MACHINE LEARNING BASED AGE PREDICTION OF COCONUT

Submitted by

Parvathy H

(Reg. No: 31020024)

In partial fulfillment of the requirements for the award of

Master of Science in Computer Science with Specialization in Machine Intelligence

of

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY, KOCHI

Conducted by

Indian Institute of Information Technology and Management-Kerala

Technopark Campus

Thiruvanathapuram-695 581

May 2022

MACHINE LEARNING BASED AGE PREDICTION OF COCONUT

Submitted by

Parvathy H

(Reg. No: 31020024)

In partial fulfillment of the requirements for the award of

Master of Science in Computer Science with Specialization in Machine Intelligence

of

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY, KOCHI

Conducted by

Indian Institute of Information Technology and Management-Kerala

Technopark Campus

Thiruvanathapuram-695 581

May 2022

BONAFIDE CERTIFICATE

This is to certify that the project report entitled "MACHINE LEARNING BASED AGE PREDICTION OF COCONUT" submitted by Parvathy H (Reg. No: 31020024) in partial fulfillment of the requirements for the award of Master of Science in Computer Science with Specialization in Machine Intelligence is a bonafide record of the work carried out at Indian Institute of Information Technology and Management - Kerala under our supervision.

Supervisor

Dr. Jose Joseph

Assistant Professor

IIITM-K

Course Coordinator

Dr. Asharaf S

Professor

IIITM-K

Director

Prof. Saji Gopinath

Director

IIITM-K

Declaration

I, PARVATHY H a student of MSc Computer Science with specialization in Machine Intelligence, hereby declare that this report is substantially the result of my own work , except, where explicitly indicated in the text and has been carried out during the period March 2022-June 2022.

Place: Thiruvananthapuram

Date: 06/06/2022

Parvathy H

ACKNOWLEDGEMENT

I take this opportunity to express my deep sense of gratitude and sincere thanks to all who helped me to complete the work successfully. My first and foremost thanks go to God Almighty, who showered immense blessings on my effort.

I wish to express my sincere thanks to Dr Jose Joseph for his guidance and for providing all the necessary facilities and support. I would also like to thank Dr Rajesh M.K, Principal Scientist, CPCRI, for providing image data and insights on the problem attempted to address.

I wish to express my sincere gratitude to all the teaching and non-teaching staff members of my Department. Finally, I thank my parents, all my friends, and near and dear ones who have directly and indirectly contributed to the success of this work.

Parvathy H

ABSTRACT

Coconut is Kerala's most important cultivated crop, accounting for 39 percent of the state's net sown area, according to 2013-14 statistics. The lack of skilled labor and high wage rate pose challenges to coconut farming in the state. Better technology integration can aid laborers and farmers in enhancing the productivity of coconut cultivation. Coconuts at different stages of development are suitable for different purposes. Coconuts in seven to nine months of age are considered tender and are harvested for drinking purposes, and those with maturity greater than twelve months are harvested for commercial nut production. Thus there is a need to classify coconuts based on their age. Specific expertise is required to ascertain the age of coconuts. Traditional methods include time-consuming, cumbersome processes and rely upon skilled palm climbers and farmers capable of finding the age of coconuts. Machine learning can help predict the age of coconuts and thus can address this problem by enabling anyone to identify coconut as tender or mature easily. This project aims to classify coconuts according to their age using machine learning techniques and image data. Further, a suitably trained machine learning model is incorporated into an Android app for ease of use.

Contents

List of Tables

List of Figures

List of Symbols and Abbreviations

$BiFPN$ Bi-directional Feature Pyramid Network

CNN Convolutional Neural Network

CV Computer Vision

DL Deep Lerning

GPU Graphics Processing Unit

$ICAR - CPCRI$ Indian Council of Agricultural Research - Central Plantation Crops Re- search Institute

$ILSVRC$ ImageNet Large Scale Visual Recognition Challenge

$R - CNN$ Region-Based Convolutional Neural Network

RPN Region Proposal Network

Chapter 1

Introduction

Smart farming approaches driven by machine learning are revolutionizing all phases of agriculture like crop yield prediction, disease detection, soil parameter prediction, and harvesting, to name a few. These smart techniques can help address challenges like fulfilling the high food demands of the increasing population.[1]

Computer vision is a sub-field of machine learning that brings visual capabilities to a system. Image classification and object detection are two usual tasks in computer vision and transfer learning based classification and detection methods are suitable for fruit identification, localization, maturity detection, and classification. Deep learning based computer vision techniques have enabled intelligent harvesting systems based on information from images.[2]

Coconut is Kerala's most important cultivated crop, accounting for 39 percent of the state's net sown area, according to 2013-14 statistics. The lack of skilled labor and high wage rate pose challenges to coconut farming in the state. Better technology integration can aid laborers and farmers in enhancing the productivity of coconut cultivation. [3]

Despite the high demand for coconut, farmers face the prospect of severe financial losses due to the challenges involved in coconut harvesting, like low labour supply and high labour cost due to crop height and its complex trunk structure.[4]

1.1 Motivation

Coconuts at different stages of development are suitable for different purposes. At seven to nine months of age, coconuts are considered tender and are harvested for drinking purposes. Those with a maturity greater than twelve months are harvested for commercial nut production [4]. In addition to this, coconut at different ages are used in varied value added products.

Coconut is considered a healthy food and hence there is a high demand for shifting the market towards value-added products such as coconut milk, coconut ice cream, coconut cream, coconut vinegar, coconut water concentrate, packaged tender coconut water, nata-de-coco, coconut cream, coconut flour, desiccated coconut, coconut protein powder, and the like [5]. All these

different products need coconuts belonging to different weeks or months of age. High expertise is required to correctly predict the age of coconut in such an intricate manner.

Ascertaining the correct age of coconuts will help farmers in the timely harvesting of coconuts and to supply them to manufacturers of value-added products. This can open up better commercial possibilities for coconut cultivation. Automatic age prediction can also be a stepping stone for fully automated coconut harvesting by drones.

Thus there is a need to classify coconuts based on their age. Specific expertise is required to ascertain the age of coconuts. A machine learning powered software application that can predict the age of coconut correctly can be of great use for palm climbers, coconut farmers and for automating coconut harvesting in the future.

1.2 Aim, Scope and Approach

Traditional methods for determining the age of coconuts are time-consuming and cumbersome in nature. This project aims to classify coconuts according to their age into tender or mature using machine learning techniques and image data. Transfer learning based solutions are[6] explored in the project. Further, a suitably trained machine learning model can be incorporated into an Android app for ease of use, decreasing the dependency on highly skilled

palm climbers and farmers to find the age of coconuts.

The initial approach is to consider the problem as a classification task based on only the image data that was made available. Application of pre-trained models based on VGG16, ResNet50, and EfficientNet-Lite architectures are analyzed to classify coconut as tender or mature.

A simple classifier may not be sufficient to perform the prediction correctly in more natural settings. The presence of leaflets, leaf stalks, and the background can contribute to the scene [4]. Hence first, detecting coconut from its background is needed before classifying it. The similarity between coconut and its background makes the detection of coconuts in their natural environment very challenging. Object detection based on EfficientDet and Faster-RCNN can be studied in this regard.

Chapter 2

Literature Review

2.1 Related Work

Thamban C. et al. talk about the coconut production landscape in Kerala, its challenges, how technology adoption can improve productivity, and the role of research institutions like ICAR- CPCRI in that.[3]

In a survey by Chiagoziem C. Ukwuoma et al.[7], they discuss how advancements in computer vision have impacted agriculture and related areas, especially the complex problem of fruit detection and classification. They talk in detail about how deep learning has become a popular choice of technology to perform fruit classification due to its ability to extract image characteristics and then introduce classification effectively.

A Faster R-CNN with ResNet-50 model that can enhance the performance of

coconut detection in two major maturation stages is proposed by Parvathi Subramanian and Tamil Selvi Sankar[4]. Research work focused on autonomous recognition of coconut as tender and mature using seven different CNN architectures - VGG16, VGG19, Xception, MobileNet, InceptionV3, InceptionResNetV2, and ResNet50 was discussed by Parvathi Subramanian and Tamil Selvi Sankar[6].

