GRADUATION PROJECT

Date Fruit Variety Classification

Jamal Al Mahrooqi 19786096
Abdallah Al Maamari 12056078

Department of Mathematical and Physical Sciences

University of Nizwa

January 29, 2023

Contents

List of Figures

List of Tables

Dates

ORIGINALITY REPORT

17%	14%	12%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to IIT Delhi Student Paper	3%
2	Juhi Kailash Ajwani, Ranjanbala Jain, Abhishek Badhe, Jyeshtha Prabhu, Atharva Narkhede. "ExFAURL Scanner: A Framework for Malware Detection in Binary files and URLs using Machine Learning Techniques", 2022 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), 2022 Publication	1%
3	akladyous.medium.com Internet Source	1%
4	wiki.seeedstudio.com Internet Source	1%
5	Submitted to Government College of Engineering Aurangabad, Maharashtra State, India Student Paper	<1%
6	archive.is Internet Source	<1%

Abstract

The classification of variety of objects has been and essential tool to facilitate information retrieval. This project includes the art of Image processing algorithm and Convolutional Neural Network (CNN) Deep learning algorithm, to classify the complications of the data, in our case the data is the fruit date production..

Image Processing is a method of extracting useful information from images by manipulating the pixels. This technique is used to enhance the quality of the image, reduce noise, and extract features that are useful for further analysis. The algorithm used in this project is designed to extract features such as color, shape, and texture from the images of fruit date production.

The combination of Image Processing and CNN deep learning algorithms allows for accurate and efficient classification of the data. The output of the algorithm is a set of labels that correspond to the different types of fruit dates. This information can then be used to facilitate information retrieval and improve decision-making in the field of fruit date production.

Overall, this project demonstrates the potential of using Image Processing and CNN deep learning algorithms for the classification of fruit date production. The results of this project can be used to improve the efficiency and accuracy of information retrieval in the field, and to make more informed decisions about fruit date production.

Chapter 1

Introduction

The world has always been keen to providing the best services in many different forms, one of them in agriculture, where dates are considered a delicacy in multiple countries specially Arab countries, according to the Islamic religion it has multiple benefits. To provide high-quality of dates, an automated classification technique may be helpful to provide vast amount of products. This is to reduce the cost and time in factory production for there are multiple types of dates, different sizes, colors, shapes and tastes.

1.1 Project aim

Our project aim is to create labelled data-set that would include images of four different types of popular date varieties - Barni, Khenaizi, Segai, and Khalas - and classify them by training a Convolutional Neural Network (CNN) with an acceptable accuracy.

1.2 Background of the study.

Image processing is a field of computer science that deals with the manipulation and analysis of images. It involves the use of algorithms and techniques to extract useful information from images, as well as to enhance or modify images for a variety of purposes. There are many algorithm and techniques to achieve image classification including image smoothing, image sharpening, shadow removal, image color geometry and shape. there are concept included as well like categorizing and labeling groups of pixels or vectors within an image based on specific rules. Image classification in general has many benefits in multiple firms.

Phoenix dactylifera, also know as dates, produced from the date palm tree

1.3 Statement of the problem

To Solve the problem of miss-classifying different types of dates this can not only help introduce different variety of dates, but it may help production factories in working with several types of dates and analyze the images with result that can surpass human level accuracy.

1.4 Objective of the study

The general goal of this project is to give a machine as much data as possible, to the point where it could eventually predict like us humans, where it can distinguish between multiple objects with different sizes and shapes. Dates is just one of the objects out of 45.719843824 billion objects. The main goal is distinguish between different types of dates. the objective of this research is to identify and discover multiple features an image has. Image classification is perhaps the most important part of digital image analysis.

1.4.1 Research Questions

To what extinct can the model predict?

-The model can predict an accuracy of around 90.

How can the model be improved?

-The model can be improved drastically by increasing the quantity of the data-set and discover more features to add into the model. What are the problems occurred during the research? -The most common problem we have faced is that the date fruit changes with time, this changes can be in its colour and shape which can make it seem like the other type of date which eventually gives false result and effect accuracy.

Chapter 2

Background Concepts

Computer vision is a rapidly high growing technology which if operated correctly can not only mimic human level vision but mimic it more efficiently and accurately with short amount of time, this can be taken advantage in production factories. Here, vast amount of products are produced, separated and packaged in short amount of time, in our case Distinguishing between dates can gradually help in separating different types of dates in production factories. Tho there are many features that can classify dates besides vision, some of these features cannot be detected visually, but can be detected by its sugar content.
Some applications of computer vision include:

1. Object recognition: Computer vision algorithms can be used to identify and classify objects in images or video. This can be useful for tasks such as automatically tagging images with keywords or for sorting and organizing large datasets.

2. Image and video analysis: Computer vision algorithms can be used to analyze images and videos in order to extract information such as the location of objects, the movement of people or vehicles, and the overall scene structure.

3. Augmented reality: Computer vision can be used to overlay digital information onto the real world, enabling applications such as virtual try-on for clothing, or real-time translation of signs and labels.

4. Autonomous systems: Computer vision can be used to enable autonomous systems such as self-driving cars, drones, or robots to navigate and interact with their environment.

2.0.1 Dates fruits

High-quality dates are produced in large quantities in the Arab Gulf region, where numerous distinct varieties of date fruit are grown. Dates fruits are soft and sweet and is known for its rich flavor. They come in different variety of color and shape, following table shows some type of dates and their shape and color, it is important to know that the shape and the color can vary depending on the condition of it growth, maturity, the way they are stored and many more factor.

Type of Date Fruit	**Shape**	**Color**
Medjool	Large and plump, slightly oblong	Dark brown with slightly golden tinge
Deglet Noor	Small to medium, pointed tip and slightly curved	Golden with translucent appearance
Barhi	Small and round, wrinkled texture	Golden yellow
Halawy	Medium, slightly curved	Shiny dark brown
Zahidi	Medium, slightly pointed tip	Light brown
Thoory	Small to medium, slightly pointed tip	Light to medium brown
Khadrawy	Small to medium, slightly pointed tip	Dark brown
Dayri	Medium, slightly pointed tip	Light brown
Ajwah	Large and round, slightly pointed tip	Dark brown
Hayani	Small to medium, slightly pointed tip	Light to medium brown
Maktoom	Large and plump, slightly oblong	Shiny dark brown
Fardh	Medium, slightly pointed tip	Light to medium brown
Barhee	Small and round, wrinkled texture	Golden yellow
Black Sphinx	Small to medium, slightly pointed tip	Shiny dark brown
Tariq	Small to medium, slightly pointed tip	Light to medium brown
Khalas	Large and plump, slightly oblong	Dark brown
Khenaizi	Medium, slightly pointed tip	Light to medium brown
Barni	Small and round, wrinkled texture	Golden yellow
Segai	Medium, slightly curved	Dark brown

Table 2.1: Types of Dates

There are five phases of date maturity shown in 2.1, we observed that at the Hababauk stage that the fruit is extremely unripe and dark-green in color, in Kimir stage the date fruit is still unripe greener than previous stage, low in sugar content and larger in shape. Khalal considered fully mature stage and it has reached is maximum size but still unripe, the color vary from yellow to green. At Rutab stage the dates are considered fully ripe and texture has soften and consumable, now they are at level were the sugar content is high and the color has changed to light-brown, and some of the dates at this stage have the bottom halve is darker than the top. The Final Stage is Tamer where the date are left in a controlled environment to ripen, at this level the sugar content will increase but becomes less moist and the color changes to brown and could go darker depending on the type and storing environment of the date

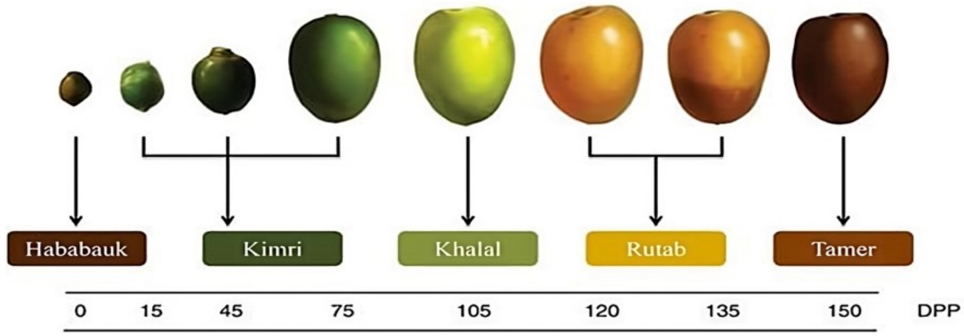

Figure 2.1: Date Maturity Stages

2.1 Image representation and processing

In order to understand how image classification algorithms work, we shall first explain how digital images are represented on the computer. There are Two way of representing image but we are concerned with bitmap representation as it is the one used in digital image for capturing high detail.

