a model fit on a separate but similar job tends to adversely narrow the variety of models. In the Figure 3.4 the transfer learning strategies are briefly shown.

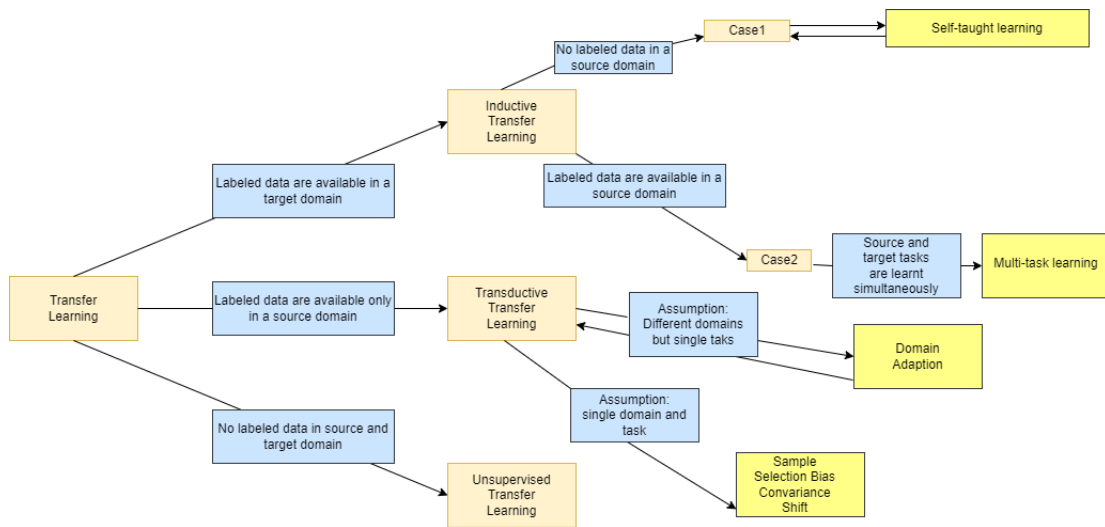


Figure 3.4: Transfer Learning Strategies

Unsupervised Transfer Learning: This environment is comparable to inductive transfer as a whole, with an emphasis on unsupervised activities in the particular sphere. Although the functions are distinct, the destination and source domains are comparable. In this case, neither of the domains has access to labeled data [10].

Transductive Transfer Learning: The source and destination actions, in this case, are comparable, but their associated domains are not [2]. The source and destination actions, in this case, are comparable, but their associated domains are

not. In this instance, the originating region has a lot of tags, while the destination realm contains zero. This could be subsequently categorized into subgroups where the feature regions or edge likelihood of occurrence differ. In figure 3.5, it yields an overview of the diverse contexts and potential outcomes for each of the aforementioned strategies.

Learning Strategy	Related Areas	Source & Target Domains	Source Domain Labels	Target Domain Labels	Source & Target Tasks	Tasks
Inductive Transfer Learning	Multi-task Learning	The Same	Available	Available	Different but Related	Regression Classification
	Self-taught Learning	The Same	Unavailable	Available	Different but Related	Regression Classification
Unsupervised Transfer Learning		Different but Related	Unavailable	Unavailable	Different but Related	Clustering Dimensionality Reduction
Transductive Transfer Learning	Domain Adaptation, Sample Selection Bias & Co-variate Shift	Different but Related	Available	Unavailable	The Same	Regression Classification

Figure 3.5: Types of Transfer Learning Strategies and their Settings Workflow

3.3.1 Transfer Learning Approaches to Deep Learning

Transfer learning research has concentrated on fields like natural language processing and image recognition. Numerous models attained state-of-the-art performance [12]. Deep transfer learning, which is based on these pre-trained neural networks, is what deep learning is used for.

Layered architectures used in deep learning systems allow for learning various features at different layers. Higher-level characteristics are compiled at the network's outermost layers, and as we move deeper into the network, they get more precise. The outcome is achieved by connecting these levels to the last layer [1]. Because of this, it is now possible to use well-known pre-trained networks—including the Google Inception Model, Oxford VGG Model, and Microsoft ResNet Model—without the absence of their last layer, a fixed feature classifier for further activities.

Idea: use outputs of one or more layers of a network trained on a different task as generic feature detectors. Train a new shallow model on these features.

Assumes that $D_S = D_T$

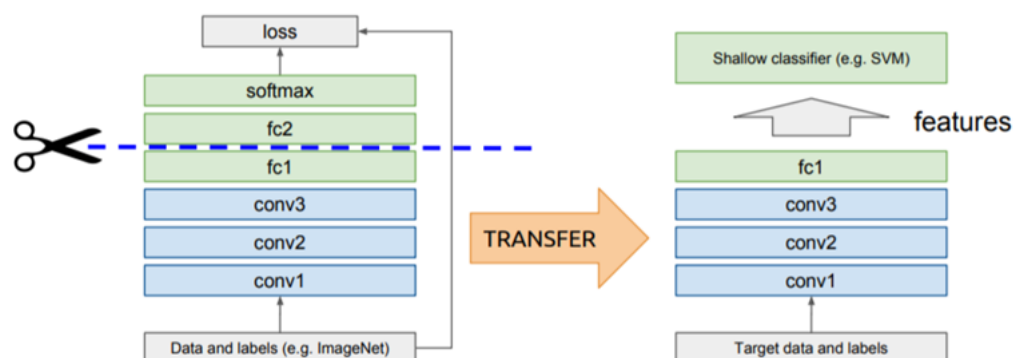


Figure 3.6: Transfer Learning Using Pre-Trained Deep Learning Model.

Figure 3.6 illustrates transfer learning using feature extractors that have already been trained on deep learning models. The essential concept is to retrain the model with new input for a new task by using the weighted layers of the previously trained model to extract features without altering the weights. Due to the fact that the pre-trained models were created using a sufficiently vast and diverse dataset, they may be effectively employed as a general model of the visual world [1].

Finally, let us show a flowchart of how transfer learning actually operates. In Figure 3.7 the transfer learning process flowchart is shown.