Usage of a Des-YOLO algorithm, which results in fewer network parameters and improved detection speed for real-time detection of apples, can help harvesting robots detect apples quickly and accurately in a complex environment, is proposed by Chen et al.[8]. A robust classification model based on modified ResNet29 architecture that can better generalize indoor and outdoor muskmelon stages of maturity and applies to mobile phone and automatic sorting system is introduced by Huamin Zhao et al.[9]

Siddesha S and S K Niranjan proposed a model for classifying coconuts by color and texture features using SVM classifier [10].Classification of real coconut datasets based on acoustic signals acquired through a tapping system and artificial neural network (ANN), random forest (RF), and support vector machine (SVM) is discussed by June Anne Caladcad et al.[11]. A fuzzy logic based method that can categorize coconut maturity stages that takes in fuzzy inputs of colour and sound is introduced by Javel et al.[12].

YOLOv4 based coconut detection model trained on coconut video dataset

that can be operated on edge devices is introduced by Joshi et al. [13]. Here they have prepared a drone based video dataset for coconuts that captures natural settings, true distribution of image data, shadow and natural conditions. A modified YOLOv3 tiny model with a DenseNet based feature extractor for automatic detection of oil palm fruits that tries to improve detection and computation performance so that it can be in an embedded system and is trained on images captured using UAV taken is proposed by Junos et al. [14].

YOLO-Grape, an efficient grape detection model based on the YOLOv4-tiny network that can identify six varieties of grapes with high accuracy, is proposed by Li et al.[15]. Here, they have added an attention mechanism to the network and have used a soft non-maximum suppression method. Machine learning and image processing techniques have been used by Kallapur et al. for the detection of aromatic coconuts [16]. Here they rely upon the principle that the colour of the region of interest at the bottom part of the coconut shell is correlated to its age. Computer vision techniques were applied by Lim et al. [17] for the evaluation of the quality of de-husked coconut post-harvest.

2.2 Theory

2.2.1 Computer Vision

Computer vision (CV) can be seen as a vast and multidisciplinary field that tries to give computers the ability to see, understand the meaning of images and videos, and take action based on that. Computer vision differs from image processing because while image processing creates a new simplified or enhanced image from an existing one, computer vision is concerned with understanding the contents of an image [18].

Object classification and object detection are two common tasks in computer vision.Classification aims to determine what category of an object is present in an image. Detection strives to find where the objects are in the image.

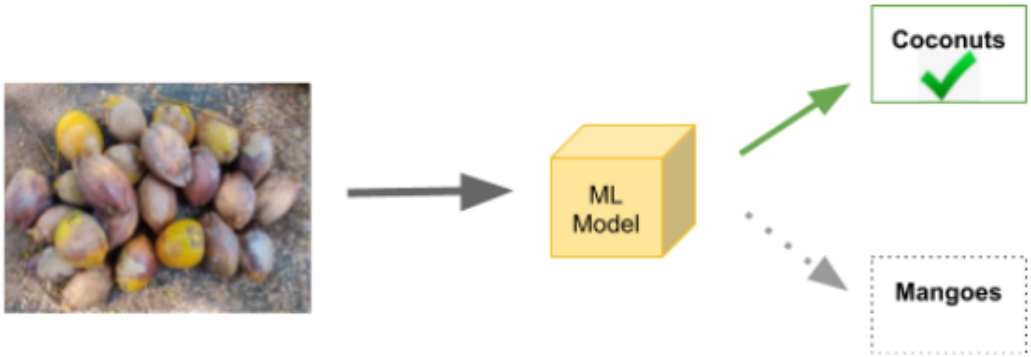

Figure 2.1: Object classification

.

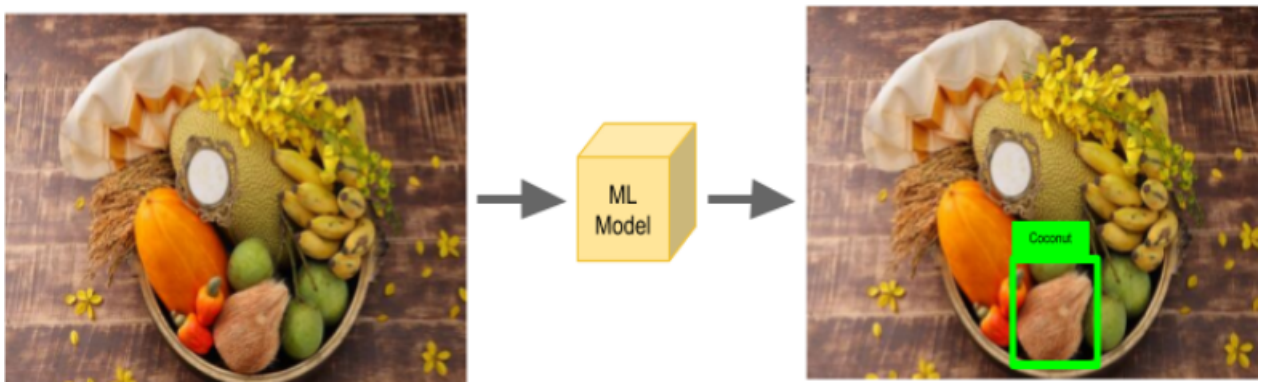

Figure 2.2: Object detection

.

2.2.2 Convolutional Neural Network

Deep learning is a subdomain of machine learning where the system learns the features for itself from the data input provided. Deep learning methods have become popular for CV mainly because they enable automatic feature extraction from images. Deep neural network architectures can have the capacity to learn more complex features.[18].

Several CNN architectures have come up over the decade that has been extensively applied to computer vision tasks. In the field of DL, CNN is the most popular choice of algorithm mainly due to its ability to automatically identify features without human intervention [19].CNN structure was inspired by the sequence of cells forming the visual cortex of the animal brain. CNNs have several layers, and a typical CNN architecture is shown below, with the different layers like the convolutional layer, non-linearity

layer, pooling layer, fully connected layer and softmax layer. [20][19][21].

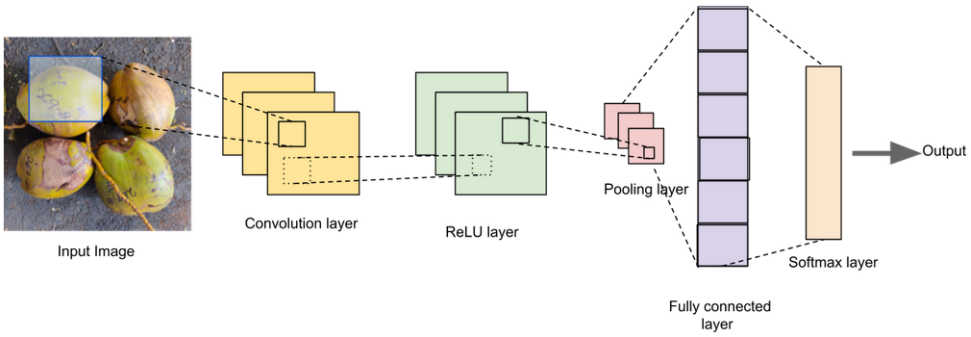

Figure 2.3: Layers of a typical Convolutional Neural Network

.

The convolutional layer is essential in a CNN architecture and performs convolution on the input image based on filters or kernels to perform feature extraction from the input.A filter or kernel comprises a grid of discrete numbers called kernel weights. Weights are initialized randomly and are adjusted at each training epoch to better extract features. The kernel slides over the input image, calculating the dot product representing the output's feature map. For example, for a 2x2 kernel, this convolution operation follows concepts described by the following figure.Non-linearity layer or activation function maps the input to output in a non-linear fashion. Sigmoid, ReLU, and Tanh are usual activation functions used in CNNs.Pooling layers help to sub-sample feature maps. Common pooling methods include max pooling, min pooling, average pooling, and global average pooling [19].