2.1.1 Bitmap Images

Bitmap formats refer to how bits in a pixel map to a specific color in the image, it is the most common format used to represent digital photograph that require high detail. A bitmap images take the form of matrix or 2D array,

each element is called picture element (pixel) and each horizontal line is called scan line and the number of pixel in this scan line represent the width of the image, while height of the image represent by the number of scan line. Image resolution determined by the number of pixel and the higher the number of pixels a more detail it contain. Generally speaking, there are two type we can represent bitmap images. Grayscale representation where each pixel is single value, the value represents the brightness of pixel, black pixel with a value of zero and white pixel having a value of 255. Second type is colored image where the pixel is represented by 3 channels of colors, namely Red, Blue, and Green. The final value of the pixel is the result of combination of these three channels. Bitmap images are well suited in application

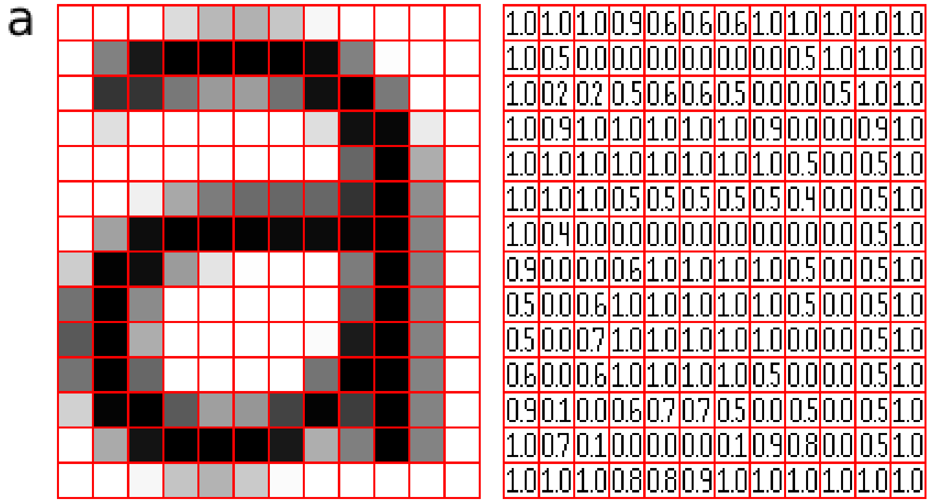

Figure 2.2: Raster Image

where high level of detail is needed such as digital photograph, graphic design and digital art. Bitmap images comes with unavoidable downside such as large file size, loss of quality when scaled.

Figure 2.3: Three-Channel Colors

Figure 2.4: Grey Scale

2.2 Image Classification

2.2.1 What is image classification?

Image classification refer to the task of categorizing raster image and giving them labels based on pre-trained model. It is an efficient technique to analyze unstructured image data. Almost all the images that been capture from the camera are unstructured in a way to obtain a specific information from it, for instance finding a cat in the image. Hence the need for AI and image classification to analyze the image and obtain the desired data from it. Classification technique has been integrated into the industries. Such applications are security application, object recognition in driverless cars, and automated inspection and quality control in assembly line. The techniques that are used in image classification are mainly divided

Figure 2.5: Self-driving AI

to two techniques, supervised and unsupervised.

2.2.2 How does Image classification work?

Machine learning algorithm analyze the raster image in form of matrix of pixels to generate a statistical model (model), the size of this matrix relies on image resolution. The algorithm process a given pixels matrix and grouping them into specified classes. These algorithms depend lot on the quality of the data, hence a will-optimize data will result in efficient model and a poorly optimized dataset will result in a less efficient model.Using the proper algorithm for the job and having an excellent dataset to train the model are crucial. It's critical to remember that the model's performance also depends on the features selected for extraction (if CNN is not used), the preprocessing methods, and the algorithmic settings.

After extracting the features either manually or learned through a convolutional neural network (CNN), there are used as input to a classifier. Classifier is trained to predict the class for a given image based on the features. A basic method like logistic regression or a more sophisticated algorithm like a neural network can be used as the classifier. Once the model is trained, we can use it to classify new images, and the process of doing so is by first getting the image features then feeding them into the classifier to make a prediction the shape of the output is discussed later in following chapters.

2.3 Deep Learning Convolutional Neural Network (CNN)

Artificial neural network

Artificial neural network (ANN) is a computational model modeled after the structure and operation of the human brain. It is made up of several linked ”neurons,” which are basic processing units that take in information, process them mathematically, and output results. Each neuron’s input and output are passed through a set of weights, which stand for the degree of connectivity between the neurons.

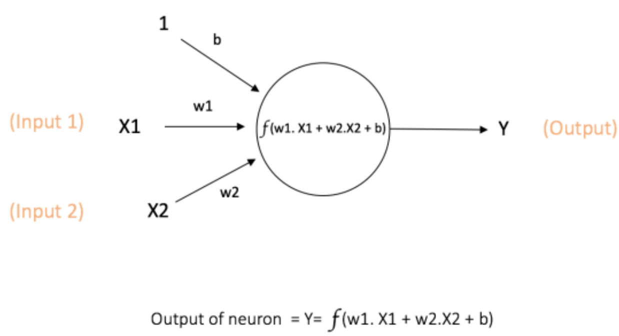

Figure 2.6: Artificial Neuron

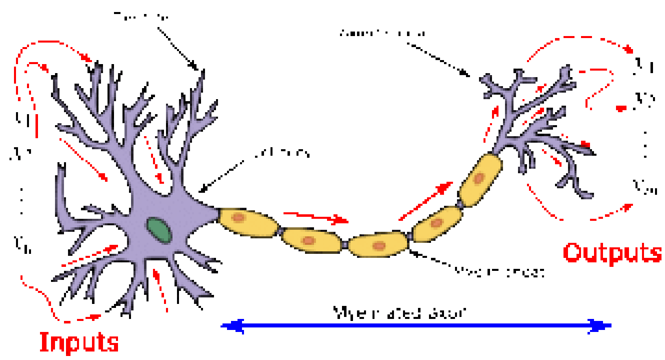

Figure 2.7: Human Neuron

The input signals to an ANN are passed via a number of layers of neurons, each of which computes the input and delivers the result to the layer below. The network as a whole is produced by the last layer. This process of passing input through multiple layers of neurons is called **feedforward** The weights of the neurons within an Artificial Neural Network (ANN) can be adjusted through a process known as **backpropagation**. This algorithm optimizes the weights based on the input-output relationship, thus enabling the network to learn from examples and improve its performance over time. ANNs are considered a powerful tool for machine learning and have been applied to a wide range of tasks, including image and speech recognition, natural language processing, and control systems.

Deep Neural Network (DNN)

The Neural network is called deep based on the number of hidden layers and number of unit in single layer. The popular type of Deep Neural Network DNN are:

1. Convolutional Neural Networks (CNNs): Commonly used for image classification or analyzing video, these types of Neural Network are able to identify features or pattern in the image by using convolutional layers.

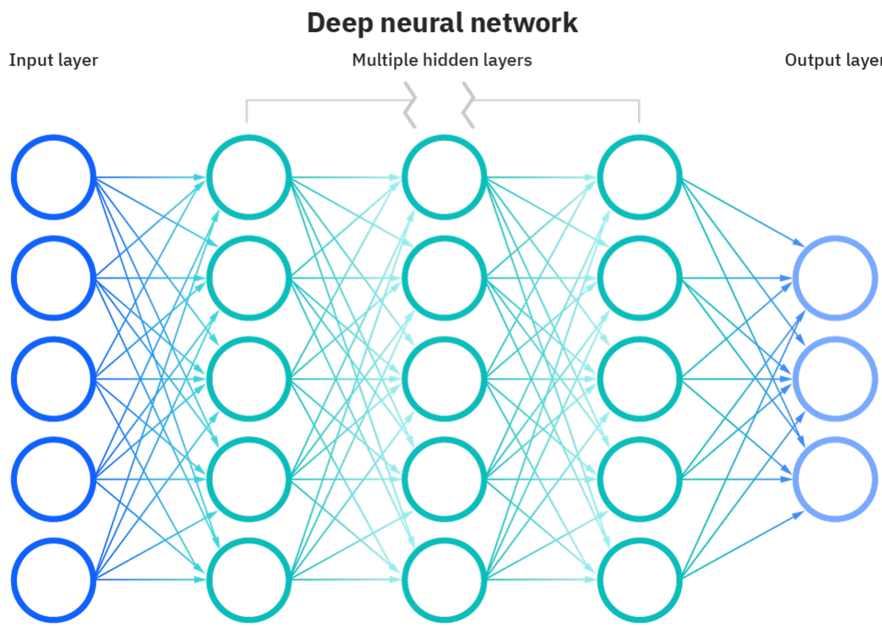

Figure 2.8: Deep Neural Network

2. Recurrent Neural Networks (RNNs): For sequential data, such as audio, text, and time series, these are employed. The network can store information about past inputs thanks to the recurrent layers, making it appropriate for sequential data.

3. Generative Adversarial Networks (GANs): Consisting of a generator and a discriminator, two neural networks that have been trained to work together to produce new data that has some similarity with training data.

4. Transformers: Largely used for Natural Language Processing tasks such as texts summarization and language translation. What make Transformer Model special is self-attention mechanism that allow it to weigh the importance of different words when making prediction.

These DNNs have been used to perform a variety of tasks including speech and picture recognition, natural language processing, and decision making. They are the most popular and well-established kinds of DNNs.

