Design and Development of a mobile application for children education using machine learning

BY

Sarwar Ahmed

ID: 191-15-12513

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

Supervised By

Ms. AFSARA TASNEEM MISHA

Lecturer

Department of CSE

Daffodil International University

Co-Supervised By

Md. Sadekur Rahman

Assistant Professor

Department of CSE

Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

24 January 2023

APPROVAL PAGE

This Project titled "**Design and Development of a mobile application for children education using machine learning**" submitted by Sarwar Ahmed, ID: 191-15-12513; to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 24 January 2023.

BOARD OF EXAMINERS

Chairman

____

Dr. Touhid Bhuiyan
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

SArefin 24.01.23

Internal Examiner

____

Dr. Mohammad Shamsul Arefin
Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

24.01.23

Internal Examiner

____

Md. Sabab Zulfiker
Senior Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

24.1.2023

External Examiner

____

Dr. Ahmed Wasif Reza
Associate Professor
Department of Computer Science and Engineering
East West University

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Ms. Afsara Tasneem Misha, Lecturer, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

Afsara

Ms. Afsara Tasneem Misha
Lecturer
Department of
CSE
Daffodil International University

Co-Supervised by:

Sadek

Mr. Md. Sadekur Rahman
Assistant Professor
Department of
CSE
Daffodil International University

Submitted by:

24.01.2023

Sarwar Ahmed
ID: 191-15-12513
Department of
CSE
Daffodil International University

ACKNOWLEDGEMENT

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

We are really grateful and wish our profound indebtedness to **Ms. Afsara Tasneem Misha,** Lecturer, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of Machine Learning to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to Dr. Touhid Bhuiyan, Professor and Head**,** Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

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

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

ABSTRACT

The purpose of this project is to demonstrate how any type of fruit, vegetable, or animal can be rapidly clarified, illustrated, and given a suggested sentence that goes along with it. Children can use a straightforward mobile application to learn the name of the fruit they want to see. The user uploads a picture of any fruit, vegetable, or animal to the classification area where they can see the precise names of the objects in the picture. How users can use this application in their everyday lives and what kind of programming is required for image classification. This presentation will conclude with a suggestion for a live picture classification system, designed specifically to help kids expand their knowledge. The image would be found by an image classification system, which would then show the categorization in a text box. The concept is built upon image classification, which is inherently very challenging to put into practice. Using a simple smartphone application, kids may discover the name of the fruit they want to see. Any animal, plant, or fruit may be photographed and uploaded to the category area, where users can see the precise names of the objects in the picture. How to use this application in everyday life and what kind of programming is required for photo classification. This talk will conclude with a suggestion for a live picture classification system designed specifically to help kids build their knowledge. Locating the image and displaying the classification in a text box are two examples of an image categorization approach. The concept is built on the classification of images, which is inherently very difficult to accomplish. Because we combined the Google Colab platform with the Google Flutter mobile application and the Google Dart language, we will employ the most well-known SDK (Software Development Kit) MLKIT (MLKIT digital ink recognition and google MLKIT picture labeling) for the project's advancement. An Android app will be released for the project.

TABLE OF CONTENTS

CHAPTER CONTENTS

LIST OF FIGURES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Introduction

In the field of education, image classification is still quite new. Essentially, it is utilized in a technical or contemporary setting where software developers or corporations use this equipment to more effectively utilize technology. For some people in the past, it was almost a fantasy, but students now are aware of image classification in the modern educational system. The primary purpose of using IC in the educational sector is to help students gain a foundational understanding of cutting-edge technologies. Students have the option to develop their talents by using this knowledge. The primary objective of educational institutions is to raise the standard of instruction. We reside in a contemporary era. We are seeing many changes in our daily lives today that seemed unimaginable a few decades ago. It is now simpler than ever to classify the names of any fruits, vegetables, or animal life. Now that they can quickly view the name and identify the things, they have access to the fund's name. We can now not only display any fruit but also show the name and accuracy of the result thanks to the rapid advancement of technology and the abundance of SDK (Software Develop Kit) available to programmers nowadays. Raising the bar for teaching is the main goal of educational institutions. In the area of education, seeks to provide students with a solid grasp of contemporary technology. A few decades ago, many of the changes we are now seeing in our daily lives were unimaginable. Sorting the names of any fruits, vegetables, or living things is now easier than ever. Image classification first appeared in this context. We can put to rest any lingering questions we have about the identity of any object that can only be captured using a mobile application by utilizing this area of technology.

1.2 Motivation

When children travel, they may come across unfamiliar fruits, vegetables, and animals, but they may not know what kind of fruit it is. Additionally, when they visit an open-air zoo and come across an animal they are unsure of its species. Children may try to write on the board but are unable to see the proper text. All of these issues can be solved with machine learning. This issue's motivation is both general and specific. Data is frequently insufficient for specialized picture categorization applications. This is particularly true in real life, where it is extremely challenging to get data from every state and country. Due to a lack of information, it is impossible to classify any fruit, vegetable, or animal species, among other crucial responsibilities. Techniques that blend pre-trained models with domain knowledge have been developed. People will find it very beneficial if we have a system that enables them to take a photo, upload it to the system, and view the name. In essence, picture categorization is a sophisticated form of computer technology that can represent any kind of item and has some use in the field of education.

Objectives:

Check the object's true name.

- This system can be used anywhere, with or without an internet connection.
- There are numerous options for object detection.
- There is an option to view text that is suggested based on drawn text.
- After uploading any object image, users can view the accuracy level.
- Also, there we can detect multiple objects.
- Add many data to detect the object.
- There is an option to see multiple object name on the same page.
- After Capturing the image there is showing multiple object names on the label section.

1.3 Rationale of the Study

Image classification, which has some use in the field of education, is essentially a sophisticated computer technique for categorizing any form of item. We can create an application that will enable us to see the tagging of items in real-time through some procedures using picture classification. can be a picture or a movie. A set of features that were taken from the image and added to it as a result of image processing is the end result. The majority of image processing methods work by treating images as two-dimensional signals and then subjecting them to common signal processing methods. The goal of image processing is to produce meaningful results that meet particular requirements. The smartphone camera becomes a live tagging camera when using the SDK. The text box in the mobile application displays the image label when the camera recognizes the image of the real thing. Traffic camera systems frequently involve image classification. He has a very bright future there, it won't be incorrect to claim that. If we broaden our focus, however, we may apply this technology to other domains, such as spotting any virus on humans, veggies, or farmland where users can test the image categorization system.

1.4 Research Questions

Every research project's research questions are an essential and significant component. It needs to be brief, clear, and important. In essence, for a research question to be relevant to the study topic, it must directly address it through data analysis. The questions about the study that we will try to address in the research report are listed below:

- What kind of data will we require to complete our analysis?
- What should be done to prepare the dataset?
- Acquiring knowledge of the analytical algorithm?
- How can the method be compared to other methods used in related fields?
- Who will be the project's target audience?
- How can we use the app to become more admirable?

1.5 Expected Output

- Our research is anticipated to offer practical answers to issues. Some of them include:
- They can recognize any type of unidentified fruit image.
- Used this method both online and offline.
- Our system makes an effort to display item labels with 90% correctness.
- Receiving the text recommendation via digital ink recognition.

1.6 Project Management and Finance

Due to the fact that it is founded on research, this project has three useful components. They are algorithms based on machine learning, report authoring, and mobile programming. Different tasks are given to team members. For this project to be constructed successfully, numerous investments will be needed. Because this project is mobile-based, costs have been made for it. Our team consists of just one person; hence the total expense is divided into just one part.

1.7 Layout of Report

Our project report is organized as follows:

- Chapter One contains the project introduction, project motivation, research questions, projected results, and report structure.
- Chapter 2 includes a literature overview, our viewpoint on this industry, related work, a research summary, the scope of the problem, and problems.
- The portion on research methods is in Chapter 3.
- Chapter 4 includes Experimental Results and a Discussion of the project's development.
- Chapter 5 contains the Conclusion and Future outlook 5.

CHAPTER 2
BACKGROUND

2.1 Preliminaries/Terminologies

In today's technologies, we are used to detecting any branded item. We recognize the image and name the elements using a smartphone application, while there are other websites that are well-known for their online object recognition and labeling services. We can employ a new technology called image categorization.