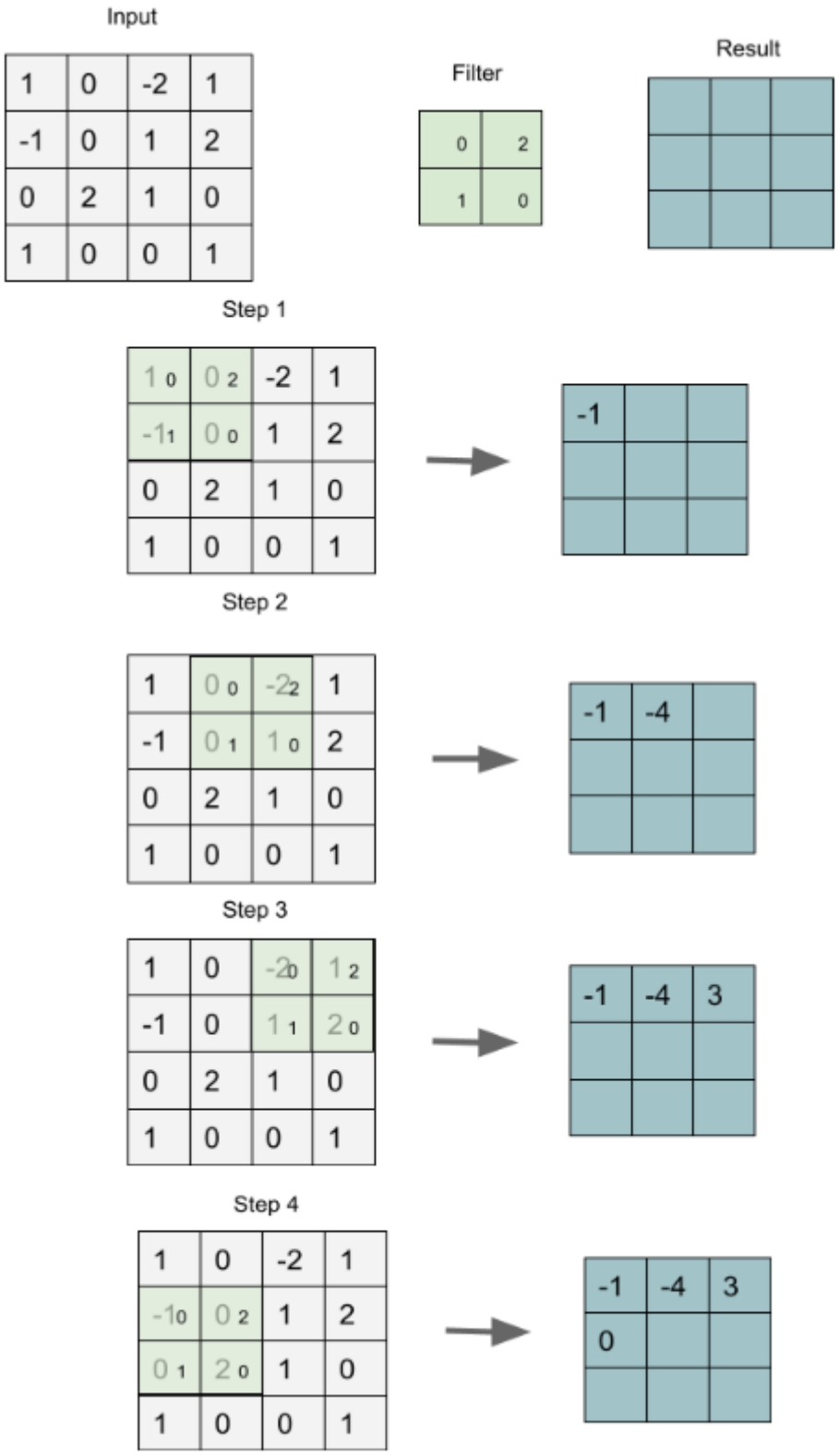

Figure 2.4: Convolution with kernel to extract feature map - concept 11

.

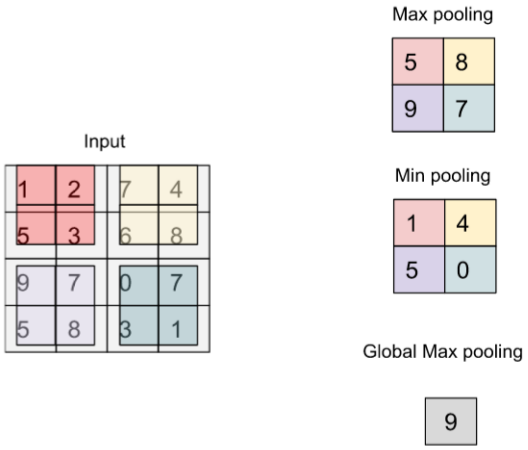

Figure 2.5: Pooling types and concept

.

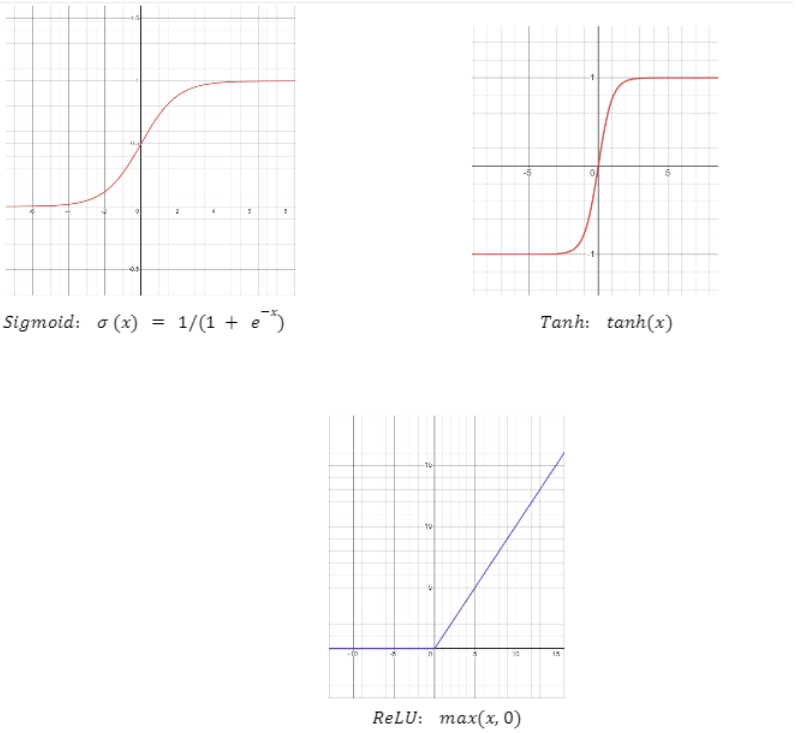

Figure 2.6: Common activation functions

Fully connected layers are typically placed at the end, and each neuron in this layer is connected to all neurons of the previous layer [19]. This layer acts as a classifier. The softmax layer produces the distribution of probability over the classes as the output [22].

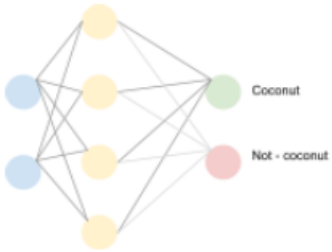

Figure 2.7: Fully connected layer

2.2.3 Transfer Learning

Transfer learning is a machine learning methodology that aspires to solve problems arising from the scarcity of data instances by transferring knowledge between domains. It aims to weigh in on knowledge from a source domain to enhance learning performance or minimize labelled data examples needed at the target domain. [23].