2.3.1 CNN

A convolutional Neural Network (CNN) is a framework built using the deep neural network concept. It is widely used in computer vision, but it can also be used in speech recognition and Natural Language

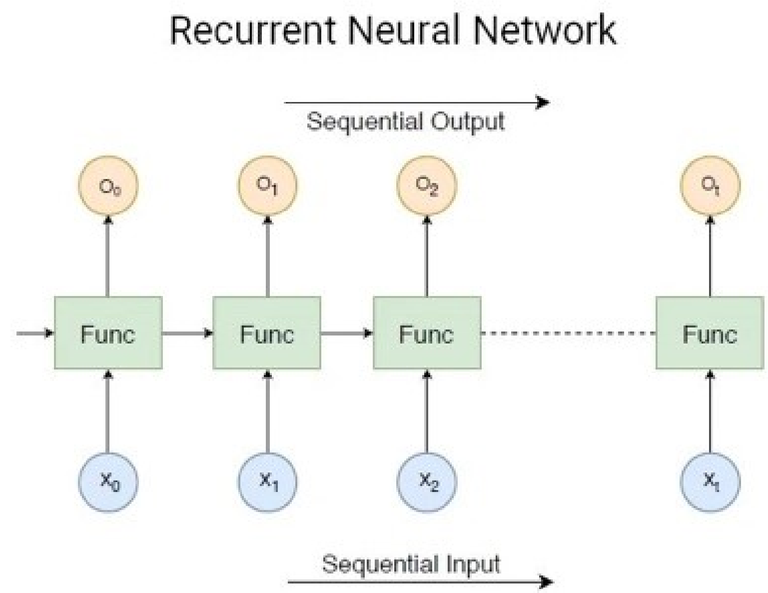

Figure 2.9: Recurrent Neural Network

Processing. image classification can utilize the ability of CNN to automatically extract features from the data to reduce the pre-processing of the image data and increase the efficiency of the generated model. CNN architecture includes four main type layers.

1. The convolution Layer is a filter to an input that results in activation, this layer is the main building block in CNN.

2. ReLu Layers or Rectified linear unit layers are activation functions used to reduce overfitting and build the effectiveness and accuracy of convolutional neural networks.

3. The pooling Layer task is to lower the number of factors being considered by gathering all the results from previous neurons and processing this data.

4. The fully-Connected Layer flattens the input received from the layer preceding it and gives the final result.

2.3.2 Flutter

Flutter is a free and open-source mobile application development framework created by Google. It is used to develop applications for Android, iOS, Linux, Mac, Windows, and the web. Flutter uses the Dart programming language, which was also created by Google.

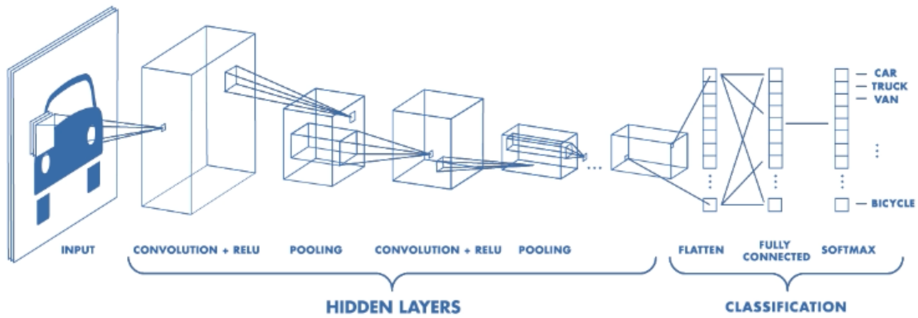

Figure 2.10: CNN layers

Figure 2.11: Flutter

One of the key features of Flutter is its use of reactive programming. This means that the framework allows for the creation of responsive and dynamic user interfaces, where changes in the underlying data are immediately reflected in the user interface. This results in a smooth and seamless user experience.

Performance

Flutter is considered to be highly performant when compared to other mobile development frameworks one of the reaseons behind this is its use of a reactive programming model. This model allows the app to respond immediately to user interactions and updates to the app's state, without the need for complex and costly layout calculations.

Flutter's performance is also enhanced by its use of the Dart programming language. Dart is a fast and efficient language that is optimized for mobile development and provides a smooth and responsive experience for the user.

Flutter's hot-reload feature allows developers to experiment, build UIs, add features, and fix bugs faster. Comparing Flutter to other mobile development frameworks such as React Native and Xamarin, Flutter has a more efficient layout and rendering engine, which results in better performance. React Native and

Xamarin use JavaScript and C respectively, which are not as fast as Dart.

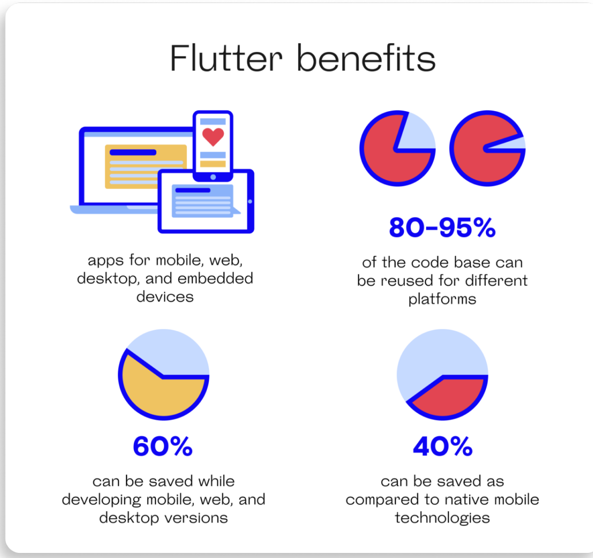

Figure 2.12: Benefits

2.3.3 Architectural layers

Flutter's architecture is based on a layered approach, which separates the app's functionality into different layers. These layers are:

1. The Application Layer: This is the highest layer of the app and it is responsible for managing the app's state, handling user input, and calling the other layers to perform their respective tasks. The application layer is typically implemented as a set of widgets and it is where the app's business logic is defined.

2. The Framework Layer: This layer provides the app with a set of pre-built classes and functions that can be used to perform common tasks such as managing the app's state, handling user input, and making network requests. The framework layer also provides the app with a set of pre-built widgets that can be used to create the app's layout.

3. The Engine Layer: This is the lowest layer of the app and it is responsible for rendering the app's user interface, managing the app's lifecycle, and providing the app with access to the device's resources. The engine layer is implemented in C++ and it communicates with the framework layer using Dart FFI.

4. The Foundation Libraries: This is collection of libraries that provide basic classes and functions for common tasks such as strings, collections, and IO.

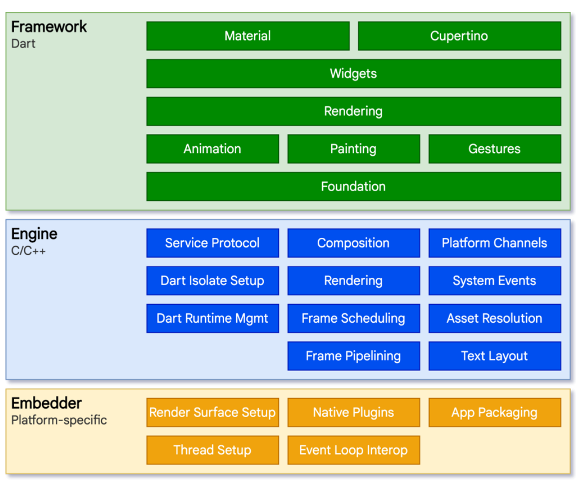

Figure 2.13: Flutter Architectural layers

In summary, the architecture of Flutter is based on a layered approach that separates the app's functionality into different layers, which allows for a clear separation of concerns and makes it easy to understand

and maintain the code. The layers communicate with each other through well-defined interfaces, making it easy to test and debug the app. Additionally, the architecture is designed to be highly performant and responsive, making it a great choice for developing high-performance mobile applications.

Chapter 3

Literature Survey

With the increase in date fruit production around the world, many complications can occur in the process of harvesting and separation, here computer vision can drastically help. Many researchers have explored this area of work and suggested many ways of solving the problem efficiently. This chapter focuses on the research that helps solve distinguishing between different types of dates and their forms.

3.0.1 Date Maturity

Date fruits have many forms to distinguish them from each other and the major issue to consider is their maturity. The date matures in different colors, so the model may confuse its physical look with other types of dates. Date maturity changes the look of the date which can affect the model's result, so based on this issue we must find a different way to classify a date. A method proposed in [13], recommends a grading system that includes a weighing system, firmness tester, aroma sensor, image acquisition, image prepossessing, image segmentation, feature extraction, and Knowledge-based comparison and decision-making.

3.0.2 Surface defects

One of the main reasons why the date fruit is difficult to distinguish from each other is their defective surfaces such as bruises, scars, or insect damage which can be used as a feature for classification. There are many ways this was solved using many ways including [1], here the fruit goes through a series of stages which are shown in the figure below.