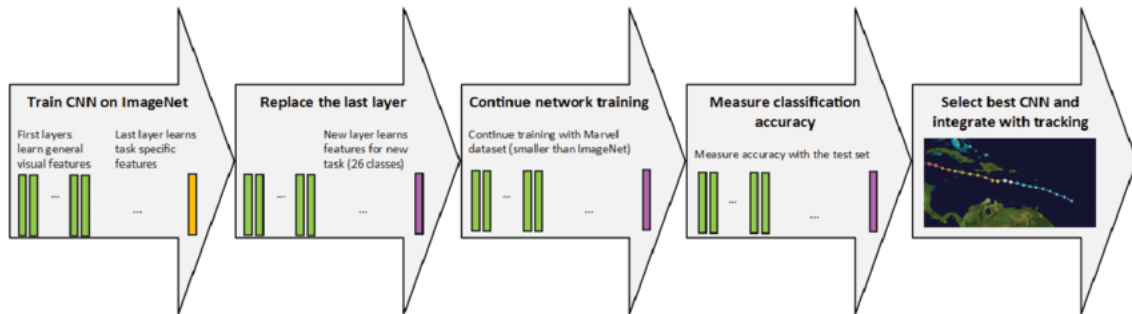


Figure 3.7: Transfer Learning Process.

3.4 VGG-19 Network Design

The VGG model has 19 layers in its variation, known as VGG-19. It consists of one SoftMax layer, three Fully Connected layers, five MaxPool layers, and sixteen convolution layers [15]. Also, VGG-19 has 19.6 billion FLOPs. However, there are further VGG variations, including VGG-11, VGG-16, and others.

3.4.1 Background of VGG-19

The conventional Convolutional neural networks were outclassed by AlexNet, which was released in 2012 [15]. Therefore, we may think of VGG as AlexNet’s successor even though a separate team at Oxford developed it called the Visual Geometry Group (VGG). In order to build on its predecessors’ concepts, the VGG employs deep convolutional neural layers. This gives the VGG its name.

Let’s first have a look at ImageNet and establish a basic understanding of CNN before moving on to the VGG19 Architecture.

3.4.2 ImageNet

The hierarchical organization of the 14,197,122 images in the ImageNet database follows that of WordNet. Support for image and vision researchers, academics, and other stakeholders is the goal of this initiative.

Researchers from all around the world were challenged to provide solutions for the ImageNet LargeScale Visual Recognition Challenge (ILSVRC), one of the contests that ImageNet supports, to achieve the lowest top-1 and top-5 error rates [6]. For

the competition, a validation set of 50,000 pictures, a test set of 150 000 photos, and a 1,000-class training set of 1.2 million photos are offered.

3.4.3 Convolutional Neural Network

The hierarchical organization of the 14,197,122 images in the ImageNet database follows that of WordNet. Support for image and vision researchers, academics, and other stakeholders is the goal of this initiative.

Researchers from all around the world were challenged to provide solutions for the ImageNet LargeScale Visual Recognition Challenge (ILSVRC), one of the contests that ImageNet supports, to achieve the lowest top-1 and top-5 error rates [6]. For the competition, a validation set of 50,000 pictures, a test set of 150 000 photos, and a 1,000-class training set of 1.2 million photos are offered [7].

Convolution: To identify characteristics in an image

ReLU: to smooth out the image and highlight borders

Pooling: a technique to correct distorted pictures

Flattening: transforming an image into a functional representation.

Full connection: to allow a neural network to handle data.

A CNN functions similarly to an ANN in most respects. However, because we are dealing with images, a CNN has more layers than an ANN [13]. While a CNN uses a multi-channeled picture as its input, an ANN uses a vector as its input.

3.4.4 VGG-19 Architecture

VGG is a sophisticated CNN that is used to categorize photos [13]. The VGG19 model's layers are as shown in the Figure 3.8 :

Architecture walk-through:

A fixed-size (224 * 224) RGB input was provided as input, suggesting that the matrix was formed (224,224,3).

The only pre-processing that was accomplished was to compute the average RGB value of each pixel across the entire training phase.

Made use of kernels with a stride size of 3 * 3 pixels so this allowed them to completely hide the idea of the image.

Spatial padding was used to preserve the spatial resolution of the image.

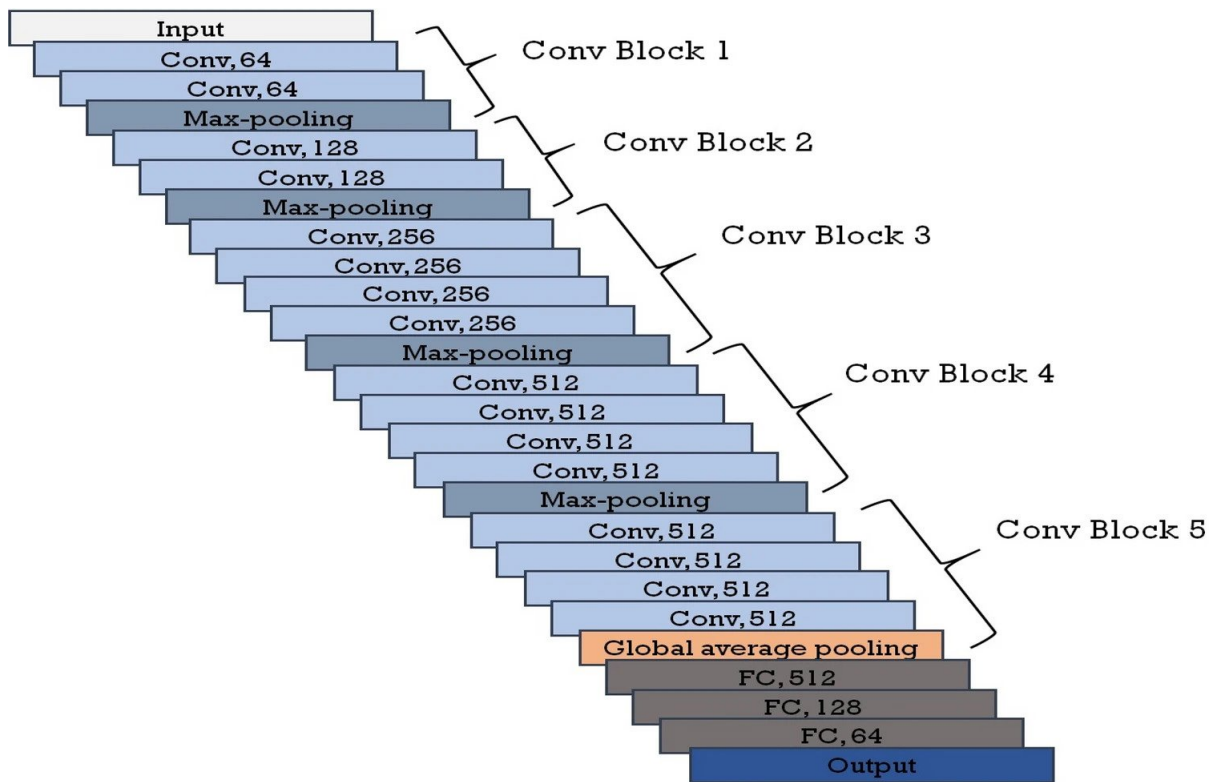


Figure 3.8: Layers of VGG-19.