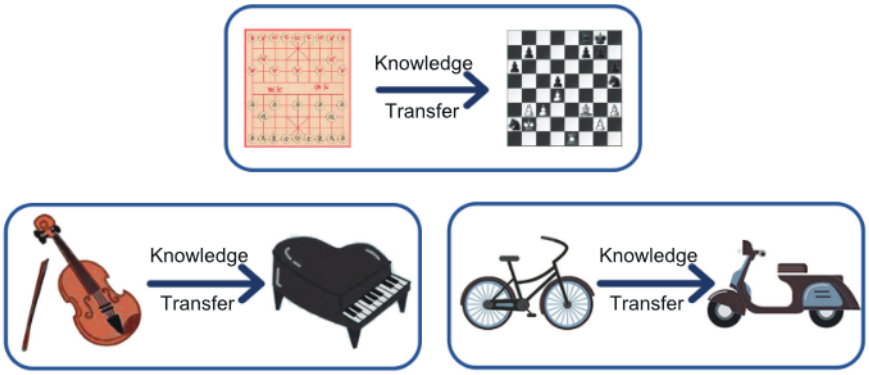

Figure 2.8: Transfer learning - intuitive examples [23]

A lack of similarity between the source and target domain can cause knowledge transfer to fail. It is worth noting that similarities between the domains do not always ensure proper learning and could be misleading in some cases. A person with extensive musical knowledge gained through playing the violin can learn to play the piano more efficiently than a person with no musical knowledge.

Transfer learning helps create high-performance learners with more easily obtained data. In scenarios where target data is rare, expensive to collect or label, or is inaccessible, transfer learning's need arises. It also relaxes the requirement that feature distribution in training and testing data has to be similar [24] [23].

In a common approach for transfer learning, CNN models trained on a large dataset (like ImageNet) in one domain are taken as base models. The lower

or the layers closer to the input would have learnt more general features while those towards the end, features more specific to the data. Hence, replacing the original classifier layer with a new classifier to fit target data can avoid the need to train a huge CNN model. The learnings of base layers can be transferred to the task at hand. For this, the layers of the base model which are not to be trained are frozen so that they will not lose the learned weights, and those layers to be trained are unfrozen [25].

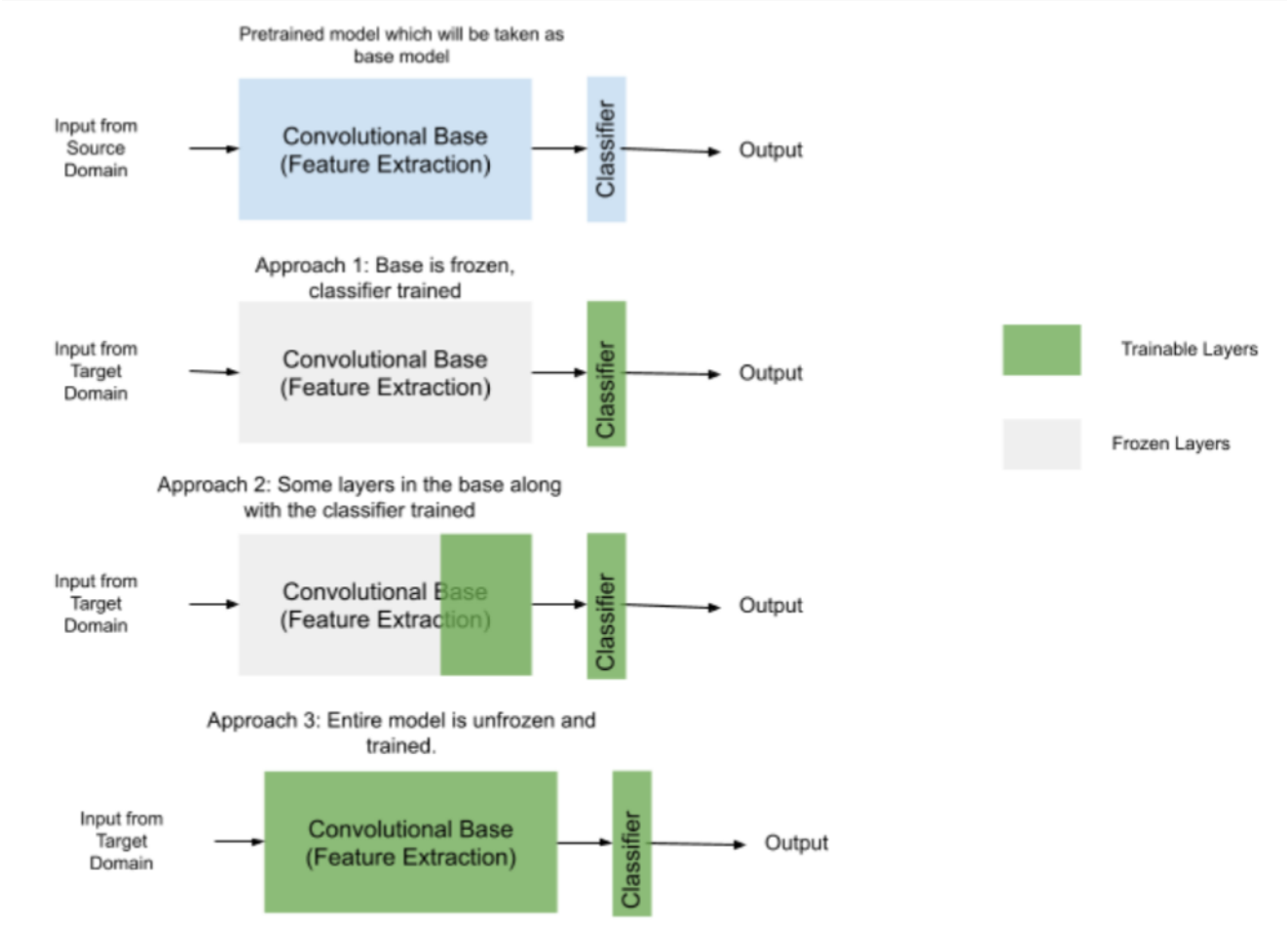

Figure 2.9: Transfer learning approaches [25]

2.2.4 Deep CNN Architectures

CNNs have proven to be among the best machine learning algorithms for many computer vision tasks like image classification, segmentation and detection. Major computer vision competitions see the front-runners leveraging the power of deep CNNs, and novel CNN architecture development is an active research area [26]. Some of the deep CNN architectures explored in this project include VGG, ResNet, Faster R-CNN and EfficienDet.

2.2.4.1 VGG

VGG architecure is from the Visual Geometry Group, Oxford University and secured first place for image localization and second place for image classification in the ImageNet Challenge 2014. VGG architecture was able to acheive better performance by increasing depth of the network to 16 to 19 weight layers. Very small convolution filters of size 3 x 3 is used here. These factors helped VGG to produce better accuracy on ILSVRC and other image recognition dataset. [27].The CNN layer architecture in VGG is inspired by Ciresan et al. (2011) [28] ; Krizhevsky et al. (2012) [29]. [25].

Network	A, A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

Table 2.1: Number of Parameters for VGG Architecture in Millions[27]

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 2.10: ConvNet configurations of VGG Architecture.ReLU activation function not shown for breivity [27]

VGG architecture is computationally costly due to many parameters and convolutional layers. VGG-16 network has 16 layers. Out of these 16, 13

are convolutional layers, and 3 are max-pooling layers, as suggested by the previous figure. ReLU activation is applied between these convolutional and max-pooling layers. Parallelising VGG-16 for embedded systems is very challenging due to the high number of parameters involved [30].