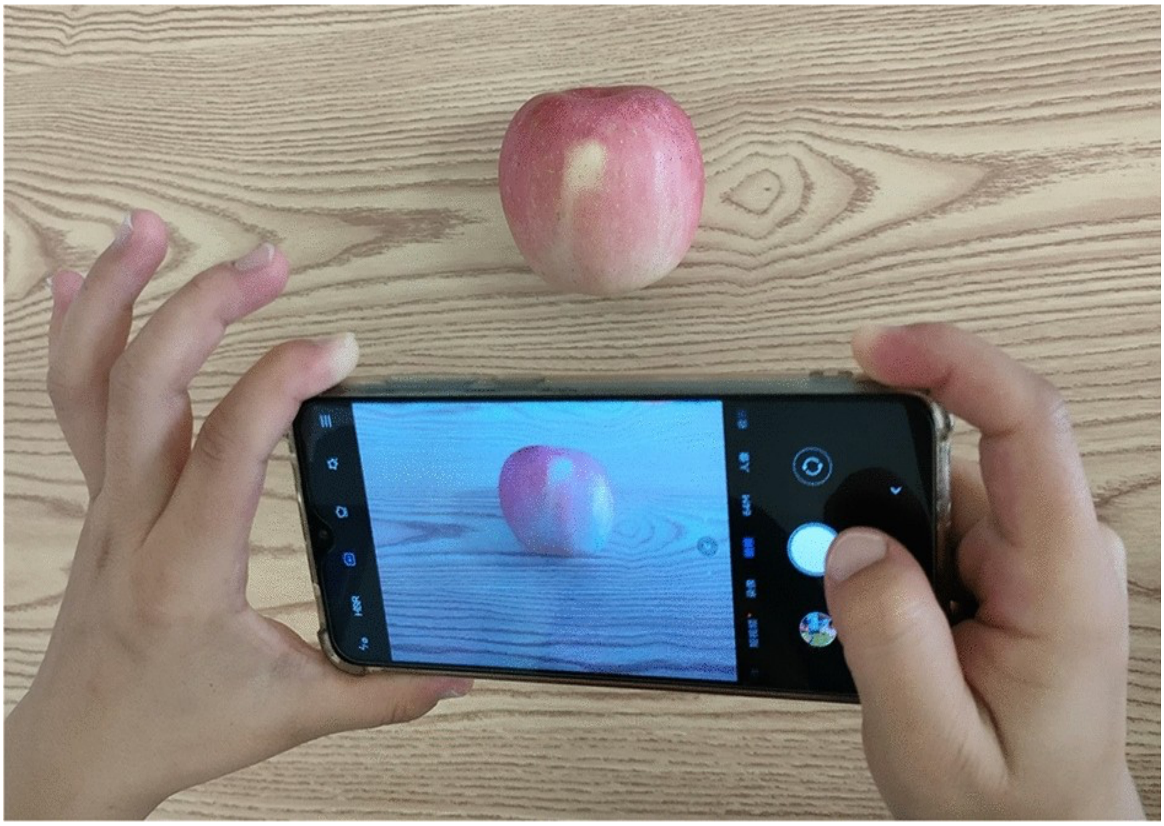


Fig 2.1: Benefits of Image classification in daily-life

Image classification is a procedure that enhances everyday experiences by adding computer-generated perceptual data to everyday things. A variety of methods can be used to represent an object.

Here, image classification is the most effective technique to represent an item from daily life. The main metric for assessing image recognition software is picture identification accuracy. Speed and adaptability are typically taken into account afterward. A real-life item needs access to labeled data that can be used as a test set in order to compare various possibilities. Many times, solutions that are taught using their own data perform better than those that are gathered and trained beforehand. if a pre-trained solution is already available and would produce the needed accuracy, one can decide not to spend the money to design a custom model. New Image processing apps can be designed and made using the technological software engine known as the Software Development Kit (SDK). Different types of SDKs, such as MLKIT (image labeling), MLKIT (digital ink recognition), etc., are used to create picture classification apps. With such SDKs, a virtual camera is opened, an object is detected, and the labeling name is then added to the text box section.

2.2 Related work

Google lens:

Google Lens is an image identification tool that uses neural network-based visual analysis to identify things in images and present pertinent information about those objects.

Dog detection app:

This approach aids in the detection of all dog breeds and aids in the identification of individual dogs by showing which breeds they belong to. displaying the accuracy outcome on a different side as well.

Photo Sherlock Search by photo:

The app offers a picture search based on photos taken with a camera or from a gallery. can be used to discover details about pictures on the internet, for instance, to identify the true owner of a picture from a social network

Search By Image app:

With the use of the live camera and the search platform known as Search By Image, you may snap any photo you like and look up any picture information you need.

2.3 Comparative Analysis and Summary

The research report makes it clear that a way will be suggested that may be very helpful for users and for people who find it confusing when fresh fruits are not sorted. We combine several instructions into a single platform to produce a better user interface (User Interface). inside of this system. Many companies can utilize this technology to give their customers an overview of dog and cat detection in retail situations.

2.4 Scope the problem

- Analyze several dataset kinds in this situation.
- The issue with the data's shifting format
- Accuracy is not entirely upheld.
- Increased API requirements for iOS and Android.

2.5 Challenges

- challenges we encounter
- The lack of sufficient learning resources is due to the fact that image categorization is a very recent development in technology.
- Because we use several types of SDK, it won't function on devices with lesser configurations.
- To work with image categorization, a device with a higher configuration is required.

CHAPTER 3
RESEARCH METHODOLOGY

3.1 Research Subject and Instrumentation

In this paper, we provide a fresh classification system and method for locating plant diseases. As demonstrated, the suggested system will be able to identify and categorize six different classes. The suggested approach is one of the few in the body of research that can classify six different categories. Most publications in the currently available literature fall into one of three to five categories. The recommended deep CNN acquisition will go through preprocessing steps where the images will be treated to background removal, scaling, and augmentation in an effort to learn the approach. Data augmentation is also used to expand the dataset. Despite the authors' inability to address the potential problems with overfitting that were highlighted in the literature study, the majority of papers as well as existing works use scant information, which can lead to overfitting. In this study, we use data augmentation, which simply modifies the original photos in a few minor ways to produce brand-new pictures. The background will be eliminated, the pictures will be resized, and they will be upgraded for the proposed deep CNN acquisition in order to better grasp the approach. In this study, we use data augmentation, which effectively modifies the original photos in a small number of ways to produce new images. Despite the authors' failure to address the potential problems with overfitting that were recognized in a literature review, the majority of publications still use insufficient information, which might lead to overfitting. The nodes of a graph are mathematical operations, while the edges of a network are multidimensional data arrays known as tensors. The fundamental components of a TensorFlow system are one or more worker processes and the client, which interacts with the master through the Session interface. Each worker process is responsible for allocating computational resources to appropriate users and for managing the graph nodes on those resources in accordance with the master's instructions. Examples of minor modifications include rotation, translation, and scale-in/scale-out.

3.2 Data collection procedure/Dataset Utilized

It's critical to choose a data-gathering platform after deciding on an image capture sensor for the project. The datasets for the studies in this review that used machine learning to pinpoint plant diseases were gathered using a variety of methods. Researchers frequently adopt the technique of gathering datasets via web tools. The fruit, vegetable, and animal dataset has become the most common data set for training and developing models based on machine learning to identify plant diseases and gauge their severity. The dataset as a whole contains 8,309 images.The majority of the images were shot in a lab environment with consistent backdrops and controlled lighting. Since these images are not representative of conditions that might occur in the field, using machine learning models trained to recognize images from the dataset of fruits, vegetables, and animals could not generalize more accurately to pictures taken there.

Fruits , vegetables and animal	Dataset data number
Banana	165
Orange	177
Grapes	190
Kiwi	109
Watermelon	199
Pineapple	200
Mango	160
Cucumber	162
Carrot	177
Capsicum	171
Onion	181
sweet potato	168
Paprika	200
Ginger	201
Panda	203
cat	260

Fig- Table-1 dataset info

3.3 Statistical Analysis

This study's network was developed, tested, and trained using the TensorFlow framework. Data flow graphs are used to carry out numerical computations in this open-source machine learning framework created by Google. The edges of the network represent tensors, which are multidimensional data arrays, while the graph's nodes represent mathematical operations. The fundamental components of a TensorFlow system are one or more worker processes and the client, which interacts with the master through the Session interface. Each worker process is in the responsibility of arbitrating access to one or more computational devices (such as CPU cores or GPU cards) and operating graph nodes on those devices in accordance with instructions from the master. TensorFlow has a number of noteworthy features, such as the capacity to map computations to multiple machines, which sets it apart from most other frameworks of a comparable nature; built-in support for automatic gradient computation; the capacity to partially execute subgraphs of the entire graph; and the capacity to add constraints to devices, such as ensuring that two or more objects are placed in the same space. computes a kernel tensor of the form [filter height, filter width] over the input of shape [batch, in height, in width, in channels], as well as a 2D convolution.

These operations are done by this one:

- Flattens the filter to have the following dimensions: [filter height * filter width * in channels, output channels] 2D matrix
- By removing image patches from the input tensor, it creates a virtual tensor with the following dimensions: [batch, out height, out width, filter height * filter width * in channels].
- For each patch, right-multiply the filter matrix and the picture patch vector.
- If the padding setting is set to "same," the input is padded to zero so that the output retains the same height and width; if it is set to "valid," the input is not padded to zero, so the output may be smaller across the width and height.

3.4 Proposed Methodology/Applied Mechanism

In deep learning, the concept of transfer learning refers to the strategy of using past models to tackle a new problem. Flow learning is a method of learning that is employed while working with photos, albeit it is not a specific type of DL algorithm. Finishing a new project requires using information from earlier training. The specific duty will be associated in some manner with the method that was employed; for instance, this may entail going to label things into a certain format. A considerable portion of an outdated training model is typically involved in making it adapt to the new, unknown input. In this scenario, three distinct deep learning algorithms were built using the transfer learning technique, which consumes fewer resources while delivering higher accuracy from its pretrained weights. Utilizing knowledge from prior training is necessary to complete a job. Making an old training model adapt to the new, unknown data often involves a significant amount of work. The particular task will be connected in some way to the approach used; for example, it could include going to label items in a certain format.