Pooling stride 2 across $2 * 2$ -pixel windows, max-pooling was done.

A rectified linear unit was utilized to introduce nonlinear characteristics into the model, improving classification precision and computing efficiency [14]. This model outperformed earlier models that used tanh or sigmoid functions significantly .

Created three fully connected layers with a total size of 4096 for the first two, For the third layer's classification, 1000 channels were used with the 1000-way ILSVRC, and the final layer's softmax function was added for the last layer to find out the final output. [13].

Chapter 4

Alternative Approaches

4.1 YOLO-v5

One of AI developers' most widely used and preferred algorithms is YOLO, or "You Only Look Once." The core preference has always been real-time object detection. On June 9, 2020, the author, Glenn Jocher, published YOLOv5. Glenn unveiled a remarkable upgrade to YOLOv5 based on the PyTorch framework.

This version is really impressive and performs better than any of the previous versions, coming close to EfficientDet AP in terms of FPS. The graph of Figure 4.1 below demonstrates this [18].

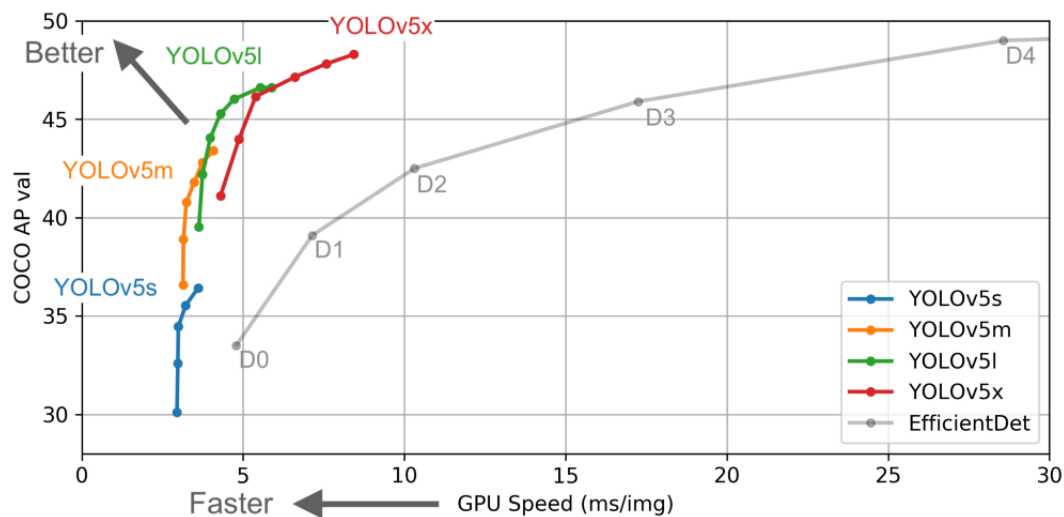


Figure 4.1: YOLOv5 Performance Comparison

Like other single-stage object detectors, YOLO v5 contains three vital components since it is a single-stage object detector.

i. Backbone: CSPDarknet

ii. Neck: PANet

iii. Head: YOLO Layer

Data is initially loaded into CSPDarknet for feature extraction, after which it is sent into PANet for feature fusion. Yolo Layer then outputs the results of the detection, including the class, score, position, and size. As shown in the Figure 4.2 the network architecture of YOLO-v5.

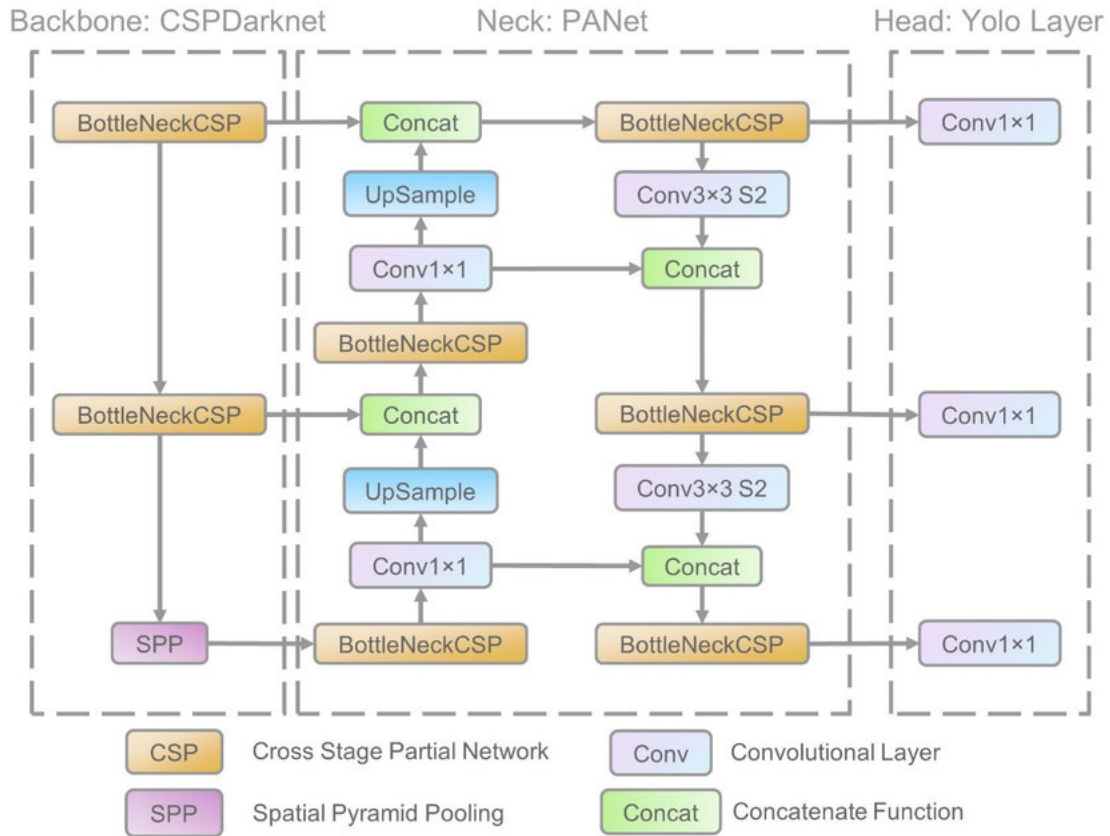


Figure 4.2: YOLOv5 Architecture

CSPNet are utilized as the framework in YOLOv5 to unsheathe highly informative features from an input picture. The basic purpose of Model Backbone is to derive important attributes from an input data.