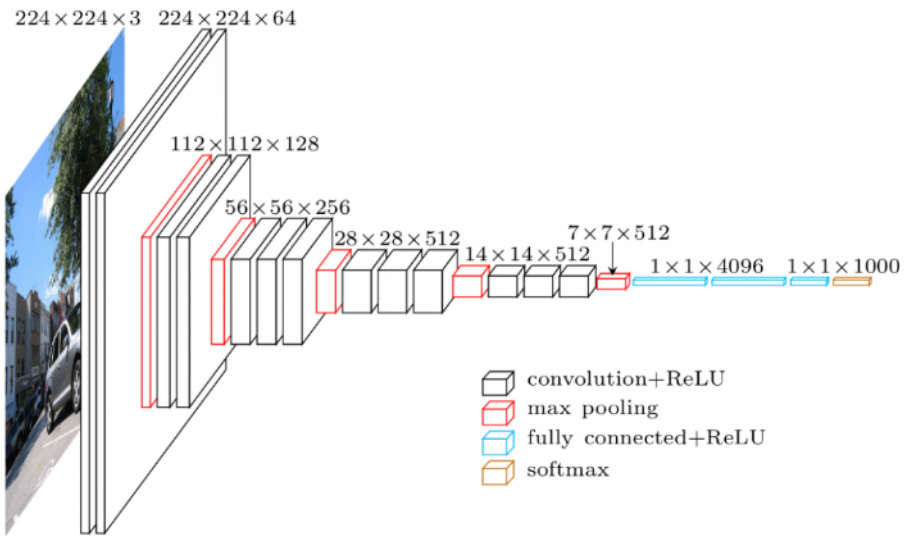

Figure 2.11: VGG -16 architecture [30]

2.2.4.2 ResNet

ResNet or Residual Network won the ILSVRC 2015. As the depth of neural networks increases, the difficulty of training them also increases. ResNet introduces a residual learning framework which reformulates the layers to learn residual function concerning current layer input.

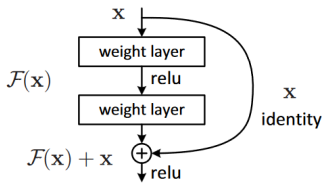

Figure 2.12: Building block of residual learning [31]

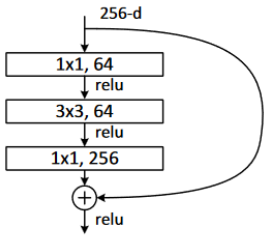

Figure 2.13: "Bottleneck" building block used in ResNet50 [31]

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 2.14: ResNet Configurations [31]

Simply stacking more layers need not provide better accuracy as this leads to vanishing/exploding gradients problems. The deep residual framework in ResNet tackles this degradation of training accuracy. Skip connections and residual mappings help ResNet to do this. This is because it is easier to push residual to zero than to fit an identity mapping by a stack of non-linear layers.

$H(x)$	Desired underlying mapping
$F(x)$	$H(x) - x$
$F(x) + x$	Original mapping

Table 2.2: Notations concerning building blocks of ResNet[31]

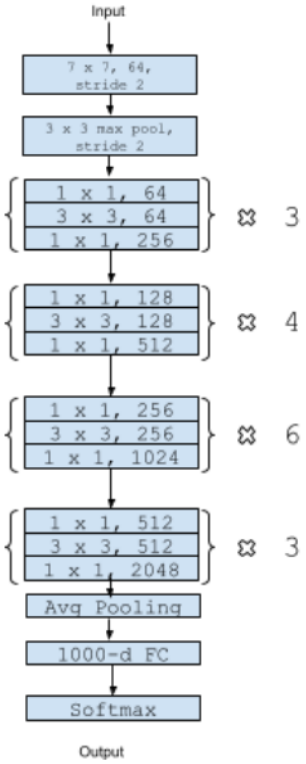

Figure 2.15: ResNet50 block-diagram [31]

In ResNet50, 3-layer 'bottleneck' blocks are used. [31].

2.2.4.3 EfficientNet

By carefully balancing scaling in all three dimensions, width, depth and resolution, mobile-size EfficientNet models can be built. This is an effective compound scaling method to easily scale up baseline models to obey any target resource constraints in a principled way without losing model efficiency [32].

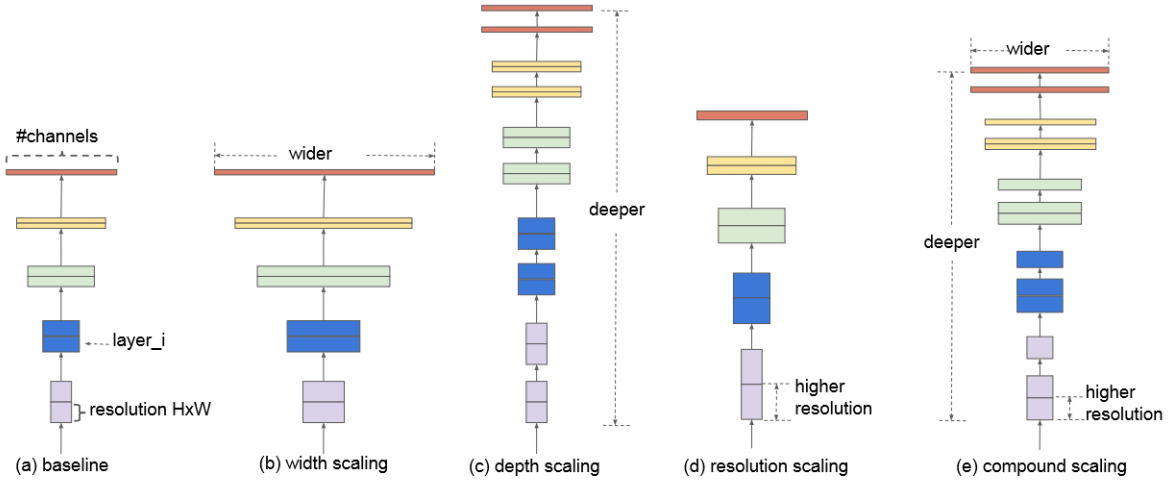

Figure 2.16: Compound scaling concept in EfficientNet architectures : (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) compound scaling method that uniformly scales all three dimensions with a fixed ratio[32]

2.2.4.4 Faster R-CNN

Object detection is more challenging than classification and requires more complex methods. Fast R-CNN is a single-stage training algorithm capable

of learning the classification of object proposals and refinement of their spatial locations. Fast R-CNN uses selective search to find regions of interest and achieves near real-time detection rates. [33] [34].

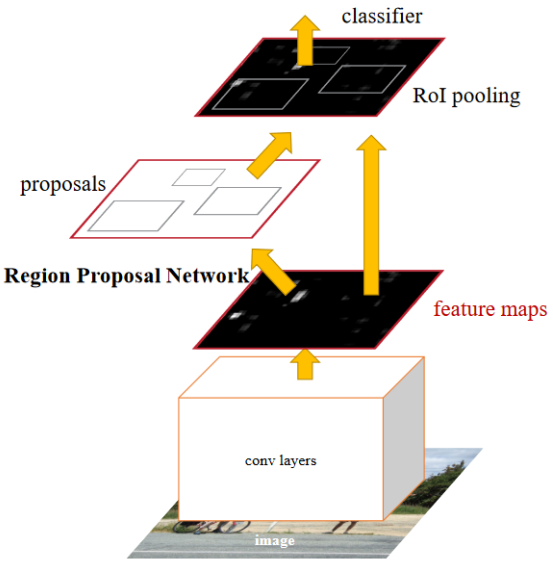

Figure 2.17: Faster R-CNN Architecture[34]

In contrast to selective search used in Fast R-CNN, a Region Proposal Network (RPN), which is a deep convolutional network, is used in Faster R-CNN. The training scheme followed will alternate between fine-tuning for the region proposal task and the object detection task, enabling quicker convergence with shared convolutional features for both tasks. The RPN can be thought of as the "attention" of Faster R-CNN architecture. [34].

2.2.4.5 EfficientDet

Many object detection architectures cannot be used in real-world resource constraint environments due to the model's large size and expensive computation cost. EfficientDet aims to address this issue by providing a scalable object detection architecture without sacrificing accuracy. EfficientDet largely follows one-stage detection techniques like SSD, YOLO, feature pyramid networks and focal loss based approach [35] [36] [37] [38][39].