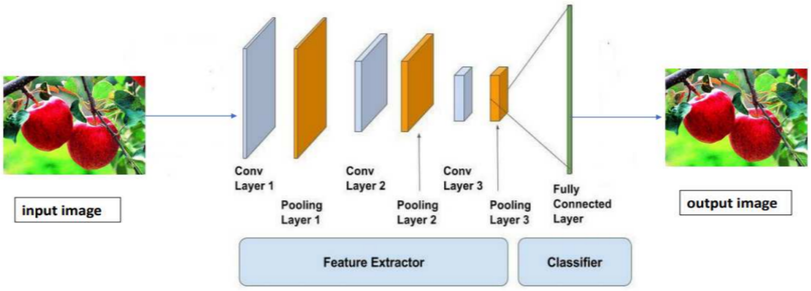


Figure-2.3 Using CNN, fruit picture feature extraction

CNN is a multilayer feed-forward NN that was created specifically to handle image data. It was influenced by biological NN. Convolutional and nonlinear activation pooling layers are used in CNN. For a three-dimensional input and filter, Convolution in two dimensions is carried out via the convolution layer. Height, weight, and input channels are abbreviated as HI WI C. Assume that each margin's fruit picture size is P and that the stride is S.

3.5 Implementation Requirements

Image improvement was used to enhance the quality of the baseline dataset, and augmentation was used to expand it in size. Smoothing and boosting picture detail results in photographs that are flatter and have better contrast. By modifying the locally aware edge contrast, this is achieved. This approach maintains the strong edges by defining a minimum ideal intensity amplitude as a criteria value. In this investigation, the threshold and enhancement values were both set at 0.15 and 0.5. An anisotropic diffusion filter is used in the method for lessening contrast. The center of the range zero-frequency element is moved using the Fourier transform. It is vital that the researchers take every precaution to prevent overfitting in any machine learning endeavor. The authors proposed many solutions, including augmentation, early stopping, dropout approach, stochastic pooling, L1 and L2 regularization, and dropout regularization, to address these issues. To increase the dataset and reduce the chance of overfitting, we advise employing data augmentation. Data augmentation is a simple process that entails making little changes to the original photographs in order to produce new ones. We use the following approaches in this study to cover scale-in/scale-out, rotation, and translation procedures. These three simple methods produce brand-new images that are almost exact replicas of the originals. After preprocessing, the dataset had very few photos. We had to make some adjustments to the raw photographs in order to increase the quantity while keeping in mind that substantially increasing the number of photos would result in oversampling. Create a dataset that enables the example to function as intended. The images were also flipped both vertically and horizontally, rotated between -15 and +15 degrees, sliced at a frequency of 0.2, and blown in and out at a frequency of 0.2. The fill type for the ImageClassification function was then set to constant black and did not have any degraded regions in order to ensure that the created pictures were consistent. Then, it was decided that the pictures would truly vary at random within a brightness range of between 0.5 and 1.5.

CHAPTER 4
EXPERIMENTAL RESULTS and DISCUSSION

4.1 Experimental Setup

This methodology is a process for collecting data or information that includes a number of concepts and notions. The "android Studio" platform can be used to implement the image classification method. We can do experiments relating to image categorization on this platform.

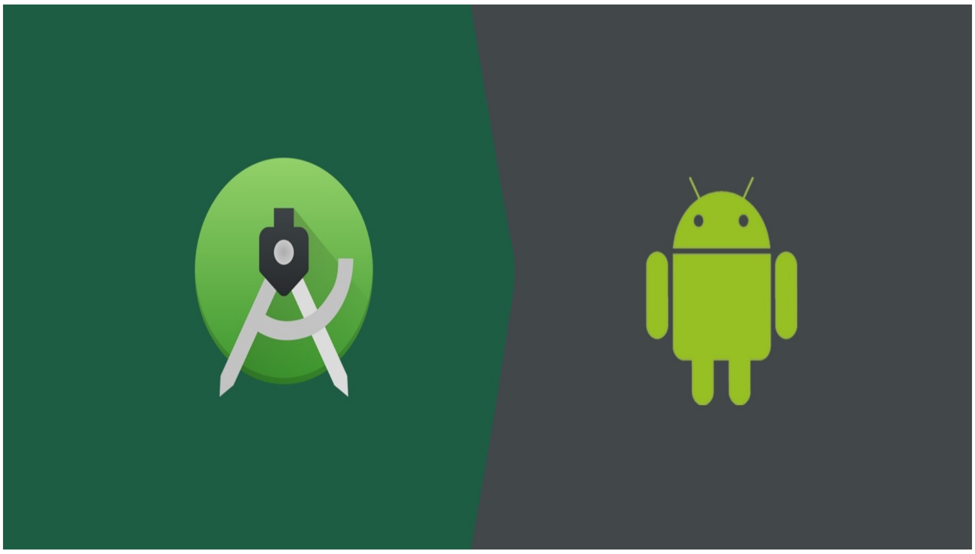
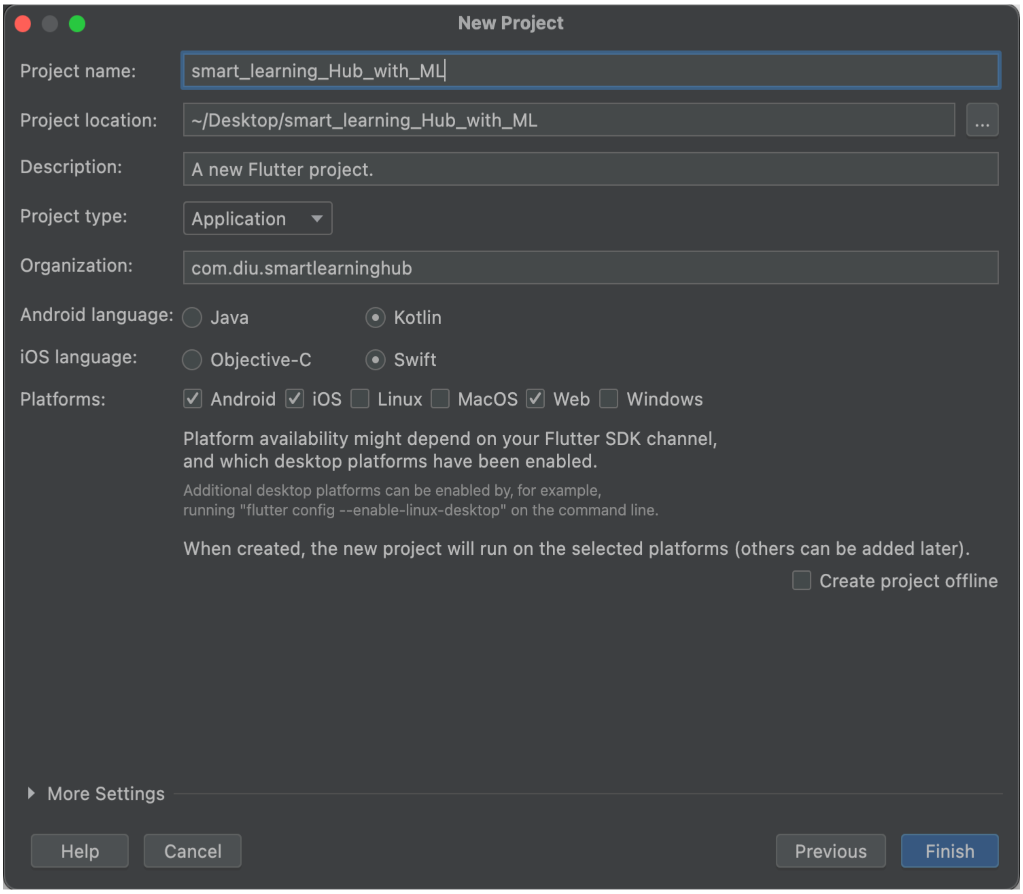


Fig 3.1: Working platform of capture image for image classification.

Most of us have heard of Android Studio, which enables cross-platform app creation, drag-and-drop functionality, and programming for gaming. Android Studio is the name of the official Integrated Development Environment (IDE) for Google's Android operating system, which is based on IntelliJ IDEA from JetBrains. It is available for download for Windows, macOS, and Linux-based operating systems.

Steps of the working process

The technique is a process used to collect data or information that incorporates numerous concepts and philosophies. New project environment creation The application settings must be managed before developing an application or system. In order for our project to operate without a hitch on the platforms it will be used on, we must set the appropriate information.



New Project

Project name: smart_learning_Hub_with_ML

Project location: ~/Desktop/smart_learning_Hub_with_ML

Description: A new Flutter project.

Project type: Application

Organization: com.diu.smartlearninghub

Android language: ○ Java ● Kotlin

iOS language: ○ Objective-C ● Swift

Platforms: ☑ Android ☑ iOS □ Linux □ MacOS ☑ Web □ Windows

Platform availability might depend on your Flutter SDK channel, and which desktop platforms have been enabled.

Additional desktop platforms can be enabled by, for example, running "flutter config --enable-linux-desktop" on the command line.

When created, the new project will run on the selected platforms (others can be added later).