Feature pyramids are mostly built using Model Neck. It allows models to resize elements successfully. It is advantageous to be able to recognize the same item in multiple scales and volumes. Furthermore, feature pyramids are indeed very effective and support the success of models on unobserved data. Other models use other feature pyramid approaches, including FPN, BiFPN, PANet, etc. PANet is utilized in YOLOv5 as a neck to get feature pyramids.

The model head in YOLOv5 is the same as the heads in YOLOv3 and YOLOv4 versions. The final detecting phase is mostly carried out using the model Head. A final output vector with class probabilities, objectness scores, and bounding boxes is produced when anchor boxes are applied to the feature.

Our model YOLOv5 is working for classification along with prediction. By this model we can detect our desired output. The thing is if the model detects a label and ground truth matches correctly we consider this as TP. On the other hand, if the model detects a label but the ground truth matches the wrong thing which is not a part of input we consider this as FP. Lastly, if the model does not detect any label we consider this as FN. If the model neither predicts the label nor comprises the ground truth we consider this as TN. The below table shows how the prediction is working in our model.

Actual class	Predicted class		
		Class=Fresh	Class=Rotten
Class=Fresh	True Positive(TP)	False Negative(FN)	
Class=Rotten	False Positive(FP)	True Negative(TN)	

Figure 4.3: Concept of TP, TN, FP, FN for YOLO Model

After getting the performance of TP and FN we analyze the value of Precision and Recall.

Precision gives us the idea of how much correct the model guessed when we have given the input by labeling them. In identifying samples as positive, it assesses the model's precision and accuracy. As it is using False positive for the calculation The attention is also on the Negative samples that were mistakenly categorized as positive.

The formula of the precision is:

$$Precision = TP / (TP + FP)$$

On the other hand, Recall is the proportion of Positive samples that have been correctly categorized to all Positive samples as a whole. More positive samples are discovered with higher recall.

The formula of Recall is:

$$Recall = TP / (TP + FN)$$

4.2 VGG-16

The CNN variation known as VGG16 is one of the most important computer vision models currently available. The designers of this model used an architecture with exceptionally small (3*3) convolution filters to evaluate the networks and improve the depth, displaying a significant improvement over the state-of-the-art installations.

With the depth extended to 16–19 weight layers, around 138 trainable variables were produced.

VGG16 was found to be the model with the most exceptional performance on the ImageNet dataset out of all the setups. Let’s look at the real architecture of this arrangement.

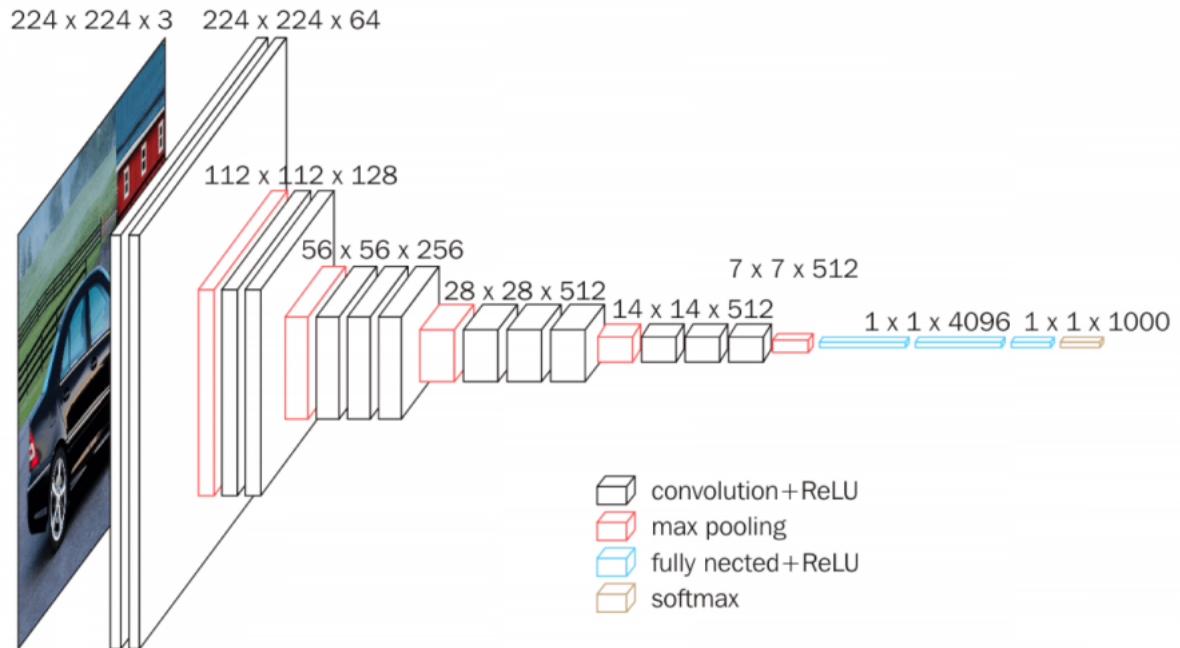


Figure 4.4: VGG-16 Architecture

As shown in Figure 4.3, the VGG-16 architecture and how the layers are structured one after another. Each network configuration takes into account an input of a constant size, (224×224) image with three channels—R, G, and B. The only pre-processing executed is to normalize the RGB values of each pixel. To do this, the mean value is deducted from each pixel [13].

ReLU activations are followed by passing the image through the initial stack of two convolution layers, whose receptive area is very modest (3×3). These two layers each include 64 filters. The padding comprises 1 pixel, but the convolution stride is constant at 1 pixel [5]. The spatial resolution is maintained in this arrangement, and the boundaries of the output activation map reflect those of the input image. Spatial max pooling is then performed on the activation maps adopting a (2×2) -pixel window with a 2. Spatial max pooling is then performed on the activation maps using a (2×2) -pixel window with a 2. The size of the activation is thereby split in half. The activation at the end of the first stack thus have the value $(112 \times 112 \times 64)$. Layers of VGG-16 are shown in Figure 4.4.

The installations then proceed via an additional stack that is similar to the first stack but includes 128 filters as opposed to the first stack’s 64 filters [5]. The size then changes after the second layer to $(56 \times 56 \times 128)$. The third stack is subsequently added, which consists of three convolutional layers and a max pool layer.