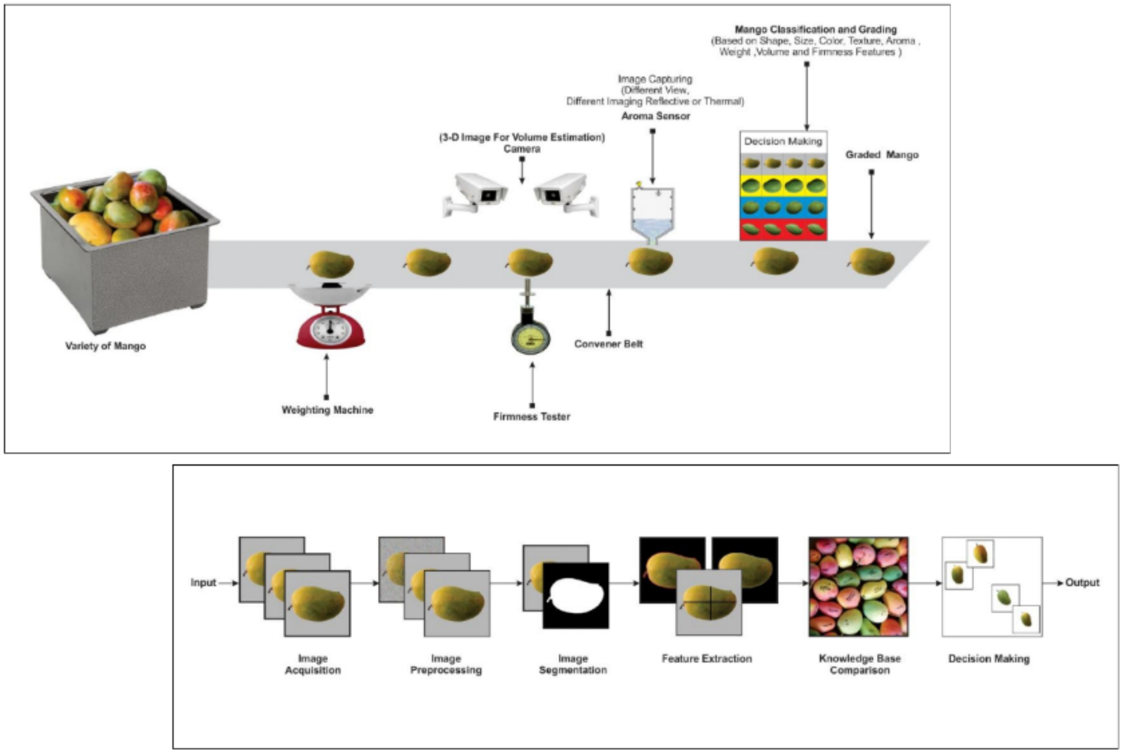

Figure 3.1: Proposed grading system

3.0.3 Tissue Pattern

To fully distinguish one object from another a sequence of different tissue patterns must be found. The local binary pattern (LBP) and Weber local descriptor (WLD) methods are used to extract the details of a date fruit's tissue pattern.

Weber local descriptor (WLD)

Weber local descriptor (WLD) is applied for addressing the challenges in five major race groups: Asian, African or American Black, Hispanic, Middle-Eastern, and White domains, [12]. where it showed an acceptable accuracy of Asian: 97.74%, Black: 96.89%, Hispanic: 92.06%, Middle: 98.33%, and White: 99.53%.

Local Binary Patern(LBP)

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. Due to its discriminative power and computational simplicity, the LBP texture operator has become a

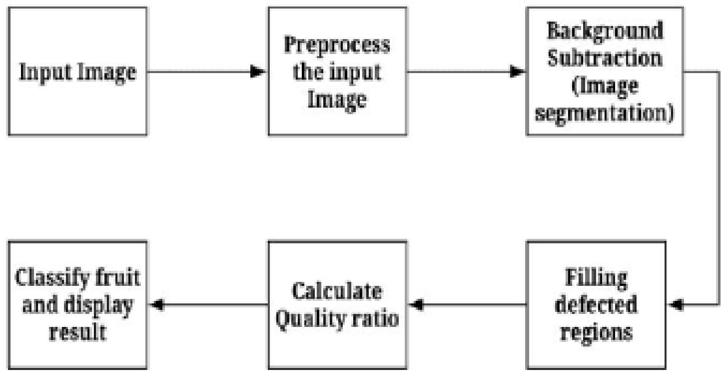

Figure 3.2: Image surface defects

popular approach in various applications. It can be seen as a unifying approach to the traditionally divergent statistical and structural models of texture analysis. Perhaps the most important property of the LBP operator in real-world applications is its robustness to monotonic gray-scale changes caused, for example, by illumination variations. Another important property is its computational simplicity, which makes it possible to analyze images in challenging real-time settings.

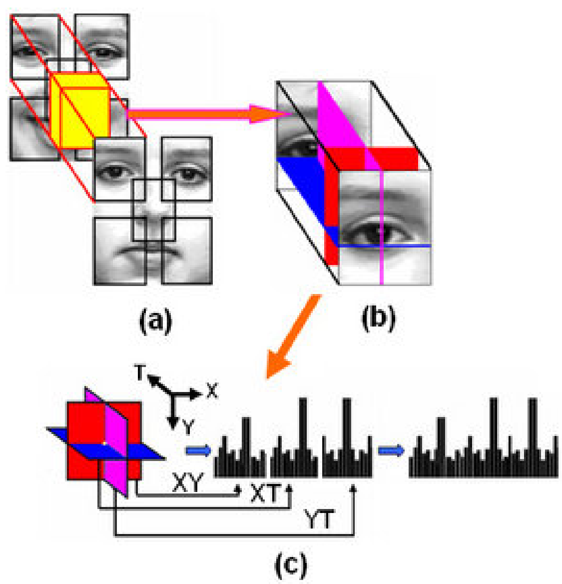

Figure 3.3: LBP on facial expression

The basic idea for developing the LBP operator was that two-dimensional surface textures can be described by two complementary measures: local spatial patterns and gray scale contrast. The original LBP operator(Ojala et al. 1996) forms labels for the image pixels by thresholding the 3 x 3 neighborhood of each pixel with the center value and considering the result as a binary number. The histogram of these 28 = 256 different labels can then be used as a texture descriptor. This operator used jointly with a simple local contrast measure pro- vided very good performance in unsupervised texture segmentation (Ojala and Pietik ainen 1999). After this, many related approaches have been developed for texture and color texture segmentation. The following notation is used for the LBP operator: LBPP, Ru2. The subscript represents using the operator in a (P, R) neighborhood. Superscript u2 stands for using only uniform patterns and labeling all remaining patterns with a single label. After the LBP labeled image fl(x,y) has been obtained, the LBP histogram can be defined as [Hi=x,yIfl(x,y)=i,i=0,. . . ,n1,]

The value of the LBP code of a pixel (x_c, y_c) is given by:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^P \qquad s(x) = \begin{cases} 1, if\ x \geq 0; \\ 0, otherwise. \end{cases}$$

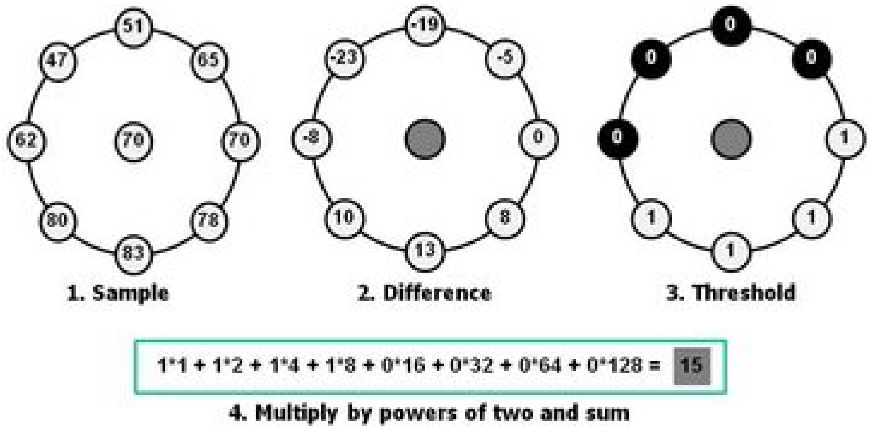

Figure 3.4: LBP computation

Chapter 4

Methodology

This project has started by a set of learning processes, which involves in learning all about TensorFlow framework and Flutter kit and a lot of research to include in this project. We started at gathering pictures of 8 different types of dates with different environment, then we had them labeled which can also be called tag, which is basically a data identification to tell which class does the data belong to which helps the ML model learn from this data and make the most accurate prediction. Secondly, Data augmentation [14] takes place. The general purpose of data augmentation is to increase data, there are many ways to increase data using data-augmentation, this includes flipping, rotations, Scaling, cropping, translations, zooming and many more. Next in preprocessing is splitting the data

The data that is fed to the model are in a form of pictures which then are put into a function to modify its size, resolution colors, illumination, texture or compositions and they are taken in single pictures. we spent a large amount of time collecting suitable dataset to obtain acceptable result which depends on the training accuracy so whenever the accuracy is low we would increase the data-set. As shown in these figures 4.3, the data are taken individually to reduce a known error in machine learning called noise, another reason why they are taken individually is due to some date's feature are easily detected by being individual which can effect the final result(accuracy).