□ Create project offline

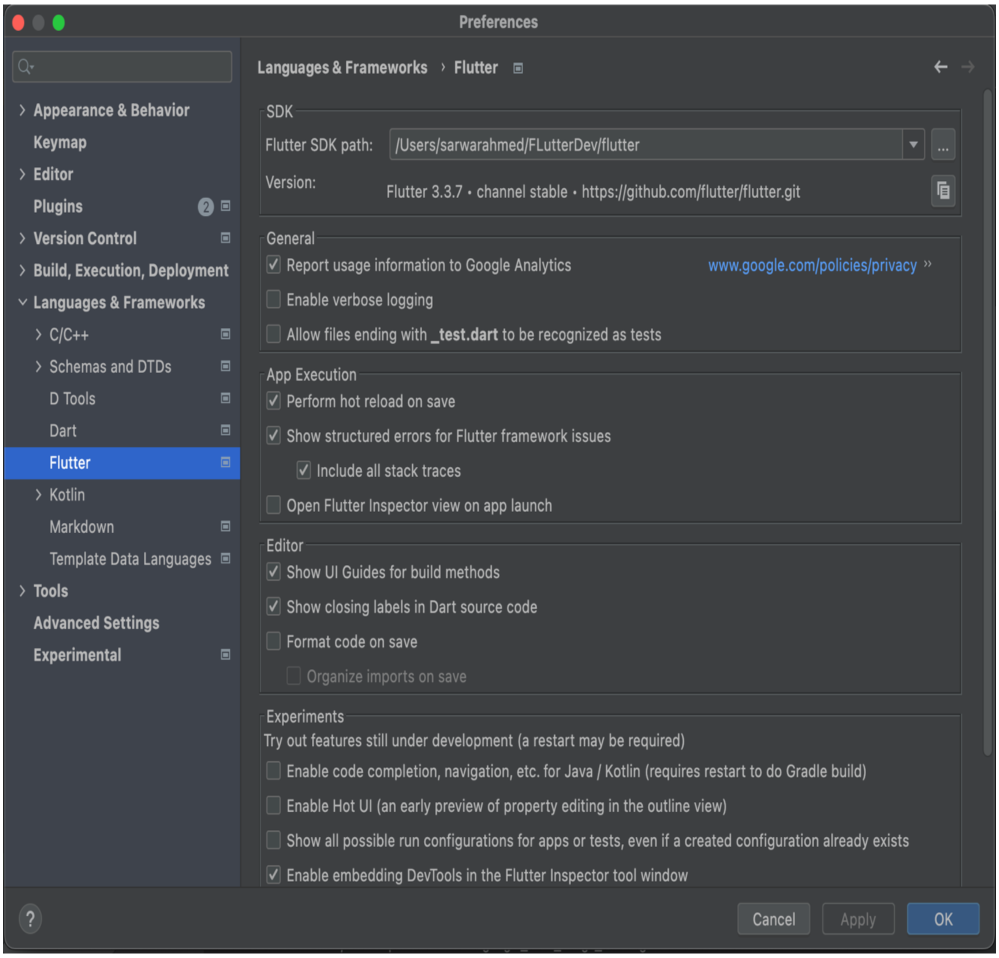
Fig 3.2.1: Create a new project and set-up

Prior to the development of an application or system, application parameters must be controlled. The method is a procedure for gathering data or information that combines several ideas and ideologies. We must set the proper information in order for our project to run well on the platforms it will be utilized on.

The project's settings are depicted in the following diagram. This means that we must choose the API level, the Project's name, the Package name, and the API level that the Project will support. The Android 9.0 Pie minimum API level is listed below. It demonstrates the project's support for the MLKIT SDK. The project path and the flutter SDK path must both be configured. The next step is to choose the programming language.

Build Settings:

After selecting Create new project settings, we must modify the settings for the environment in which we are creating this application. Build Settings are another name for this setting.



SDK

Flutter SDK path: /Users/sarwarahmed/FLutterDev/flutter

Version: Flutter 3.3.7 • channel stable • https://github.com/flutter/flutter.git

General

☑ Report usage information to Google Analytics www.google.com/policies/privacy »

□ Enable verbose logging

□ Allow files ending with **_test.dart** to be recognized as tests

App Execution

☑ Perform hot reload on save

☑ Show structured errors for Flutter framework issues

☑ Include all stack traces

□ Open Flutter Inspector view on app launch

Editor

☑ Show UI Guides for build methods

☑ Show closing labels in Dart source code

□ Format code on save

□ Organize imports on save

Experiments

Try out features still under development (a restart may be required)

□ Enable code completion, navigation, etc. for Java / Kotlin (requires restart to do Gradle build)

□ Enable Hot UI (an early preview of property editing in the outline view)

□ Show all possible run configurations for apps or tests, even if a created configuration already exists

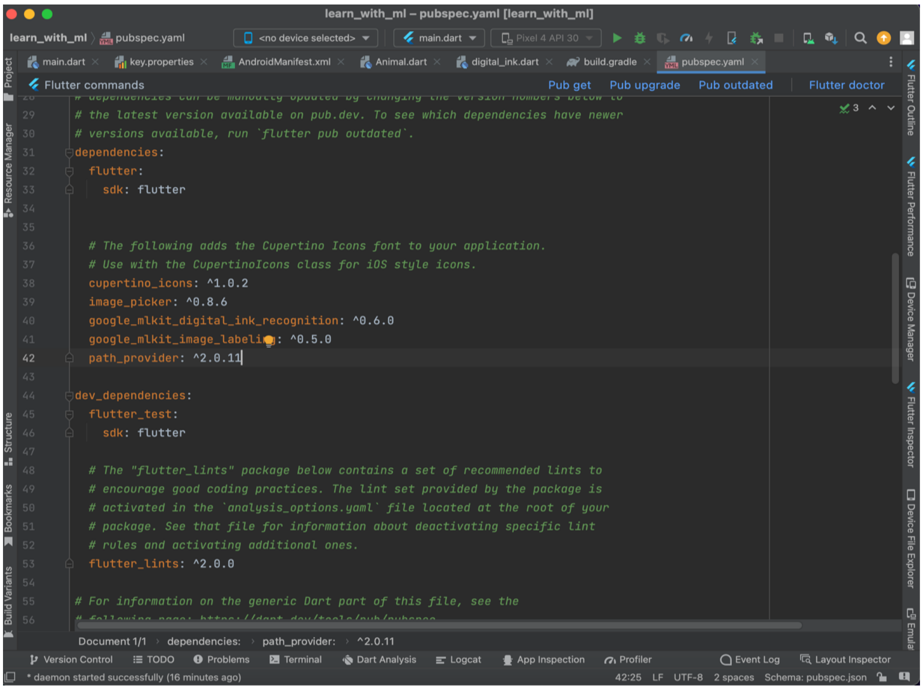
☑ Enable embedding DevTools in the Flutter Inspector tool window

Fig 3.2.2: Build settings of the project.

The illustration below shows that we have created the app for both Android and iOS. As we can see, there aren't many built-in sceneries. The flutter SDK framework is what we're using to create this project's application. Google's Flutter framework is used. For this project, we created our application using the most recent flutter SDK.

Dependency set-up

Because we utilize a plugin to make this project work properly on our mobile application, we need to set up the dependency.



```
# dependencies can be manually updated by changing the version numbers below to
# the latest version available on pub.dev. To see which dependencies have newer
# versions available, run `flutter pub outdated`.
dependencies:
  flutter:
    sdk: flutter


  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.2
  image_picker: ^0.8.6
  google_mlkit_digital_ink_recognition: ^0.6.0
  google_mlkit_image_labeling: ^0.5.0
  path_provider: ^2.0.11

dev_dependencies:
  flutter_test:
    sdk: flutter

  # The "flutter_lints" package below contains a set of recommended lints to
  # encourage good coding practices. The lint set provided by the package is
  # activated in the `analysis_options.yaml` file located at the root of your
  # package. See that file for information about deactivating specific lint
  # rules and activating additional ones.
  flutter_lints: ^2.0.0

# For information on the generic Dart part of this file, see the
# following page: https://dart.dev/tools/pub/pubspec
```

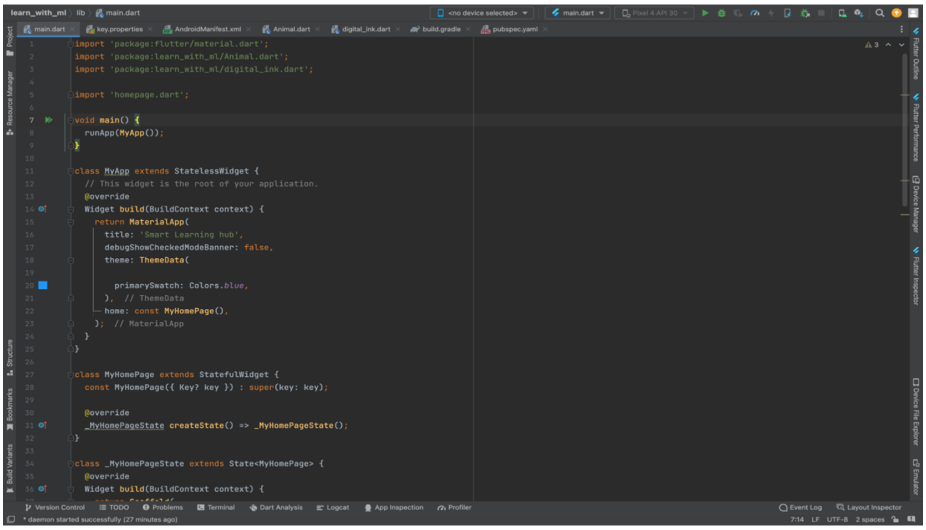
```
* daemon started successfully (16 minutes ago)
```

Fig 3.2.3: Dependency set-up

The setting for our picture categorization application is depicted in the following figure. We are using the two primary plugins for dependencies. MLKIT (mlkit image labeling) is the first, and MLKIT is the second (digital ink recognition).