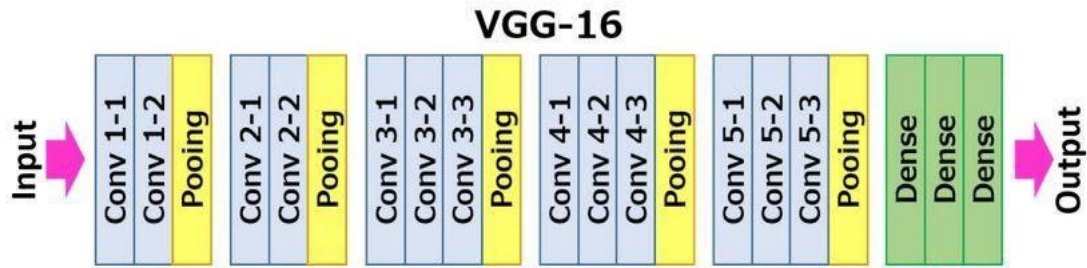


Figure 4.5: Layers of VGG-16

Because 256 filters were used in this instance, the output stack size is $(28 * 28 * 256)$. Following this, two stacks of three convolutional layers with 512 filters each are added. The result once both of these stacks has been completed is $(7 * 7 * 512)$.

The three fully connected layers that come after the convolutional layer clusters are sandwiched by a flattening layer [16]. 4,096 neurons are found in each of the first two levels. One thousand neurons make up the output layer's last fully connected layer, which represents the 1,000 potential classes in the ImageNet dataset. Following the output layer is the Softmax activation level, which is used for category classification.

When categorizing 1000 images into 1000 different categories, the object identification and classification algorithm VGG-16 has a 92.7% accuracy rate.

Chapter 5

Model Implementation & Working Plan

To conduct this study, a number of CNN models have been taken into consideration. But we settled on three models to achieve the best outcomes. Our taken-in VGG models are simple to use and comprehend. In order to compare the models, we also examined the YOLOv5 model.

5.1 Work Plan

The fruit freshness detection system's primary function is to ascertain the freshness of fruits. The input must be meticulously analyzed by this model in order to interpolate many fruit photos and produce the desired outcome. To estimate fruits' freshness, various images and image processing techniques are merged in an algorithmic methodology. The technique of identifying and highlighting distinguishing features in an image is known as image tagging. It is advantageous to automate metadata production. The system will be trained to recognize fruits and their freshness using a well-known deep learning approach. Open the image labeling tool and upload all raw pictures to be labeled. We will use this technique to annotate practice images in accordance with predefined standards. The dataset has to be set up and trained using the chosen algorithm when the labeling process is finished. Ensure the fruit detection is accurate, and the system has a high accuracy rate. The work plan is described in flowchart of Figure 5.1.

5.2 Implementation

Primarily, we apply knowledge distillation on our base/teacher model to make it less computation heavy and more accessible. For this experiment, transfer learning has been used to classify our data. Later on, a distillation class, a student model and a copy of the student model was constructed.

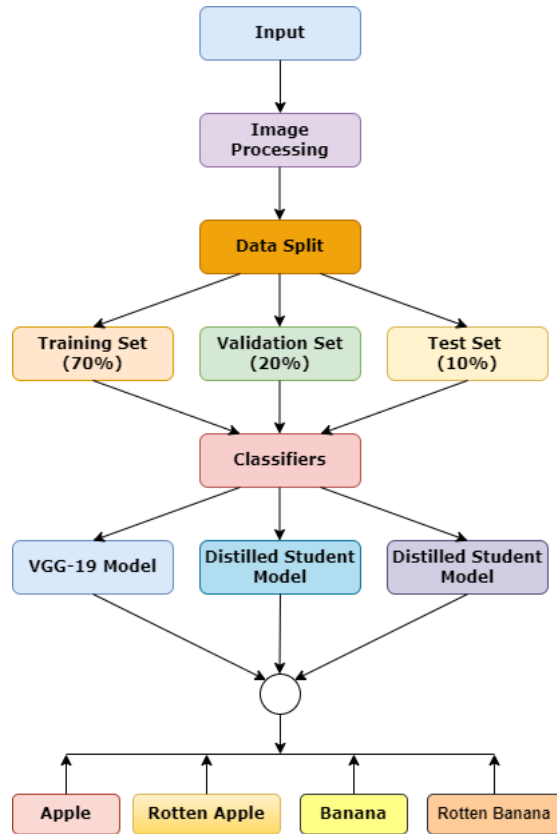


Figure 5.1: Work Plan.

5.2.1 Teacher Model

Teacher model has a large architecture which has complex layers and networks. The teacher model is trained first with the complete dataset. As it is a large model it requires some high computational performance for example high performance GPUs. In our work we used the VGG19 model as our Teacher Model. We have taken the input shape as (64, 64). After that we trained the teacher model with the full dataset. While training, we exclude the top three fully connected layers and use the Imagenet weights. Still there are almost 20 million parameters in this model. The model distribution is shown in Figure 5.2.

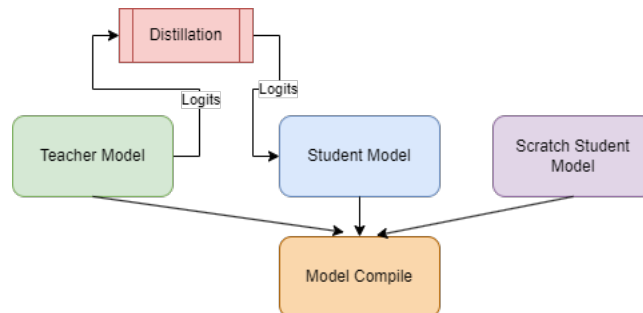


Figure 5.2: Model Distribution.

5.2.2 Distillation Function

In the distillation function, the teacher model is transferred by training on a transfer set. Using the complex system with a high altitude in its softmax to generate the transfer set, and applying a soft target distribution to each case in the transfer set. The same extremely hot temperature distilled model is trained at a temperature of 0, but once trained, it switches to 1.

As we have used VGG19 as a teacher model we have transferred this model into the distillation function after training the model. We have configured the distiller by optimizer, matrices, student loss fn, distillation loss fn, alpha and temperature. Optimizer is the Adam optimizer which is for student weight, Sparse Categorical Accuracy metrics for evaluation. Student loss fn is the difference between student prediction and ground-truth. Distillation loss fn the difference between student prediction and soft teacher prediction. Lastly, temperature is used for softening probability distributions. The higher the temperature the distribution will be softer.

5.2.3 Student Model

In the student model, we have taken the input shape same as (64, 64). This student model is a slightly modified version of VGG16 network with fewer filter numbers and small (1,1) and (3,3) kernel. There are 13 Convolution layers in this model. There are 5 layers which have kernel size of (1, 1) ,7 layers have (3,3) kernel size and one layer have (5, 5) kernel size. On the other hand, the VGG16 model has only (3, 3) kernel size. In this model we have taken the filter size 16 initially and after that we have increased the filter size gradually. But we can see in the VGG16 model initially the filter number is 64 and that's a large number which gets doubled after every convolution, increasing the number of parameters. As we have started from filter size 16 the computation will be decreased and the time and space can be saved by using our model. Lastly our model and VGG16 both were used by activation function "ReLU" in the initial stage and ended with "softmax" activation function in the very last layer.