4.1 Data

Data collection

To create our data-set, a set of eight different type of dates are being used to distinguish between four them, those five classes are Khanaizi, Barni, Segai, Khalas and mixed dates which contain the dates

	Barni	Khalas	Khenaizi	Segai	Mixed
Labels	0	1	2	3	4

Table 4.1: Data-set labels

Dates Data-set			
	Training	Testing	Validation
Barni	1664	418	208
Khalas	1276	161	159
Khenaizi	1137	197	197
Segai	1427	249	253
Mixed	844	107	105

Table 4.2: Data-set Splitting

Naghal, Barhi, Zahidi and khudri. To gather our dataset we took pictures with different cameras, with different ambiance and resolutions and luminous, more details are expressed below.

Data Pre-processing

Our model is able to accept different format of images such as jpeg and png which is taken by us and converted to its perspective dimension(224*224), then Data augmentation takes place which is a known technique used to increase the size of the dataset by using several methods of transformation.

The reason behind these data augmentation is to reduce over-fitting [21] which is a fundamental problem in machine learning which occurs when there are small amount of data-set, here data augmentation increases the aspect of the data-set and generalizing it better. data augmentation are done in a method used in [7] which is a preprocessing layer which randomly rotates images during training.

The data-set provided to the model are total of 5 classes, to create these classes, eight different types of dates were chosen, four of theses are used as primary classes and and the other four types of dates are used as a control class. Primary classes:[Barni, Segai, Khenaizi, Khalas] Control class:[mixed of Barhi, Zahidi, Naghal and khudri] The total amount of data-set is 7597 this does not include the data produced by data augmentation.

Naming system

In terms of naming the dataset, we took advantage of the keras library [8] which basically reads the data-set in alphabetical order and provide each class its own name in format of class 1 which corresponds to label 0 etc..

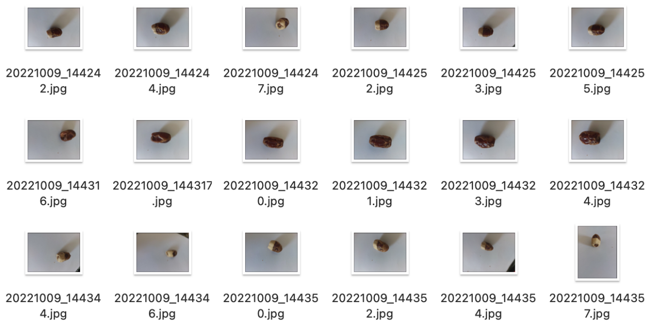

Figure 4.1: Segai

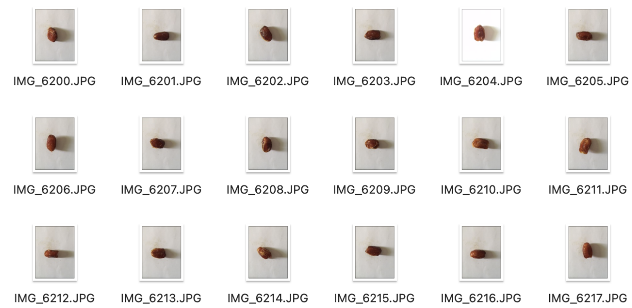

Figure 4.2: Khalas

The data that is fed to the model are in a form of pictures which then are put into a function to modify its size, resolution colors, illumination, texture or compositions and they are taken in single pictures. we spent a large amount of time collecting suitable dataset to obtain acceptable result which depends on the training accuracy so whenever the accuracy is low we would increase the data-set. As shown in these 4.3 The data are taken individually to reduce a known error in machine learning called noise, another reason why they are taken individually is due to some date's feature are easily detected by being individual which can effect the final result(accuracy).

Figure 4.3: Khenaizi data-set

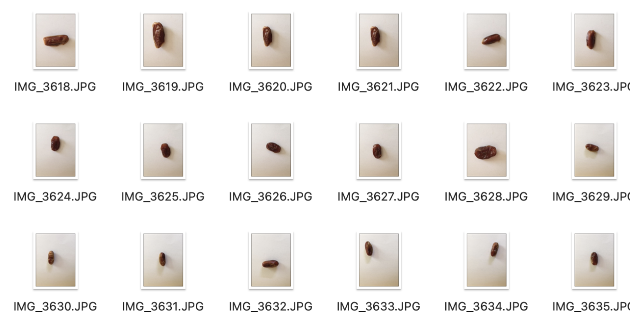

Figure 4.4: Barni

4.2 Materials and Methods

The method implemented in this project is from [19]. The main idea of this method is to use transfer-learning and to leverage the ability of pre-train image classification model on large-scale image-classification task, the intuition behind this is to serve as generic model of the visual world. and taking advantage of its convolutional layer without starting from scratch. There are good image classification models available in TensorFlow hub site [18] that can be implemented in this method, we chose MobileNet-V3 as it is suitable for mobile application and it is the latest version available. Down are in detail description about the methodology used, from preprocessing the dataset to model training.

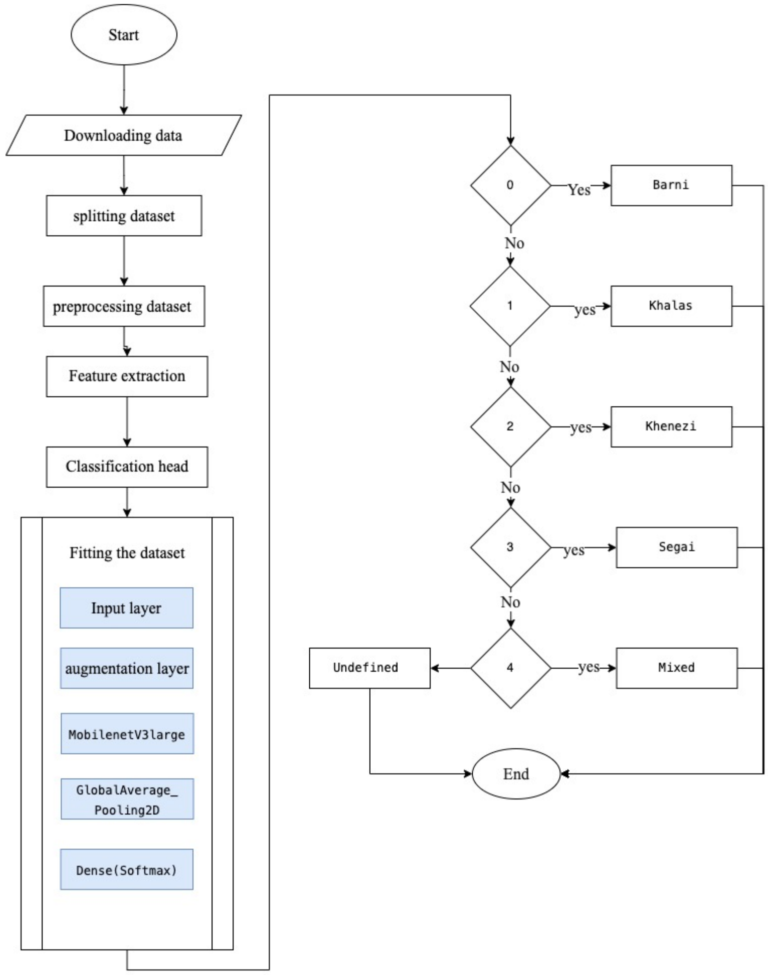

Figure 4.5: Methodology

4.2.1 Dataset preprocessing and dataset splitting.

The data-set contains 7597 pics for total of five classes. To split the data-set for training and testing, the python splitfolders [5] . The module is used to automate the process of splitting the data into three folders: testing, validation and training as shown in table Table 4.1. Module will split the dataset in the following way, 80% training, 10% testing, and 10% validation. Next step in processing the dataset is data augmentation to add more sample diversity, by applying random rotation using keras.layers.RandomRotation [] and random flip using keras.layers.RandomFlip []. In addition the image resolution had to be modified to 224*224 and rescale the image color from [0,255] range to range from [0,1] because of the MobileNet-V3 requirements. Last step in preprocessing is configuring data for performance by adding buffering prefetch using train-dataset.prefetch() method [6].

4.2.2 Feature extraction & Classification head.

The feature extractor in MobileNet-V3 convert each 224*224*3 image to 7*7*960 block of feature. To generate a prediction from a block of feature we used keraslayersGlobalAveragePooling2D() [10] to convert feature block to 960-element vector per image but before building feature extractor, freezing the convolutional layer is important to prevent the weight in this layer from changing, this is done by setting layer.trainable = False. Then adding Dense layer to convert feature vector into single prediction per image.

4.2.3 Creating the model & learning process

To build our model we will use of MobileNet-v3 a pre-trained model developed by google but without including the top layer() which is the classification layer. This model is trained on ImageNet (ILSVRC-2012-CLS) [4] contain 1.2 million images and 1000 categories. The knowledge base in the pre-train model will help us classify date fruit in our data.