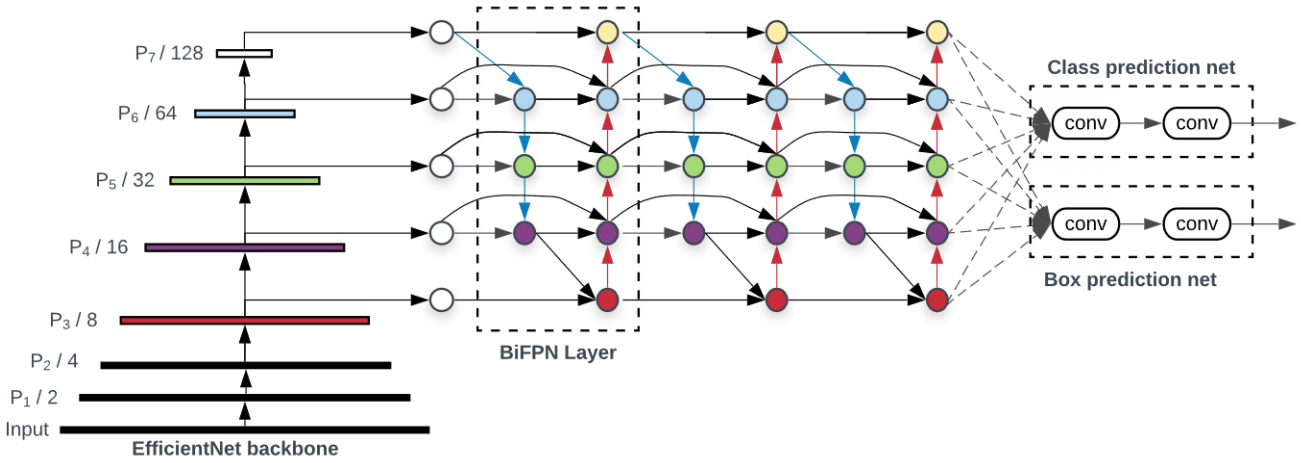

Figure 2.18: EfficientDet Architecture[39]

The EfficientDet architecture employs ImageNet-pre-trained EfficientNets as the backbone network. A BiFPN acts as a feature network. This BiFPN takes 3-7 features from the EfficientNet backbone and repeatedly applies bidirectional feature fusion. Fused features are then fed to a class and box network to produce class and bounding box predictions [39].

Chapter 3

Methodology and Tools Used

3.1 Data

Data is an essential ingredient in building machine learning models. Understanding the data is a crucial prerequisite for any machine learning application.

Data that was available for this project was image data. A coconut image dataset containing a total of 999 images of tender and matured is considered the primary source for data in this project [40].The image data were in two folders - namely, tender and matured. The matured folder contained six subfolders representing coconut varieties.

Figure 3.1: Sample images from the initial dataset provided

3.1.1 Data Preparation for Classification

Inside the folder mature, there were six sub-folders named one, two, three, four, five, and six, where the numbers indicated different varieties of coconuts. Since the project only aims to classify coconuts based on their maturity level, a variety wise categorization was unnecessary. The images were rearranged to just two folders (tender and mature) without any sub-folders. Further, the images were split into train, validation and test images as appropriate before using them.

3.1.2 Data Preparation for Object Detection

A dataset with labelled bounding boxes must be prepared to train the object detection model. Then these images with bounding boxes have to be converted to a suitable format for object detection models like the PASCAL VOC format [41], or the Tensorflow Object Detection CSV Annotation For-

mat [42].

The *labelImg* python package was used to annotate images. Once the annotation is finished, a set of new *.xml files, one for each annotated image, were generated. These .xml files were in PASCAL VOC format, and for any other preferred dataset format, the appropriate conversion was done.

3.1.3 Coconut Age Prediction

3.1.3.1 As a Classification Problem

The problem of age prediction of coconut can be considered as a classification problem. A classifier capable of predicting whether the input image belongs to class "tender" or class "mature" has been built. Transfer learning strategy was adopted, and pre-trained base models were used to build the classifier.

3.1.3.2 Building a Coconut Classifier Model

Google colaboratory (colab) was relied upon for building the model. Google Colab (Colaboratory), which provides easy access to ipython notebooks and GPU runtimes through a browser. Classifiers based on pre-trained VGG16, ResNet50 and EfficientNet-Lite0 base models were built.

VGG16 based model was created with the help of *keras* library and the ResNet50 and EfficientNet-Lite0 based ones with the help of *TFLite model*

maker. Integrating the Keras model into an Android app is not straightforward. TFLite model maker is a library that helps train a TensorFlow Lite model, which can be easily integrated with an Android app.

Some images from the coconut image dataset were kept in reserve for inference.The remaining coconut image data was split into train, validation, and test sets in the ratios 80 percent, 10 percent, and 10 percent, respectively.

Two custom image classifier models were built using TFLite model maker and the loaded coconut image data. One model was based on the default EfficientNet-Lite0 model available in TFlite, and the second was based on ResNet50. The ResNet50 model specification was provided to image classifier module's ResNet50Spec method. Then the model can be trained using create method. Four epochs were given for training the ResNet50 model. Then the trained model was evaluated based on accuracy using the evaluate method. The results were displayed on the notebook in a manner where correct predictions were shown in black color, and wrong predictions were shown in red color.

3.1.3.3 Building an Android App Using the Model

The saved tflite model based on ResNet50 was downloaded and incorporated into an Android application with the help of Android studio. ResNet50 based model was chosen to build the app because it produced more accuracy than the EfficientNet-Lite0 based one. Testing was done with coconut

images appearing on a computer screen to check the basic working of the sample app. Images were from the two classes of tender and mature. Inference was performed using the app as well as on the colab notebook by directly providing images.