Main Menu file

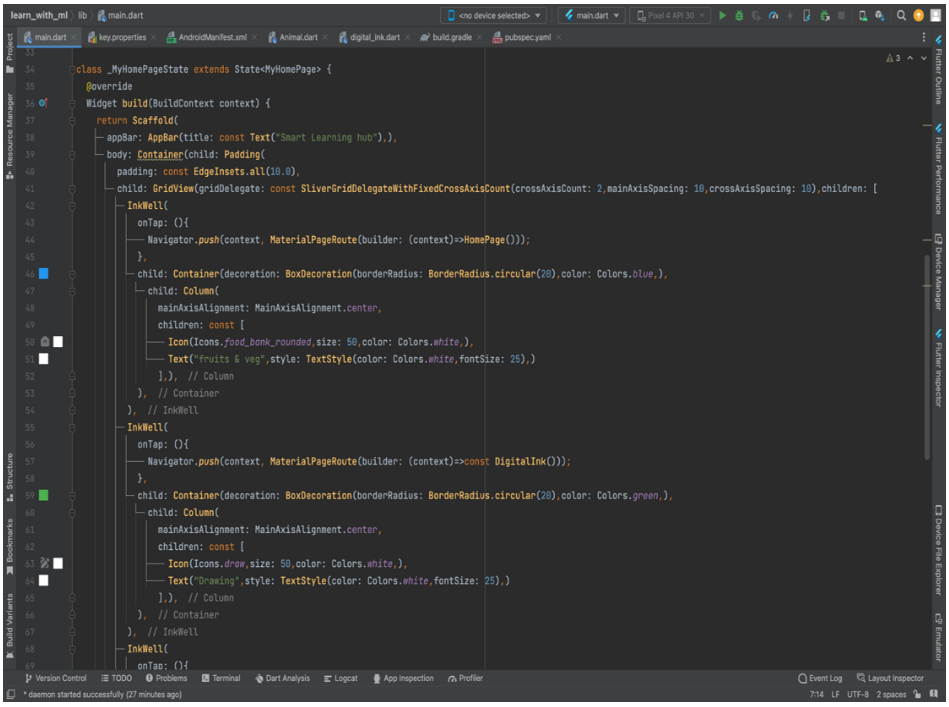
In order to implement the program correctly, we must develop the necessary code. To finish the code, we must declare some method.



```dart
import 'package:flutter/material.dart';
import 'package:learn_with_ml/Animal.dart';
import 'package:learn_with_ml/digital_ink.dart';

import 'homepage.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Smart Learning hub',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(

        primarySwatch: Colors.blue,
      ),  // ThemeData
      home: const MyHomePage(),
    );  // MaterialApp
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({ Key? key }) : super(key: key);

  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
```

```
* daemon started successfully (27 minutes ago)
```

Fig 3.2.4: Code for the main menu screen.



```dart
class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text("Smart Learning hub"),),
      body: Container(child: Padding(
        padding: const EdgeInsets.all(10.0),
        child: GridView(gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(crossAxisCount: 2,mainAxisSpacing: 10,crossAxisSpacing: 10),children: [
          InkWell(
            onTap: (){
              Navigator.push(context, MaterialPageRoute(builder: (context)=>HomePage()));
            },
            child: Container(decoration: BoxDecoration(borderRadius: BorderRadius.circular(20),color: Colors.blue,),
              child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: const [
                  Icon(Icons.food_bank_rounded,size: 50,color: Colors.white,),
                  Text("fruits & veg",style: TextStyle(color: Colors.white,fontSize: 25),)
                ],),  // Column
            ),  // Container
          ),  // InkWell
          InkWell(
            onTap: (){
              Navigator.push(context, MaterialPageRoute(builder: (context)=>const DigitalInk()));
            },
            child: Container(decoration: BoxDecoration(borderRadius: BorderRadius.circular(20),color: Colors.green,),
              child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: const [
                  Icon(Icons.draw,size: 50,color: Colors.white,),
                  Text("Drawing",style: TextStyle(color: Colors.white,fontSize: 25),)
                ],),  // Column
            ),  // Container
          ),  // InkWell
          InkWell(
            onTap: (){
```

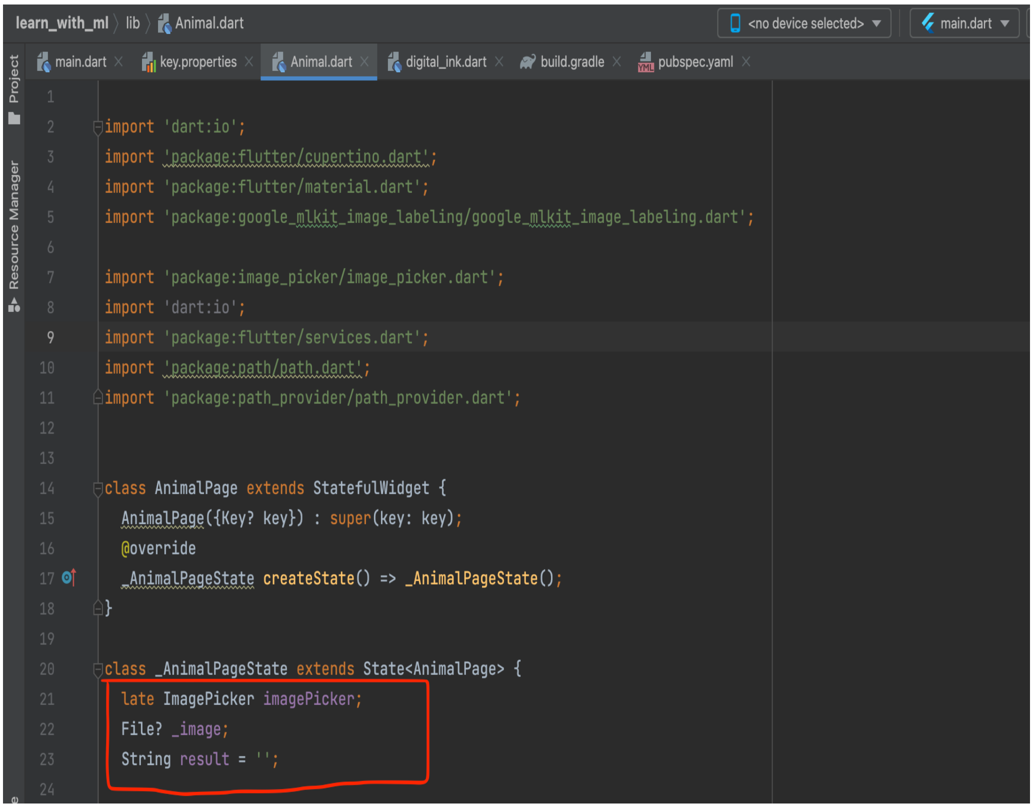
```
* daemon started successfully (27 minutes ago)
```

Fig 3.2.5: Code for main menu screen (continued).

The codes for our grid view buttons are shown in the following figures. From these codes, we press those buttons. We have several codes that we have placed with various buttons. An onTap function containing all codes has been declared. With this feature, switching between scenes is accomplished by pressing various buttons. The back button has a code so that when the user touches it, he will be taken to the home page.

Image picker

Do we need the possibility to take a picture with a tap or phone in order to classify images.



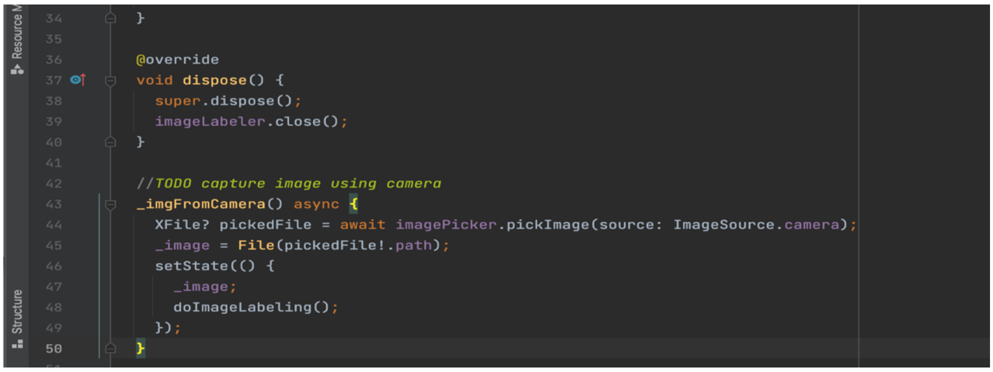
```

import 'dart:io';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:google_mlkit_image_labeling/google_mlkit_image_labeling.dart';

import 'package:image_picker/image_picker.dart';
import 'dart:io';
import 'package:flutter/services.dart';
import 'package:path/path.dart';
import 'package:path_provider/path_provider.dart';


class AnimalPage extends StatefulWidget {
  AnimalPage({Key? key}) : super(key: key);
  @override
  _AnimalPageState createState() => _AnimalPageState();
}

class _AnimalPageState extends State<AnimalPage> {
  late ImagePicker imagePicker;
  File? _image;
  String result = '';

```

Fig 3.2.6: Codes for image picker.