5.2.4 Copy of Student Model

The previous student model will be compiled from the teacher model's data. But the "Copy of Student Model" is an independent model where we can test any kind of data. The main thing is this model is not dependent on the "Teacher model". We have created this copy version of student model so that we can evaluate the sole outcome of this model and also the distilled student model.

Chapter 6

Data sets

We have chosen to concentrate our research on two fruits, apple, and banana, as our thesis focuses on detecting fruit freshness. We have collected the images from various sources and classified into six classes: Apple, Medium Rotten Apple, Rotten Apple, Banana, Medium Rotten Banana and Rotten Banana are these. We have taken 4557 images for the knowledge distillation implementation. Additionally, for the YOLO model which will do classification along with prediction we have taken four classes which are Apple, Rotten Apple, Banana, Rotten Banana and taken 5176 annotated images. After augmentation the number of images increases to 12000.

6.1 Dataset Characteristics

The selection of fruit photos for this study was kept relatively simple. Images in the fresh fruit category were highly apparent since the fruit's texture appeared to be brand-new. On the other hand, the fruit images we looked at for the rotten category had substantial decomposition that was obvious on the outside texture. This indicates that fruits in a rotting state were considered inedible. The first collection only included RGB-based photos with a size of 364*270. The photos were flattened into 64*64 after the image dataset was taken since one of our goals is to create a system that can operate effectively on low-end devices.

6.2 Dataset Split

To perform this research, we divided the acquired dataset into training, validation, and test parts. The ratio is 7:2:1 (training: 70%, validation: 20%, and verification: 10%).

6.3 Input Dataset Processing

We initially put all the photos into a NumPy array and converted them to float32 format for dataset preparation. The array of photos was then split by 255 to achieve normalization. We normalize the data such that the picture array may only hold values between 0 and 1 instead of 0 and 255. Without converting these variables to 0 and 1, our model would be less effective and require more work. A simple image processing pipeline has been utilized to increase the diversity of the image

collection. This phase gives our model greater flexibility so that it can categorize photos taken from various viewpoints. We used ImageDataGenerator from Keras to process the images, and we adjusted a few features to get the most out of the raw dataset we were provided. Since the normalization has already been completed, we have omitted the rescale option from this section. In addition, channel color shift, random rotation, horizontal flip, shear range, noise addition, and in-picture cut-out are used.

Chapter 7

Result and Analysis

Here, we assess the outcomes of every model that we used in this research. The instructor model, the newly created student model, and the state-of-the-art YOLOv5 model were all compared. The instructor model, the newly formed student model, and the state-of-the-art YOLOv5 model were all compared. The largest network has 20 million trainable parameters and is called the teacher model (VGG19). The student model, a modified version of VGG16, has around 800k trainable parameters, though. In comparison to the bigger model, this has about 21 times fewer parameters. The YOLOv5x version of the algorithm includes about 7.5 million parameters.

7.1 Base/Teacher Model

We trained the base model using the VGG19 model, which has 19 layers total, and 16 convolutional layers with the ReLu Activation function. The model was run with a batch size of 32 across 50 epochs.



Figure 7.1: Training and Validation Accuracy of Teacher Model

As the graph of Figure 7.1 shows the accuracy and validation accuracy rate of the Teacher model(VGG19). We can observe that after a few epochs, the accuracy reached a maximum of 90%. One of the causes is that the images were simple to differentiate. The training accuracy is 99.86%, and the validation accuracy is 98.08% after 50 epochs. The model is neither overfit nor underfit, one could say.

The Training and Validation Loss graph shows a progressive decrease in loss. This teacher model has logits set to true, Sparse Categorical Cross entropy as the Loss function, and Adam as the optimizer. There has been a noticeable improvement in the loss function for training and validation. Finally, the validation loss is 0.0688, whereas the training loss is 0.0189. In Figure 7.2, the graph shows training loss and validation loss of the Teacher Model.



Figure 7.2: Training and Validation Loss of Teacher Model

7.2 Student/ Distilled Model

With minor changes to its kernel size and filter units, the distillation model is based on the layered architecture of VGG16. In comparison to the base VGG19 model, this model has nearly 20 times fewer hyperparameters. This model also underwent a distillation process using the logits from the teacher model. Similar to the teacher model, the (alpha) was set to 0.1, the temperature to 5, and the batch size to 32 and 50 epochs. Accuracy rate and validation accuracy rate of the Student model is shown in the graph of Figure 7.3.

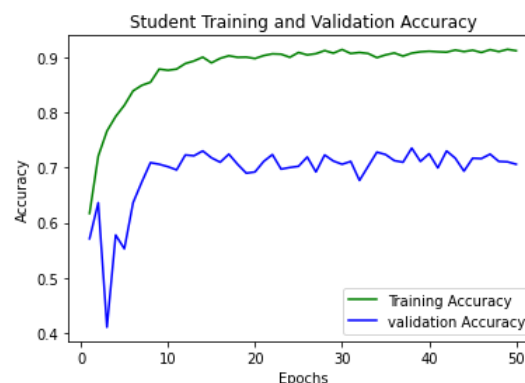


Figure 7.3: Student Training and Validation Accuracy

The distilled Student model showed some discrepancies after reaching 30 epochs. Other than this, the distilled model did an excellent job of seamlessly training the

supplied data. Peak training accuracy for the student model is 98.02%, while peak validation accuracy is 94.23%. Training loss and validation loss of the student model is shown in the graph of Figure 7.4.

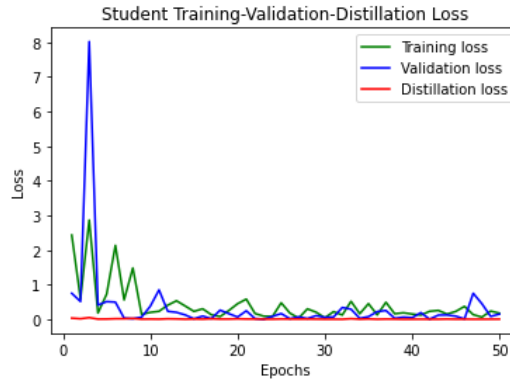


Figure 7.4: Training and Validation Loss of Student Model

A few spikes are seen throughout the training period due to the loss of the distilled and modified version of VGG16. The distillation loss curve in the above figure shows how the training loss error using KL Divergence increases, leading to a better validation loss curve. In contrast to the distillation loss, which is 0.0030, the loss for the Student model is 0.0450.