The data augmentation and feature extractor steps described above are added to the final model before compiling it. We compile the model using adam optimizer known for its efficiency in memory usage and for large parameters problem [9]. We also used SparseCategoricalCrossentropy [11] as loss function for its suitability for two or more label of classes.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_13 (InputLayer)       [(None, 224, 224, 3)]     0

 sequential_8 (Sequential)   (None, 224, 224, 3)       0

 MobilenetV3large (Functiona  (None, 7, 7, 960)        2996352
 l)

 global_average_pooling2d_3   (None, 960)              0
 (GlobalAveragePooling2D)

 dropout_5 (Dropout)         (None, 960)               0

 dense_4 (Dense)             (None, 5)                 4805

=================================================================
Total params: 3,001,157
Trainable params: 4,805
Non-trainable params: 2,996,352
_________________________________________________________________
```

Figure 4.6: Model summary

When using .predict() method [20] in our trained model, it will return an array of length five. Each index corresponds to one of our classes as shown in the table 4.1 above, to find the model prediction we simply need to find the index with the largest value.

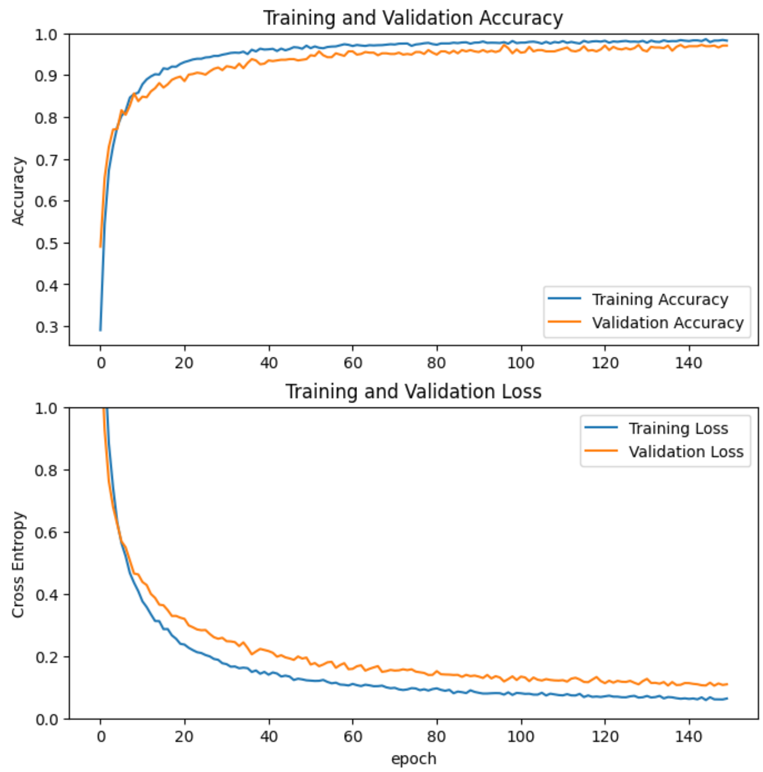

Figure 4.7: Loss Graph

4.2.4 Predictions.

To test our Model we made an app using google flutter. Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, mac-OS, Windows, Google Fuchsia, and the web from a single codebase. There are many packages used to design the app including:

Tessorflow lite

tflite/tflite.dart package [17] is a flutter plugin for accessing TensorFlow Lite API, it supports image classification, object detection (SSD and YOLO), Pix2Pix and Deeplab and PoseNet on both iOS and Android.

Using a this package provides a flexible and fast solution for accessing TensorFlow Lite models in the flutter applications making it easy to perform on-device machine learning.

Image picker

A Flutter plugin [16] for iOS and Android. With the help of the image picker package in the Flutter framework, developers may quickly select and edit images in their Flutter applications by accessing the device's camera and gallery. The package offers a straightforward and user-friendly API for selecting and cropping photos as well as capturing pictures using the device's camera. The image picker package's capability to select photos from the device's camera and gallery is one of its standout features. Users can now choose an existing photo or take a new one. The package offers the option to trim and adjust an image once it has been chosen so that it can be utilized in the app.

Dart IO

A powerful Http client for Dart, which supports Interceptors, Global configuration, FormData, Request Cancellation, File downloading, Timeout etc... [15].
The dart:io package's capability for file I/O operations like reading and writing files is one of its key features. This makes it simple for developers to create, read, write, and delete files and directories, among other file system operations. The program additionally offers assistance for carrying out file system tasks like renaming and relocating files.

Dart image

A set of classes and functions are provided by the built-in package dart:io igman the Dart programming language for conducting input/output (I/O) operations in a Dart application. Developers can use this package to access the file system, read and write files, and carry out additional I/O activities including socket and HTTP connection.

The dart:io package's capability for file I/O operations like reading and writing files is one of its key features. This makes it simple for developers to create, read, write, and delete files and directories, among other file system operations. The program additionally offers assistance for carrying out file system tasks like renaming and relocating files.

4.2.5 Flutter Widgets