Here, a variable called "Picture Picker" is utilized in the code to enable image picker functionality for the apps.

capture images using a camera

This feature uses the device's live camera to capture photos.



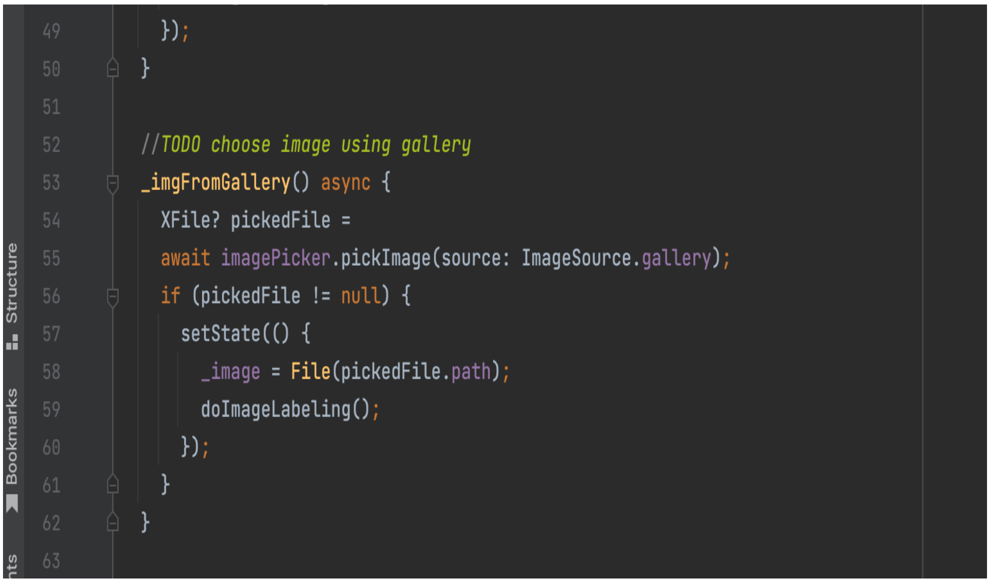
```
}

@override
void dispose() {
  super.dispose();
  imageLabeler.close();
}

//TODO capture image using camera
_imgFromCamera() async {
  XFile? pickedFile = await imagePicker.pickImage(source: ImageSource.camera);
  _image = File(pickedFile!.path);
  setState(() {
    _image;
    doImageLabeling();
  });
}
```

Fig 3.2.7: Codes for capturing images using the camera.

The _imgFromCamera() method was used in this case by the code. This feature will allow users of the app to shoot photos from a live camera.

Choose image using gallery

This feature allows you to capture photographs using the device's live camera.



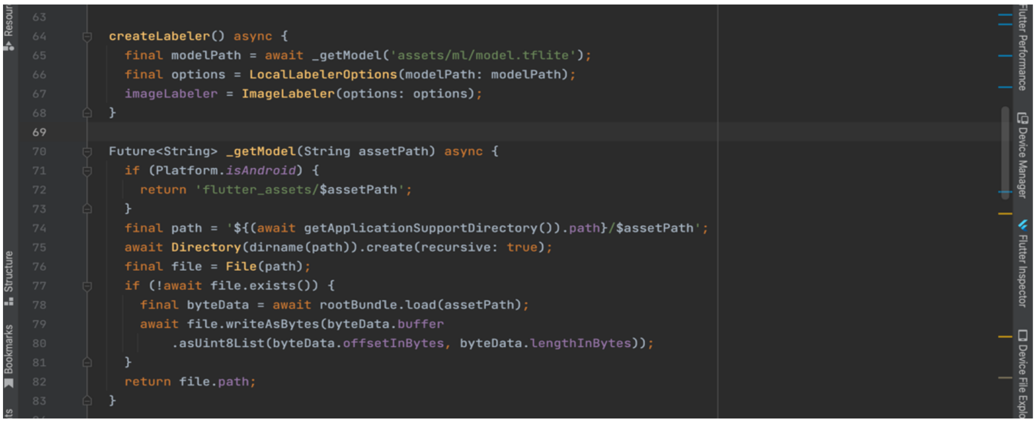
```
  });
}

//TODO choose image using gallery
_imgFromGallery() async {
  XFile? pickedFile =
  await imagePicker.pickImage(source: ImageSource.gallery);
  if (pickedFile != null) {
    setState(() {
      _image = File(pickedFile.path);
      doImageLabeling();
    });
  }
}
```

Fig 3.2.8: Code for choosing images using the gallery

The code for selecting an image from the device gallery is shown in the following figure. Here, the function _imgFromGallery() is invoked, allowing the objects to move between locations.

Get Model Path

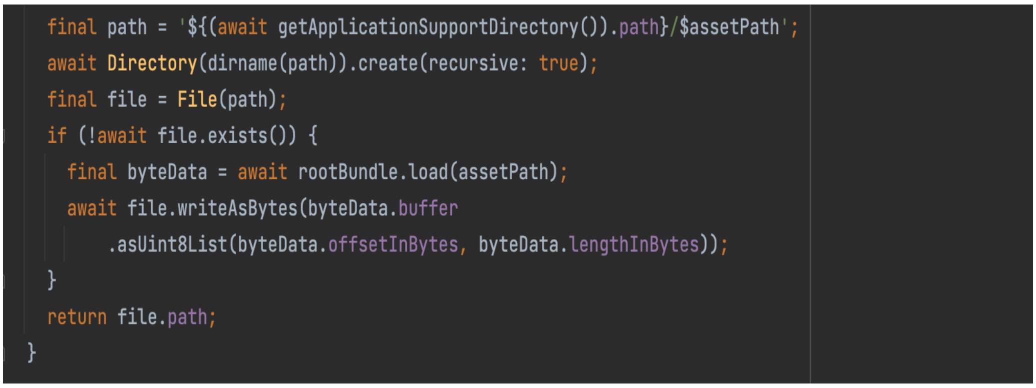
We utilized the TensorFlow lite model to classify images using our model, and we had to create a path to extract the model's data from the dataset.



```

createLabeler() async {
  final modelPath = await _getModel('assets/ml/model.tflite');
  final options = LocalLabelerOptions(modelPath: modelPath);
  imageLabeler = ImageLabeler(options: options);
}

Future<String> _getModel(String assetPath) async {
  if (Platform.isAndroid) {
    return 'flutter_assets/$assetPath';
  }
  final path = '${(await getApplicationSupportDirectory()).path}/$assetPath';
  await Directory(dirname(path)).create(recursive: true);
  final file = File(path);
  if (!await file.exists()) {
    final byteData = await rootBundle.load(assetPath);
    await file.writeAsBytes(byteData.buffer
        .asUint8List(byteData.offsetInBytes, byteData.lengthInBytes));
  }
  return file.path;
}
```

Fig 3.2.9: Getting the model path for classify.

The following code is used to retrieve data from the model. The model path function to retrieve the data has been declared in the code below.



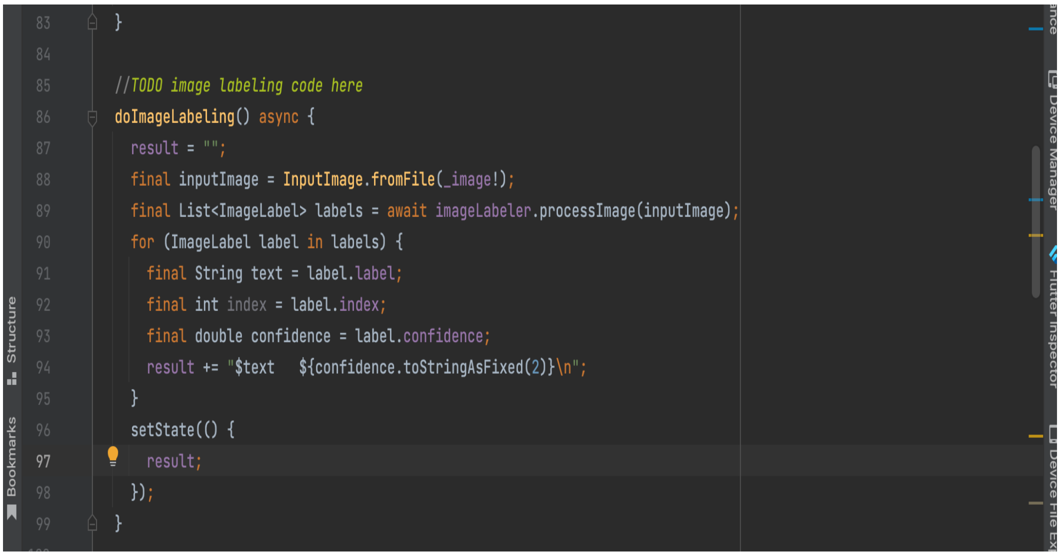
```
  final path = '${(await getApplicationSupportDirectory()).path}/$assetPath';
  await Directory(dirname(path)).create(recursive: true);
  final file = File(path);
  if (!await file.exists()) {
    final byteData = await rootBundle.load(assetPath);
    await file.writeAsBytes(byteData.buffer
        .asUint8List(byteData.offsetInBytes, byteData.lengthInBytes));
  }
  return file.path;
}
```

Fig 3.2.10: Showing assets

We have built a code to display assets in the following figure.