7.3 Scratch Student Model

We created a student model from scratch that is not distilled for comparison to assess the performance of the scratch model without the additional distillation from the extensive network. As shown in Figure 7.5 the graph of accuracy rate and validation accuracy rate of scratch student model.



Figure 7.5: Scratch Student Training and Validation Accuracy

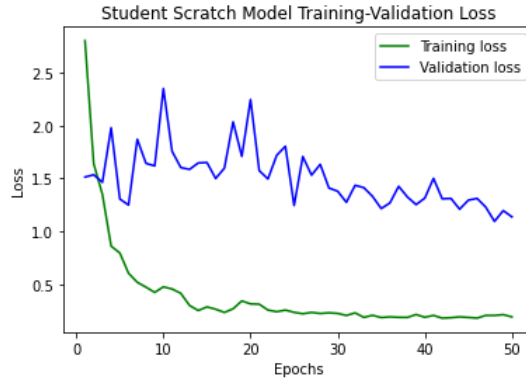


Figure 7.6: Training Loss and validation Loss of Scratch Student Model.

In Figure 7.6, the graph shows the training loss and validation loss of scratch student model. The distilled student model performs somewhat better than the student scratch model. Peak training accuracy for the aforementioned model is 96.33%, while peak validation accuracy is 96.15%. This model exhibits more loss during training and validation than the Distilled Student model.

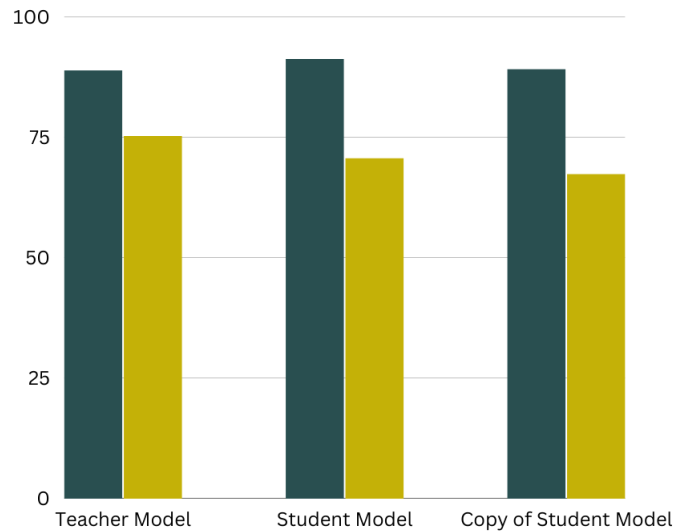


Figure 7.7: Comparison Between Constructed Models

The comparison between teacher model, student model and distilled student model is shown in Figure 7.7.

Model-wise Accuracy Comparison: In Figure 7.8 the table shows the comparison according accuracy of models.

No.	Model	Accuracy
1	Teacher Model (VGG19)	88.80%
2	Distilled Student Model (Modified VGG16)	91.18%
3	Scratch Student Model	89.07%

Figure 7.8: Comparison Table

As an additional research, we have implemented our dataset on two versions of YOLOv5 which are YOLOv5-s, YOLOv5-L. We conducted the training with 100 epochs and batch size of 32, we have achieved a prediction model for detecting fresh and rotten fruits.

Model	Epochs	Parameters	GFLOPS	GPU Memory	mAP 0.5	mAP_0.95	Precision	Recall
YOLOv5-L	100	86,238,001	204.7	13.7gb	92%	63.2%	91.6%	86.16%
YOLO v5-s	100	7,254,609	16.8	4.6gb	89%	59.67%	89.3%	85.2%

Figure 7.9: YOLOv5 Model Comparison

Figure 7.9 denotes the comparison between both the tested YOLOv5 models. YOLOv5-L is using GPU memory of 13.7 gb in 6.20 hours. It is taking 86238001 parameters and GFLOPs of 204.7. We have got a precision score of 91.6% For YOLOv5-L. Similarly, the recall score for the model is 86.16% with mAP 0.5 of 63.2%. On the other hand, the smaller version of the YOLOv5 model is using only 4.6gb of GPU memory, taking only 2.66 hours. The smaller model achieved a precision, recall score of 89.3% and 85.2% with mAP 0.5 of 89%. Comparing both the models, it is evident that the smaller model is performing well in terms of resource utilization. It is taking less time to train than other models and the inference time of the smaller model is 0.009s on average. Which is approximately 110 frames per second. However, more training time and more diverse dataset could lead up to better accuracy for the model.