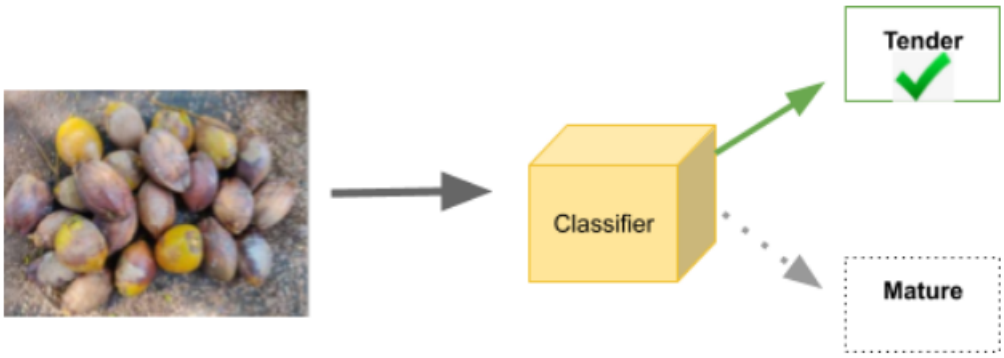

Figure 3.2: Coconut age prediction as a classification problem

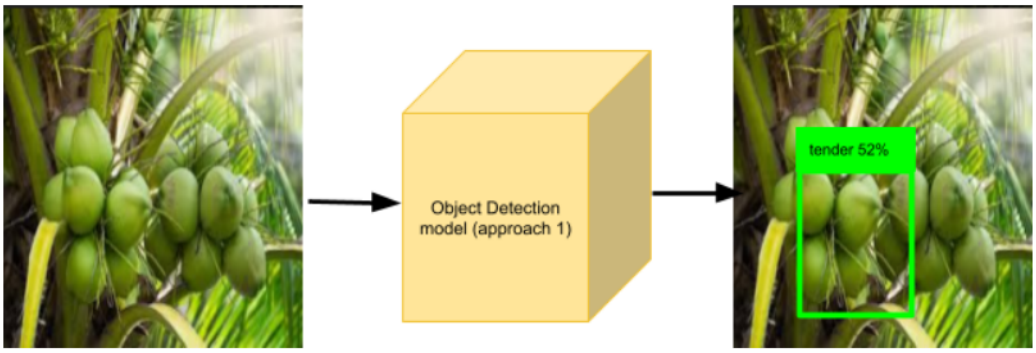

Figure 3.3: Coconut age prediction with object detection - first approach

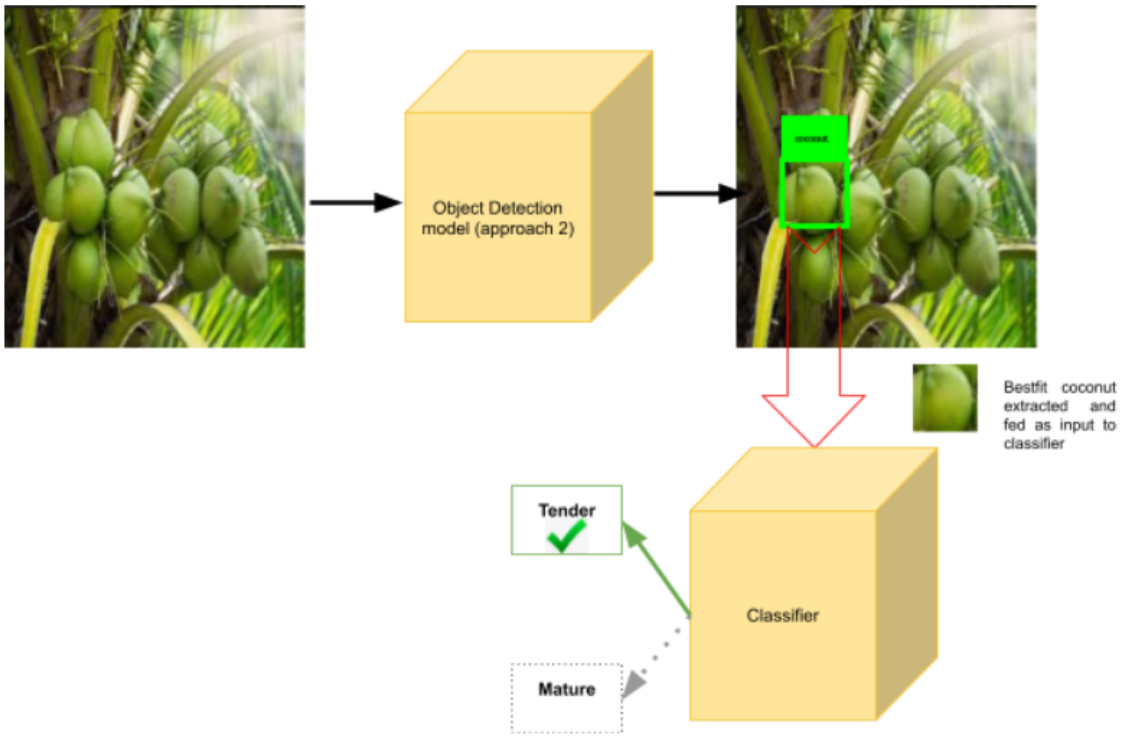

Figure 3.4: Coconut age prediction with object detection - extracting best fit coconut and feeding to classifier - approach 2

3.1.3.4 As an Object Detection Problem

In more natural settings, a simple classifier will not be sufficient. Instead, a machine learning model capable of detecting coconuts from the complex background of coconut leaves and leaf stalks is needed. This requires object detection followed by classification. A Faster-RCNN model with ResNet50 based RPN has been shown to be very suitable for detecting and classifying coconuts. [4].

An attempt was made to build an object detection model capable of predict-

ing coconut age based on pre-trained Faster R-CNN mode from Tensorflow model zoo. To create an Android application, *.tflite* format model is suitable. So, with the help of *TFLite* model maker, a model was created which could detect coconut and show a bounding box around coconut along with the class label and confidence level (tender or mature). This model was based on the Efficientdet lite0 pre-trained model, an EfficientDet object detection architecture. This *tflite* model can be used to build an Android app that can detect and classify coconuts.

3.1.3.5 Different Approaches

As an object detection problem, coconut age prediction can be performed in different methods. Earlier works attempt to directly predict the class label (tender/mature) along with confidence with a bounding box on the input image.

However, a different approach can be tried, which first detects a best-fit coconut from the input image, demarcates it with a bounding box, extracts this region identified as coconut and then gives it as input to a classifier to predict the age as tender or mature.

The second method can help reduce the complexity of the object detection model since it now only needs to identify coconut. This can also improve accuracy since the classifier built in the previous section was performing better in terms of accuracy than the object detection model.

Chapter 4

Results and Discussions

Different machine learning models were created for age prediction of coconut. The problem is first approached as just a classification problem. Then object detection is also considered. These models created were able to predict the age of coconuts with varied success rates. Classifier model performed with better accuracy than object detection model. The object detection model can be improved by expanding the image dataset.

4.1 As a Classification Problem

Figure 4.1: Results from inference performed using EfficientNet-Lite0 based classifier

Model	Accuracy
EfficientNet-Lite0	90 percentage
ResNet50	93 percentage

Table 4.1: Accuracy of Models for different base models - Coconut Age Prediction as a Classification Problem

Figure 4.2: Results from inference performed using ResNet50 based classifier

4.2 As an Object Detection Problem

Figure 4.3: Result from inference performed using EfficientDet-Lite0 based object detector

Model	AP at IoU=.50
EfficientDet-Lite0	0.007538254
EfficientDet-Lite4	0.0109431995

Table 4.2: Accuracy of Models for different base models - Coconut Age Prediction as a Object Detction Problem

4.3 Epoch vs Loss curves

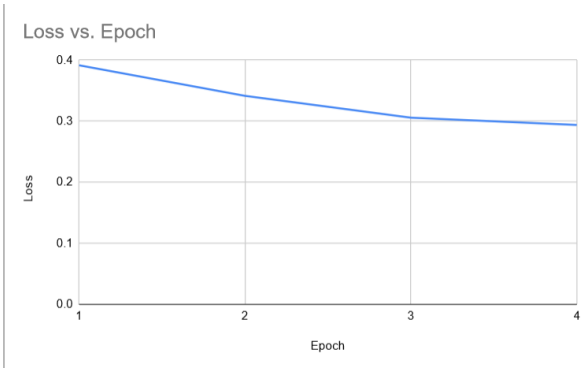

Figure 4.4: Epoch vs Loss for classifier model based on EfficientNet-lite0

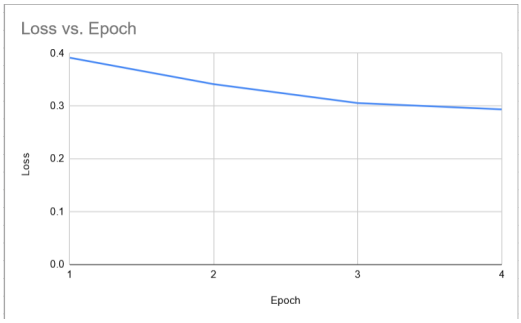

Figure 4.5: Epoch vs Loss for classifier model based on ResNet50

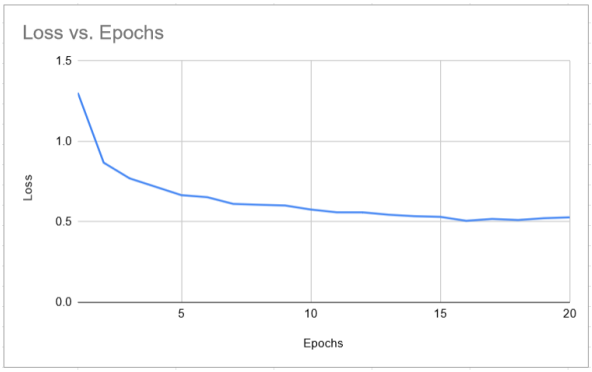

Figure 4.6: Epoch vs Loss for object detection model based on EfficientDet-lite0

For the classifier, ResNet50 based model provided more accuracy and the inference results performed on colab notebook agree with the 93 percentage accuracy obtained from *evaluate tflite()* method.

The object detection based approach produced models that did not perform well compared to the classifier models. In instances where the object detection model was able to detect tender or mature coconut from input images of a bunch of coconuts, the confidence of the label assigned stayed below 60 percent.

Result videos demonstrating the working of sample android apps built with these trained models can be found here : https://bit.ly/mlCoconut

Chapter 5

Future Work

Results can be improved by expanding coconut image dataset to include known bunches having coconuts belonging to tender and mature classes. Best-fit coconut based annotations can be created and a dataset thus created can be used to first identify best-fit coconut, then to extract the region of interest and then to feed this extracted image to a classifier with good accuracy. This way, the complexity of object detection model can be reduced and accuracy of end result can be improved.