Image labeling

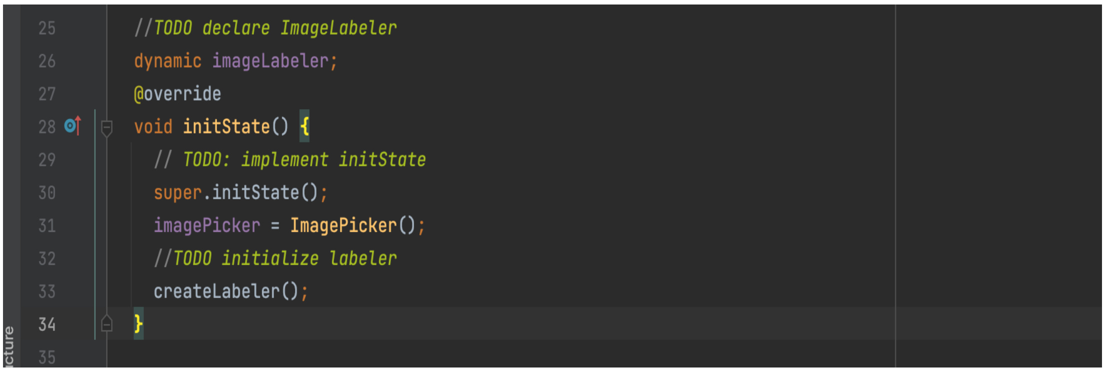
The name of the category for an image can be displayed in a Text Box. The section name in this case does not need to appear immediately beneath the image because we are working with sections.



```
}

//TODO image labeling code here
doImageLabeling() async {
  result = "";
  final inputImage = InputImage.fromFile(_image!);
  final List<ImageLabel> labels = await imageLabeler.processImage(inputImage);
  for (ImageLabel label in labels) {
    final String text = label.label;
    final int index = label.index;
    final double confidence = label.confidence;
    result += "$text   ${confidence.toStringAsFixed(2)}\n";
  }
  setState(() {
    result;
  });
}
```

Fig 3.2.11: Showing image label.

This is an illustration of what occurs when a user chooses a photo and sends it to the area of the app that manages image tagging. This image is being labeled as a component of a bigger set of photos using the DoImageLabeling() method. The category for a photograph may be shown in a Text Box. The section name does not need to be shown immediately underneath the image since we are utilizing sections.



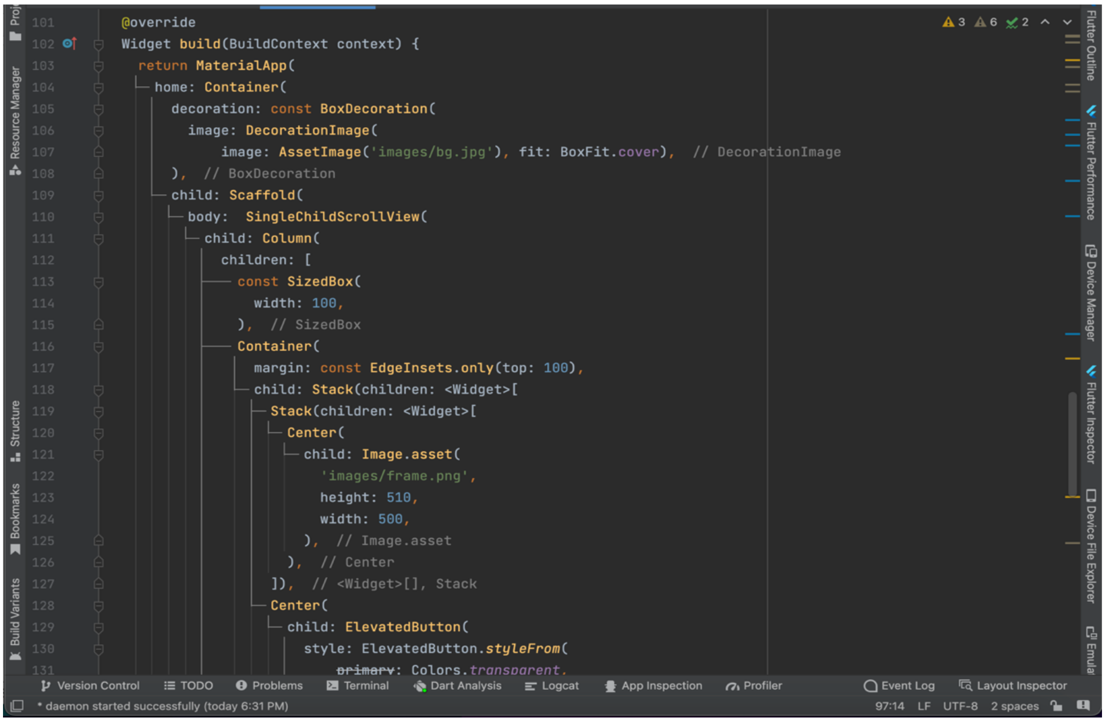
```
//TODO declare ImageLabeler
dynamic imageLabeler;
@override
void initState() {
  // TODO: implement initState
  super.initState();
  imagePicker = ImagePicker();
  //TODO initialize labeler
  createLabeler();
}
```

Fig 3.2.12: Declare Image label

As we can see, the picture labeler function is declared there. The code for the initialized labeler is shown in the accompanying figure.

Getting the UI design assets

In this project, the text box and capture button were initialized on the user interface (UI) page.



```
@override
Widget build(BuildContext context) {
  return MaterialApp(
    home: Container(
      decoration: const BoxDecoration(
        image: DecorationImage(
            image: AssetImage('images/bg.jpg'), fit: BoxFit.cover),  // DecorationImage
      ),  // BoxDecoration
      child: Scaffold(
        body:  SingleChildScrollView(
          child: Column(
            children: [
              const SizedBox(
                width: 100,
              ),  // SizedBox
              Container(
                margin: const EdgeInsets.only(top: 100),
                child: Stack(children: <Widget>[
                  Stack(children: <Widget>[
                    Center(
                      child: Image.asset(
                        'images/frame.png',
                        height: 510,
                        width: 500,
                      ),  // Image.asset
                    ),  // Center
                  ]),  // <Widget>[], Stack
                  Center(
                    child: ElevatedButton(
                      style: ElevatedButton.styleFrom(
                        ~~primary~~: Colors.transparent,
```

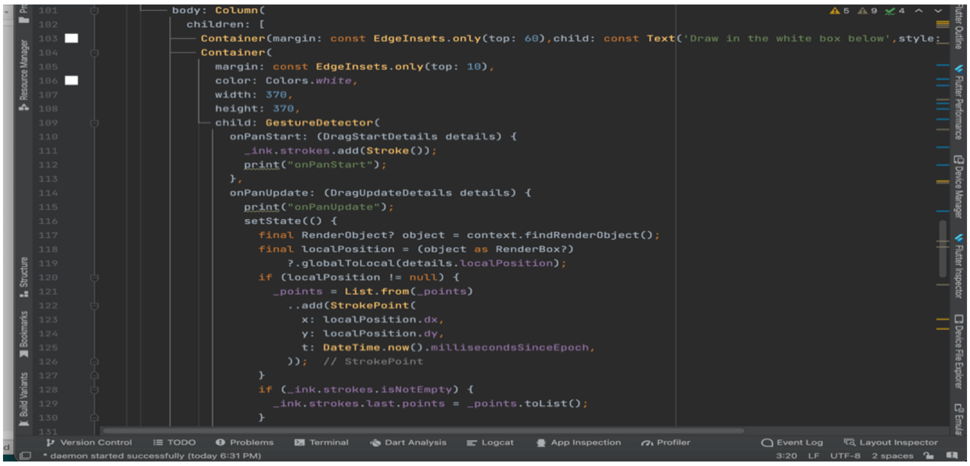
* daemon started successfully (today 6:31 PM)

Fig 3.2.13: Code for get the assets for ui design.

The code for selecting the image's remain frame and displaying the UI backdrop page image assets is shown in the following figure. After taking the photo, adjust the fixed size so that it stays within the frame.

Drawing page UI

There is the UI design code for the Drawing Pad page. Here, we can see that the hint text section and graph size have been set. created the local position Dx and Dx from scratch.Dx.