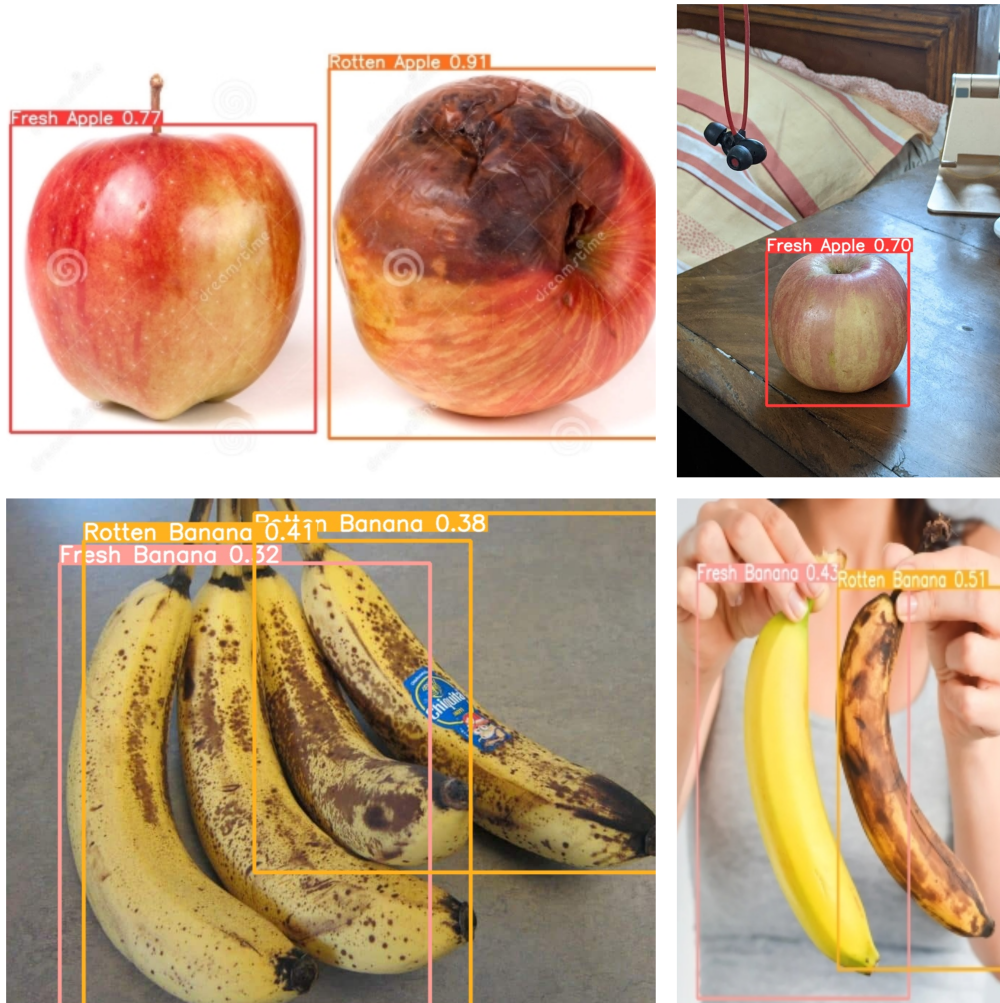


Figure 7.10: Inference test of The YOLOv5s

Figure 7.10 shows the inference test of the YOLOv5 small model. The model shows the desired output and how it could differentiate between rotten and fresh fruit. Along with the fresh and rotten detection, the model delivers a confidence score with every detection.

Chapter 8

Conclusion and Future Works

In conclusion, we may summarize the difficulties with deep learning models and fruit freshness detection. To improve training outcomes, we first examine the collection of fruit images as well as image augmentation and improvement. One of the transfer learning models we employed is a Convolutional Neural Network model called VGG19. Then, using a more extensive, already-trained network, we gradually educated a smaller network on precisely what to do, referred to as knowledge distillation. As a consequence, the model provided us with high accuracy. The accuracy of VGG19 is 88.80%, compared to the accuracy of our distilled modified VGG16 model, which is 91.18%. We chose to utilize the distilled student model instead of the scratch student model, which has 89.07% accuracy. The student model takes up less space than others, deploys more quickly, and is simpler to use on low-powered devices because it contains capabilities like VGG19.

We intend to continue working on picture segmentation to distinguish fresh fruit in the future. In YOLOv5, we will also work on knowledge distillation. Mainly because YOLOv5 is more efficient and sophisticated than VGG19. To apply this technology in wholesale industries and juice factories, we will also try to create a mobile application using our work. With the model being adjusted and tested, we will undoubtedly work to make our system more efficient.

Bibliography

- [1] Y. Bengio, “Deep learning of representations for unsupervised and transfer learning,” in *Proceedings of ICML workshop on unsupervised and transfer learning*, JMLR Workshop and Conference Proceedings, 2012, pp. 17–36.
- [2] M. T. Bahadori, Y. Liu, and D. Zhang, “A general framework for scalable transductive transfer learning,” *Knowledge and information systems*, vol. 38, no. 1, pp. 61–83, 2014.
- [3] L. Wang, S. Guo, W. Huang, and Y. Qiao, “Places205-vggnet models for scene recognition,” *arXiv preprint arXiv:1508.01667*, 2015.
- [4] Y. Kim and A. M. Rush, “Sequence-level knowledge distillation,” *arXiv preprint arXiv:1606.07947*, 2016.
- [5] H. Ide and T. Kurita, “Improvement of learning for cnn with relu activation by sparse regularization,” in *2017 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2017, pp. 2684–2691.
- [6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [7] W. Yin, K. Kann, M. Yu, and H. Schütze, “Comparative study of cnn and rnn for natural language processing,” *arXiv preprint arXiv:1702.01923*, 2017.
- [8] K. Jayasankar, B. Karthika, T. Jeyashree, R. Deepalakshmi, and G. Karthika, “Fruit freshness detection using raspberry pi,” *International Journal of Pure and Applied Mathematics*, vol. 119, no. 4, pp. 1685–1691, 2018.
- [9] D. Sarkar, “A comprehensive hands-on guide to transfer learning with real-world applications in deep learning,” *Towards Data Science*, vol. 20, p. 2020, 2018.
- [10] J. H. Cho and B. Hariharan, “On the efficacy of knowledge distillation,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 4794–4802.
- [11] Y. Fu, “Fruit freshness grading using deep learning,” Ph.D. dissertation, Auckland University of Technology, 2020.
- [12] F. Zhuang, Z. Qi, K. Duan, *et al.*, “A comprehensive survey on transfer learning,” *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.
- [13] L. Alzubaidi, J. Zhang, A. J. Humaidi, *et al.*, “Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions,” *Journal of big Data*, vol. 8, no. 1, pp. 1–74, 2021.

- [14] A. V. Ikechukwu, S. Murali, R. Deepu, and R. Shivamurthy, “Resnet-50 vs vgg-19 vs training from scratch: A comparative analysis of the segmentation and classification of pneumonia from chest x-ray images,” *Global Transitions Proceedings*, vol. 2, no. 2, pp. 375–381, 2021.
- [15] M. S. M. Khan, M. Ahmed, R. Z. Rasel, and M. M. Khan, “Cataract detection using convolutional neural network with vgg-19 model,” in *2021 IEEE World AI IoT Congress (AIIoT)*, IEEE, 2021, pp. 0209–0212.
- [16] S. Mascarenhas and M. Agarwal, “A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification,” in *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, IEEE, vol. 1, 2021, pp. 96–99.
- [17] F. Valentino, T. W. Cenggoro, and B. Pardamean, “A design of deep learning experimentation for fruit freshness detection,” in *IOP Conference Series: Earth and Environmental Science*, IOP Publishing, vol. 794, 2021, p. 012110.
- [18] W. Wu, H. Liu, L. Li, *et al.*, “Application of local fully convolutional neural network combined with yolo v5 algorithm in small target detection of remote sensing image,” *PloS one*, vol. 16, no. 10, e0259283, 2021.
- [19] N. Ismail and O. A. Malik, “Real-time visual inspection system for grading fruits using computer vision and deep learning techniques,” *Information Processing in Agriculture*, vol. 9, no. 3, pp. 24–37, 2022.
- [20] T. B. Kumar, D. Prashar, G. Vaidya, V. Kumar, S. Kumar, and F. Sammy, “A novel model to detect and classify fresh and damaged fruits to reduce food waste using a deep learning technique,” *Journal of Food Quality*, vol. 2022, 2022.