References

[1] Abhinav Sharma et al. "Machine Learning Applications for Precision Agriculture: A Comprehensive Review". In: *IEEE Access* 9 (2021), pp. 4843–4873. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3048415.

[2] Andreas Kamilaris and Francesc Xavier Prenafeta-Boldú. "Deep learning in agriculture: A survey". In: *Comput. Electron. Agric.* 147 (2018), pp. 70–90.

[3] Thamban K.P. and D. Jaganathan. "Coconut production in Kerala in Indian Coconut Journal". In: (Aug. 2016).

[4] Subramanian Parvathi and Sankar Tamil Selvi. "Detection of maturity stages of coconuts in complex background using Faster R-CNN model". In: *Biosystems Engineering* 202 (2021), pp. 119–132. ISSN: 1537-5110. DOI: https://doi.org/10.1016/j.biosystemseng.2020.12.002. URL: https://www.sciencedirect.com/science/article/pii/S1537511020303329.

[5] Asok Mithra et al. "Coconut- Value Added Products". In: *Indian Food Industry* (Dec. 2013).

[6] Parvathi Subramanian and Tamil Selvi Sankar. "Coconut Maturity Recognition Using Convolutional Neural Network". In: *Computer Vision and Machine Learning in Agriculture, Volume 2*. Ed. by Mohammad Shorif Uddin and Jagdish Chand Bansal. Singapore: Springer Singapore, 2022, pp. 107–120. ISBN: 978-981-16-9991-7. DOI: 10.1007/978-981-16-9991-7_7. URL: https://doi.org/10.1007/978-981-16-9991-7_7.

[7] Chiagoziem C Ukwuoma et al. "Recent advancements in fruit detection and classification using deep learning techniques". en. In: *Math. Probl. Eng.* 2022 (Jan. 2022), pp. 1–29.

[8] Wei Chen et al. "An Apple Detection Method Based on Des-YOLO v4 Algorithm for Harvesting Robots in Complex Environment". In: *Mathematical Problems in Engineering* 2021 (2021), p. 7351470. ISSN: 1024-123X. DOI: 10.1155/2021/7351470. URL: https://doi.org/10.1155/2021/7351470.

[9] Mohammed Faisal et al. "Deep learning and computer vision for estimating date fruits type, maturity level, and weight". In: *IEEE Access* 8 (2020), pp. 206770–206782.

[10] S K Niranjan Siddesha S. "Color and Texture in Classification of Coconut". In: *(IJITEE* 8 (8 2019).

[11] June Anne Caladcad et al. "Determining Philippine coconut maturity level using machine learning algorithms based on acoustic signal". In: *Computers and Electronics in Agriculture* 172 (2020), p. 105327. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2020.105327. URL: https://www.sciencedirect.com/science/article/pii/S0168169919324767.

[12] Irister M. Javel et al. “Coconut Fruit Maturity Classification using Fuzzy Logic”. In: *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM)*. 2018, pp. 1–6. DOI: 10.1109/HNICEM.2018.8666231.

[13] Vedant Sandeep Joshi, Jeena Thomas, and Ebin Deni Raj. “Quantized Coconut Detection Models with Edge Devices”. In: *Journal of Interconnection Networks* 0.0 (0), p. 2144010. DOI: 10.1142/S0219265921440102. eprint: https://doi.org/10.1142/S0219265921440102. URL: https://doi.org/10.1142/S0219265921440102.

[14] Mohamad Haniff Junos et al. “Automatic detection of oil palm fruits from UAV images using an improved YOLO model”. In: *The Visual Computer* (2021). ISSN: 1432-2315. DOI: 10.1007/s00371-021-02116-3. URL: https://doi.org/10.1007/s00371-021-02116-3.

[15] Huipeng Li et al. “A real-time table grape detection method based on improved YOLOv4-tiny network in complex background”. In: *Biosystems Engineering* 212 (2021), pp. 347–359. ISSN: 1537-5110. DOI: https://doi.org/10.1016/j.biosystemseng.2021.11.011. URL: https://www.sciencedirect.com/science/article/pii/S1537511021002786.

[16] Shrihari Kallapur et al. “Identification of aromatic coconuts using image processing and machine learning techniques”. In: *Global Transitions Proceedings* 2.2 (2021). International Conference on Computing System and its Applications (ICCSA- 2021), pp. 441–447. ISSN: 2666-285X. DOI: https://doi.org/10.1016/j.gltp.2021.08.037. URL: https://www.sciencedirect.com/science/article/pii/S2666285X21000650.

[17] Tito C. Lim et al. "De-Husked Coconut Quality Evaluation Using Image Processing and Machine Learning Techniques". In: *Proceedings of the 2019 6th International Conference on Bioinformatics Research and Applications*. ICBRA '19. Seoul, Republic of Korea: Association for Computing Machinery, 2019, 28–33. ISBN: 9781450372183. DOI: 10.1145/3383783.3383789. URL: https://doi.org/10.1145/3383783.3383789.

[18] Jason Brownlee. *Deep learning for computer vision: image classification, object detection, and face recognition in python*. Machine Learning Mastery, 2019.

[19] Laith Alzubaidi et al. "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions". In: *Journal of Big Data* 8.1 (2021), p. 53. ISSN: 2196-1115. DOI: 10.1186/s40537-021-00444-8. URL: https://doi.org/10.1186/s40537-021-00444-8.

[20] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. "Understanding of a convolutional neural network". In: *2017 international conference on engineering and technology (ICET)*. Ieee. 2017, pp. 1–6.

[21] Sohaib Asif et al. "A deep learning-based framework for detecting COVID-19 patients using chest X-rays". In: *Multimedia Systems* (2022). ISSN: 1432-1882. DOI: 10.1007/s00530-022-00917-7. URL: https://doi.org/10.1007/s00530-022-00917-7.

[22] Ziheng Wang and Ann Majewicz Fey. "Deep learning with convolutional neural network for objective skill evaluation in robot-assisted surgery". In: *International journal of computer assisted radiology and surgery* 13.12 (2018), pp. 1959–1970.

[23] Fuzhen Zhuang et al. “A comprehensive survey on transfer learning”. In: *Proceedings of the IEEE* 109.1 (2020), pp. 43–76.

[24] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. “A survey of transfer learning”. In: *Journal of Big data* 3.1 (2016), pp. 1–40.

[25] Pedro Marcelino. “Transfer learning from pre-trained models”. In: *Towards Data Science* 10 (2018), p. 23.

[26] Asifullah Khan et al. “A survey of the recent architectures of deep convolutional neural networks”. In: *Artificial intelligence review* 53.8 (2020), pp. 5455–5516.

[27] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[28] Dan Claudiu Ciresan et al. “Flexible, high performance convolutional neural networks for image classification”. In: *Twenty-second international joint conference on artificial intelligence*. 2011.

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012).

[30] Manolis Loukadakis, José Cano, and Michael O’Boyle. “Accelerating deep neural networks on low power heterogeneous architectures”. In: (2018).

[31] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[32] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

[33] Ross Girshick. "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[34] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015).

[35] Wei Liu et al. "Ssd: Single shot multibox detector". In: *European conference on computer vision*. Springer. 2016, pp. 21–37.

[36] Joseph Redmon and Ali Farhadi. "YOLO9000: better, faster, stronger". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7263–7271.

[37] Tsung-Yi Lin et al. "Feature pyramid networks for object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.

[38] Tsung-Yi Lin et al. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[39] Mingxing Tan, Ruoming Pang, and Quoc V Le. "Efficientdet: Scalable and efficient object detection". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 10781–10790.

[40] Dr Hemalatha Nambisan and Dr Rajesh M K. *"Coconut", Mendeley Data, V1*. DOI: 10.17632/mmc345tfjw.1.

[41] M. Everingham et al. "The Pascal Visual Object Classes Challenge: A Retrospective". In: *International Journal of Computer Vision* 111.1 (Jan. 2015), pp. 98–136.

[42] Tensorflow. *Training Custom Object Detector*. URL: https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/tensorflow-1.14/training.html.