An application needs a flawless interface and well-designed UI and compatible with a Mobile Platform. Flutter widgets provide these [3] Widgets are the building blocks of a Flutter app and are used to create the user interface. Flutter has a wide range of built-in widgets, and developers can also create their own custom widgets. This allows for a high degree of flexibility and customization in the design of the user interface.

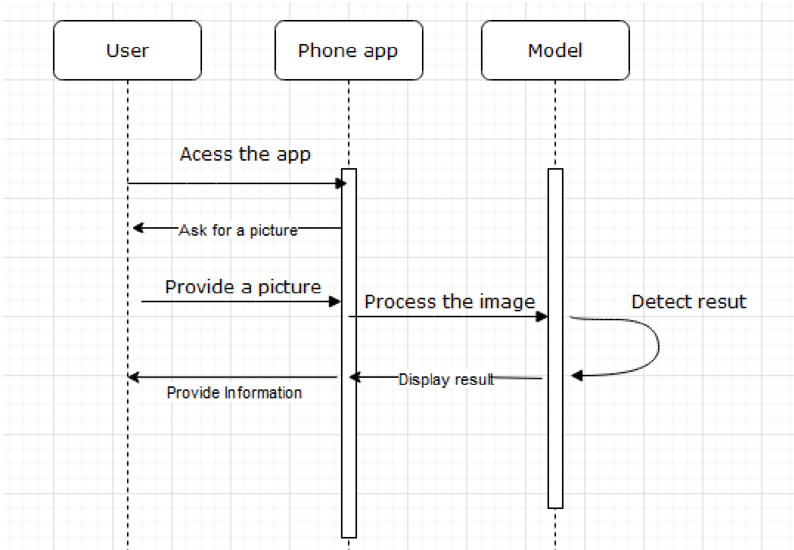

Figure 4.8: Fig 2.0 Sequence Diagram

Chapter 5

Results

There are a total of 5 different data-set consisting [Barni, Khalas, Segai, Khenaizi] and Mixed. Mixed Consists of [Barhi, Zahidi, Naghal and khudri]

5.0.1 Confusion Matrix

Confusion matrix is a way to measure and summarize the performance in a classification model visually , they can also be called error matrix, they consist of:
True Positive (TP):
The predicted value matches the actual value
The actual value was positive and the model predicted a positive value
True negatives (TN):
The predicted value matches the actual value
The actual value was negative and the model predicted a negative value.
False positives (FP):
The predicted value was falsely predicted
The actual value was negative but the model predicted a positive value
Also known as the Type 1 error
False negatives (FN):
The predicted value was falsely predicted
The actual value was positive but the model predicted a negative value
Also known as the Type 2 error

To calculate the accuracy Confusion matrix is needed:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5.1}$$

5.0.2 Calculations

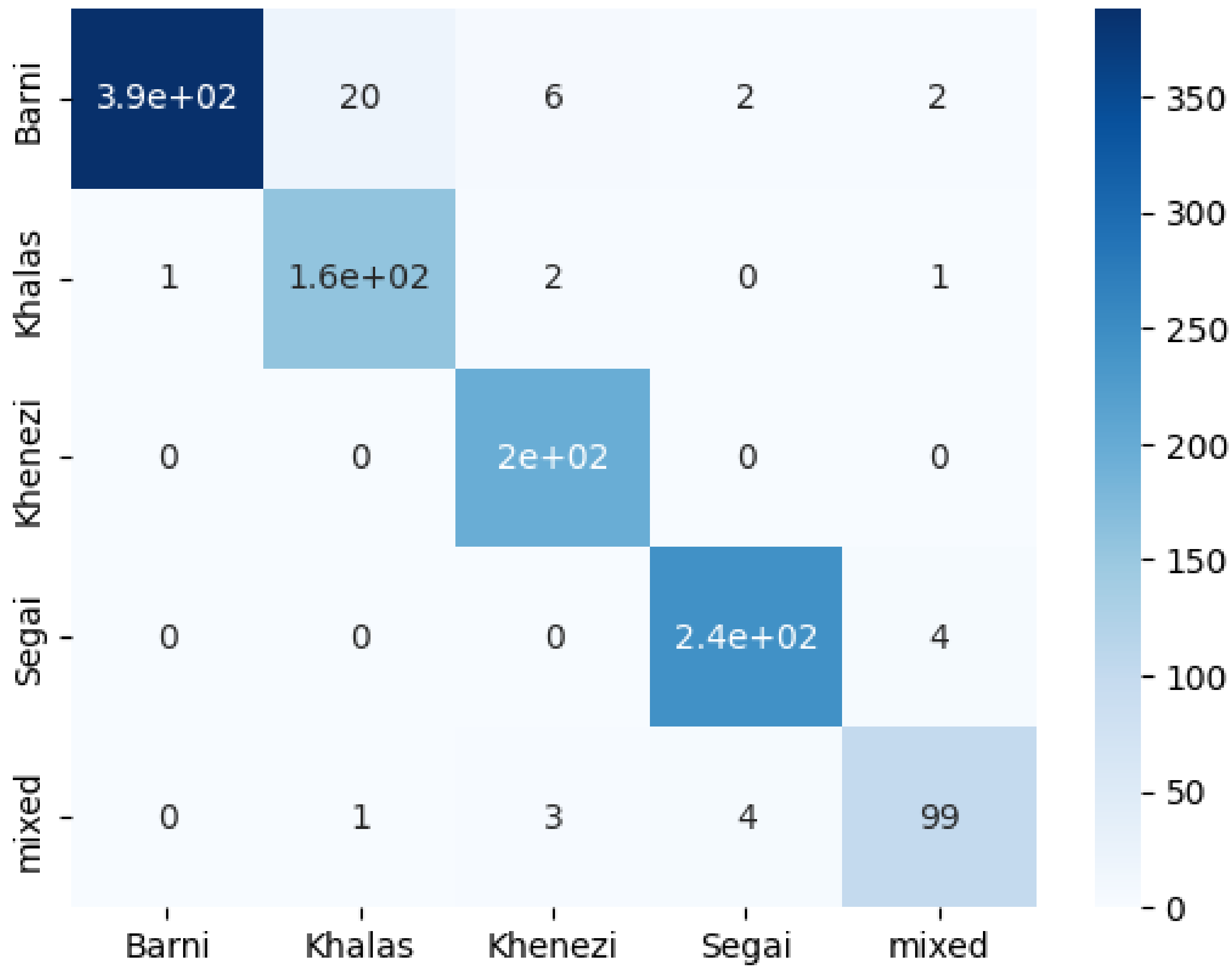

Figure 5.1: Conusion Matrix

5.1 Confusion Matrix

Calculations (TP,TN,FP,FN)

Barni:

TP: 390

TN:714

FP:1

FN:30

Khalas:

TP: 160

TN:951

FP:21

FN: 4

Kenaizi:	TN:922	FN:0
TP: 200	FP:11	

Segai:

TP:240

TN:786

FP:6

FN:4

Mixed:

TP:99

TN:1021

FP:7

FN:8

Accuracy:

Accuracy is one of the best way to measure a model's performance to a point. For it can provide the ratio of correct predictions to the total predictions. [2]

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5.2}$$

Barni : 390+714/(390+714+1+30) = **0.9726**

Khalas: 160+951/(160+951+21+4) = **0.9779**

Segai: 240+786/(240+786+6+4) = **0.9903**

Khenaizi: 200+922/(200+922+11+0) = **0.9902**

Mixed: 99+1021/(99+1021+7+8) = **0.9867**

Precision:

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. High precision relates to the low false positive rate. [2]

$$Precision = \frac{TP}{TP + FP} \tag{5.3}$$

Barni: 390 / (390 + 1) = **0.9974**

Khalas: 160 / (160 + 21) = **0.8839**

Segai: 240 / (240 + 6) = **0.9756**

Khenaizi: 200 /(200 + 11) = **0.9478**

Mixed: 99 / (99 + 7) = **0.9339**

Recall:

Recall is the ratio of correctly predicted positive observations to the all observations in actual class. [2]

$$Recall = \frac{TP}{TP + FN} \tag{5.4}$$

Barni: 390 / (390 + 30) = **0.9285**

Khalas : 160 / (160 + 4) = **0.8756**

Segai: 240 / (240 + 4) = **0.9836**

Khenaizi: 200 /(200 + 0) = **1**

Mixed : 99 / (99 + 8) = **0.9252**

F1 Score:

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it’s better to look at both Precision and Recall. [2]

$$F1Score = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{5.5}$$

Barni: 390 / 390 + 0.5(1+30) = **0.9617**

Khalas: 160 / 16-0 + 0.5(21 + 4) = **0.9275**

Segai: 240 / 240 + 0.5(6 +4) =**0.9795**

Khenaizi: 200 /200 + 0.5(11 + 0) =**0.9732**

Mixed: 99 /99 + 0.5(7 + 8) = **0.9795**

	Precision	**Recall**	**f1-Score**	**Accuracy**	**Support**
0_Barni	0.9974	0.9285	0.9617	0.9726	418
1_Khalas	0.8839	0.8756	0.9275	0.9779	161
2_Khenaizi	0.9756	0.9836	0.9795	0.9903	196
3_Segai	0.9478	1	0.9732	0.9902	248
4_Mixed	0.9339	0.9252	0.9795	0.9867	107

Table 5.1: Testing Result

5.1.1 Flutter Result

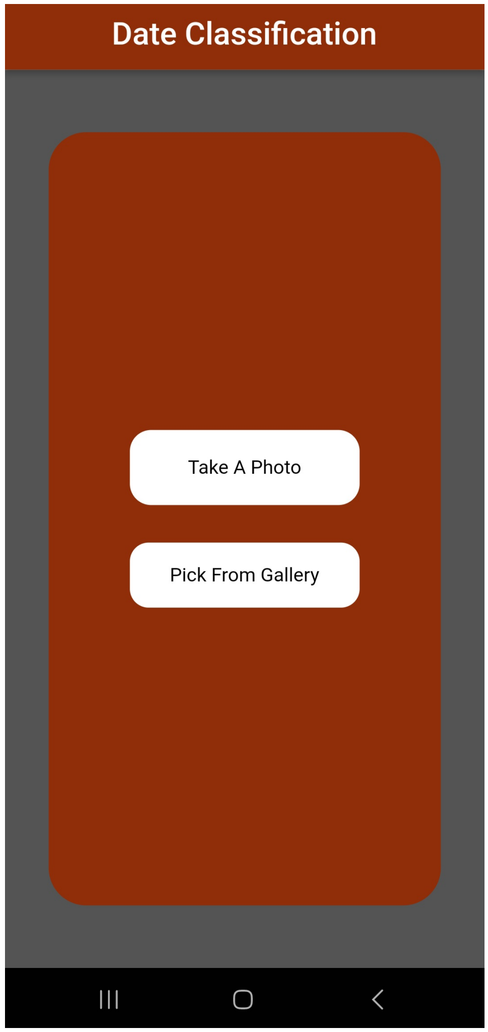

Figure 5.2: HomePage

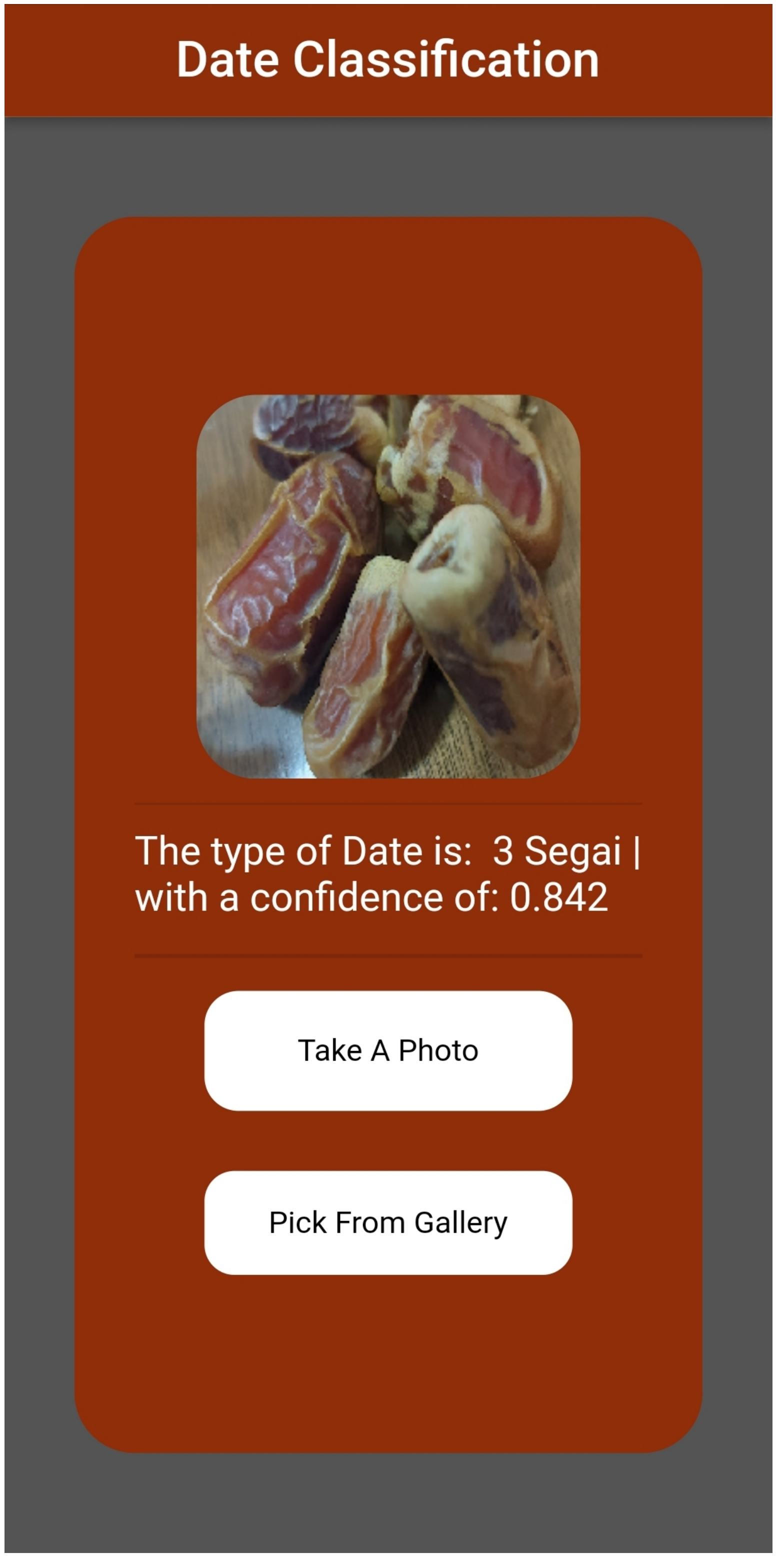

Figure 5.3: Test #1

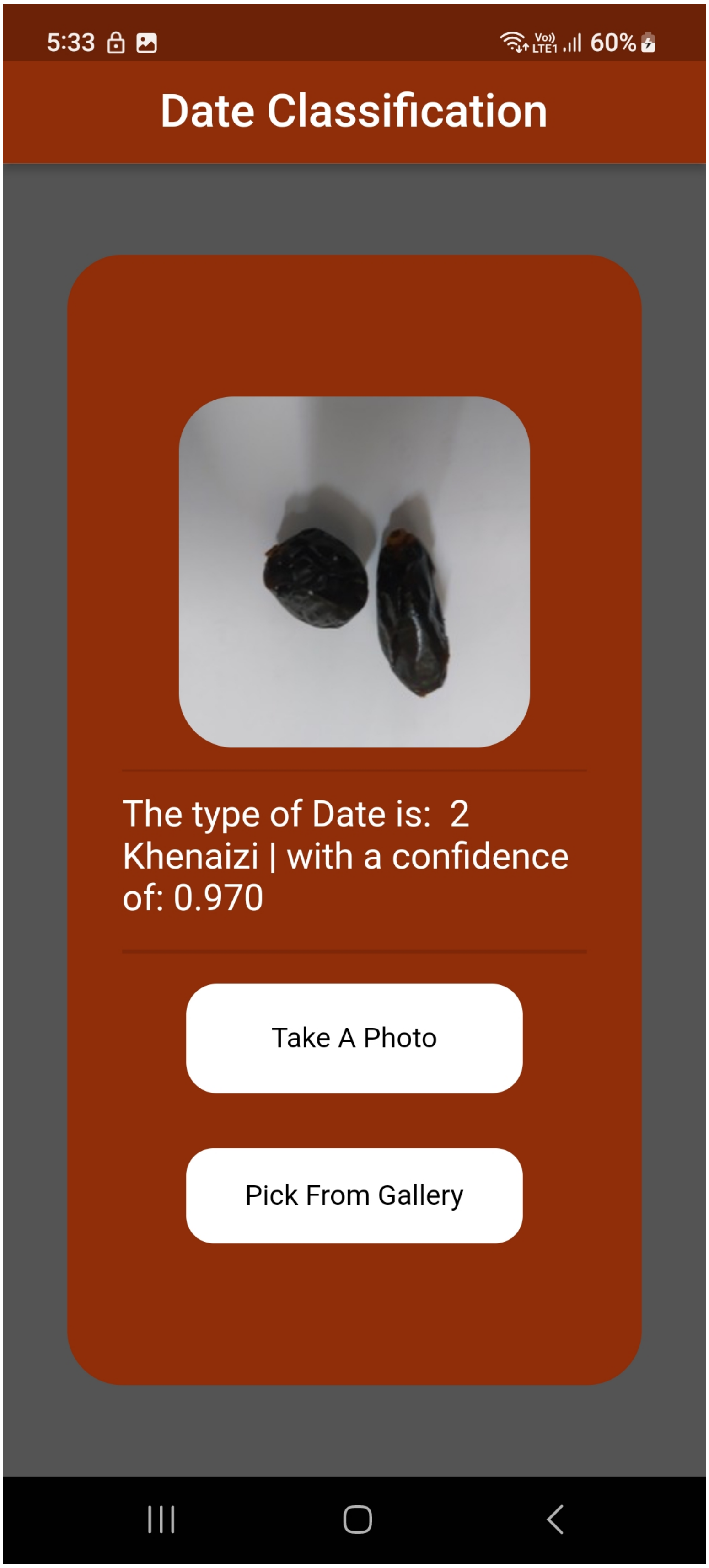

Figure 5.4: Test #2

Chapter 6

Conclusion

We were able to show how well convolutional neural networks worked for categorizing date fruits. With a performance of 96%, the model was able to correctly identify several varieties of date fruits.

As a future improvement, we plan to expand our dataset by adding more types of date fruits and different stages of maturity. This will help further improve the model's classification capabilities and make it more robust. Additionally, we will explore the possibility of deploying the model accessible to users in the agricultural industry. Overall, this research has the potential to improve the way date fruits are classified and improve efficiency in the industry.

The app was tested on a device running Android and it was able to classify images with high accuracy. The user interface was designed to be user-friendly, making it easy for the user to select an image and view the classification results.

the image classification app developed in this project demonstrates the capabilities of the Flutter framework in creating powerful and efficient image classification apps. The app can be further enhanced by adding more features and increasing the number of categories that the model can classify.

Bibliography

[1] Siddhika Arunachalam, Harsh Kshatriya, and Mamta Meena. Identification of defects in fruits using digital image processing. *International Journal of Computer Sciences and Engineering*, 6:637–640, 10 2018. doi: 10.26438/ijcse/v6i10.637640.

[2] blog.exsilio.com. Calculations. https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/#:~:text=F1%20score%20%2D%20F1%20Score%20is,have%20an%20uneven%20class%20distribution.

[3] flutter.dev. Flutter widgets. https://docs.flutter.dev/development/ui/widgets.

[4] ImageNet. Imagenet large scale visual recognition challenge. https://www.image-net.org/challenges/LSVRC/2012/.

[5] Johannes Filter. Johannes split folders. https://pypi.org/project/split-folders/, .

[6] Johannes Filter. tensorflow.org. https://www.tensorflow.org/api_docs/python/tf/data/Dataset#prefetch, .

[7] keras.io. Keras randomrotation layer. https://keras.io/api/layers/preprocessing_layers/image_augmentation/random_rotation, .

[8] keras.io. keras image data preprocessing. https://keras.io/api/preprocessing/image/, .

[9] keras.io. Keras adam algorithm. https://keras.io/api/optimizers/adam/, .

[10] keras.io. Keras pooling layers. https://keras.io/api/layers/pooling_layers/global_average_pooling2d/, .

[11] keras.io. Keras probabilistic losses. https://keras.io/api/losses/probabilistic_losses/#sparsecategoricalcrossentropy-class,.

[12] Ghulam Muhammad, Muhammad Hussain, Fatimah Alenezy, George Bebis, Anwar Mirza, and Hatim Aboalsamh. Recognition from face images using weber local descriptor. *2012 19th International Conference on Systems, Signals and Image Processing, IWSSIP 2012*, 04 2012.

[13] Sapan Naik and Bankim Patel. Machine vision based fruit classification and grading-a review. *International Journal of Computer Applications*, 170(9):22–34, 2017.

[14] Nanonets.com. Data augmentation. https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-par

[15] pub.dev. Dartio flutter package. https://api.flutter.dev/flutter/dart-io/dart-io-library.html,.

[16] pub.dev. image picker flutter package. https://pub.dev/packages/image_picker,.

[17] pub.dev. tflite package. https://pub.dev/packages/tflite,.

[18] tensorflow.org. image classification models in tensorflow hub. https://tfhub.dev/s?module-type=image-augmentation,image-classification,image-classification-logits,image-classifier,image-feature-vector,image-generator,image-object-detection,image-others,image-pose-detection,image-segmentation,image-style-transfer,image-super-resolution,image-rnn-agent,.

[19] tensorflow.org. Create the base model from the pre-trained convnets. https://www.tensorflow.org/tutorials/images/transfer_learning#create_the_base_model_from_the_pre-trained_convnets,.

[20] Tensorflow.org. tesorflow model predict. https://www.tensorflow.org/api_docs/python/tf/keras/Model#predict.

[21] Xue Ying. An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168(2):022022, feb 2019. doi: 10.1088/1742-6596/1168/2/022022. URL https://dx.doi.org/10.1088/1742-6596/1168/2/022022.