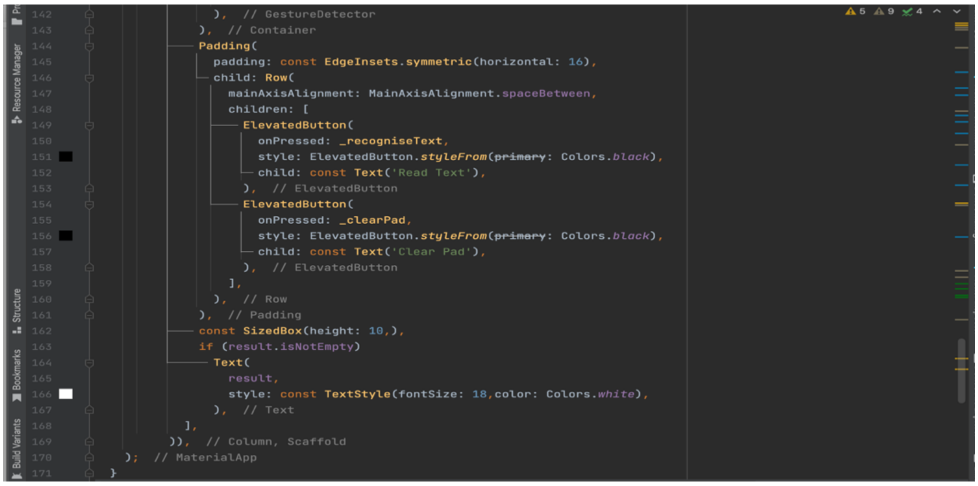


```
body: Column(
  children: [
    Container(margin: const EdgeInsets.only(top: 60),child: const Text('Draw in the white box below',style:
    Container(
      margin: const EdgeInsets.only(top: 10),
      color: Colors.white,
      width: 370,
      height: 370,
      child: GestureDetector(
        onPanStart: (DragStartDetails details) {
          _ink.strokes.add(Stroke());
          print("onPanStart");
        },
        onPanUpdate: (DragUpdateDetails details) {
          print("onPanUpdate");
          setState(() {
            final RenderObject? object = context.findRenderObject();
            final localPosition = (object as RenderBox?)
                ?.globalToLocal(details.localPosition);
            if (localPosition != null) {
              _points = List.from(_points)
                ..add(StrokePoint(
                  x: localPosition.dx,
                  y: localPosition.dy,
                  t: DateTime.now().millisecondsSinceEpoch,
                ));  // StrokePoint
            }
            if (_ink.strokes.isNotEmpty) {
              _ink.strokes.last.points = _points.toList();
            }
```

Fig 3.2.14: Drawing page design.

Read And Clear Button

Here you can see we are declaring the two buttons for the drawing graph page.



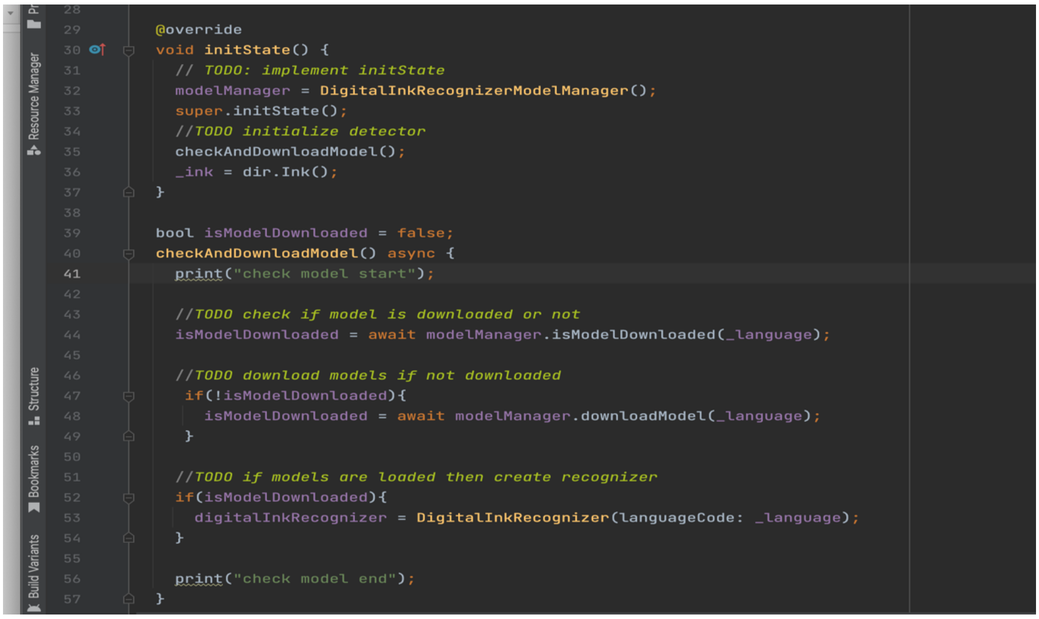
```
              ),  // GestureDetector
            ),  // Container
            Padding(
              padding: const EdgeInsets.symmetric(horizontal: 16),
              child: Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: [
                  ElevatedButton(
                    onPressed: _recogniseText,
                    style: ElevatedButton.styleFrom(~~primary~~: Colors.black),
                    child: const Text('Read Text'),
                  ),  // ElevatedButton
                  ElevatedButton(
                    onPressed: _clearPad,
                    style: ElevatedButton.styleFrom(~~primary~~: Colors.black),
                    child: const Text('Clear Pad'),
                  ),  // ElevatedButton
                ],
              ),  // Row
            ),  // Padding
            const SizedBox(height: 10,),
            if (result.isNotEmpty)
              Text(
                result,
                style: const TextStyle(fontSize: 18,color: Colors.white),
              ),  // Text
          ],
        )),  // Column, Scaffold
  );  // MaterialApp
}
```

Fig 3.2.15: Declare Read and clear button.

Declare the read and clear pad button using the image below. After writing any keyword on the pad, read the button's usage. Lastly, clear the pad graph of all draws.

Drawing Model download loop

We are trying to execute an if statement. The condition first determines if the model has been downloaded or not; if not, it calls the isModelDownloaded method once more after which the model is loaded and a recognizer is created. supplies required for the picture frame.



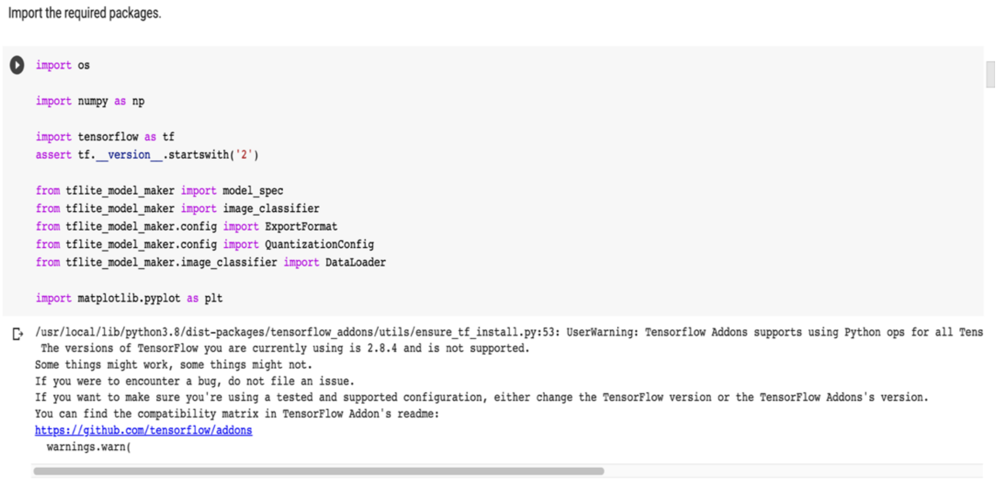
```
@override
void initState() {
  // TODO: implement initState
  modelManager = DigitalInkRecognizerModelManager();
  super.initState();
  //TODO initialize detector
  checkAndDownloadModel();
  _ink = dir.Ink();
}

bool isModelDownloaded = false;
checkAndDownloadModel() async {
  print("check model start");

  //TODO check if model is downloaded or not
  isModelDownloaded = await modelManager.isModelDownloaded(_language);

  //TODO download models if not downloaded
   if(!isModelDownloaded){
     isModelDownloaded = await modelManager.downloadModel(_language);
   }

  //TODO if models are loaded then create recognizer
  if(isModelDownloaded){
    digitalInkRecognizer = DigitalInkRecognizer(languageCode: _language);
  }

  print("check model end");
}
```

Fig 3.2.16: Download the model from online.

The looping model download procedure is shown in the accompanying picture, which also successfully downloads the model.

setup for Google Colab

For image classification, we're utilizing the Google Colab import package.



Import the required packages.

```
import os

import numpy as np

import tensorflow as tf
assert tf.__version__.startswith('2')

from tflite_model_maker import model_spec
from tflite_model_maker import image_classifier
from tflite_model_maker.config import ExportFormat
from tflite_model_maker.config import QuantizationConfig
from tflite_model_maker.image_classifier import DataLoader

import matplotlib.pyplot as plt
```

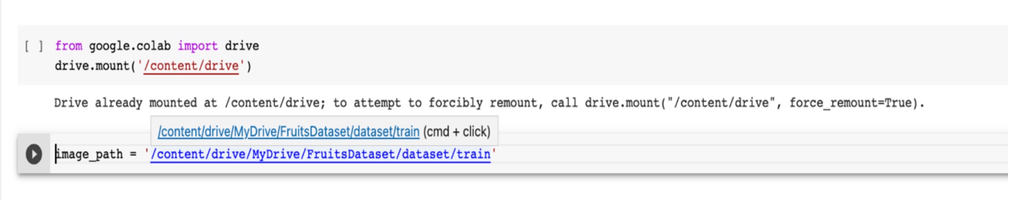
```
/usr/local/lib/python3.8/dist-packages/tensorflow_addons/utils/ensure_tf_install.py:53: UserWarning: Tensorflow Addons supports using Python ops for all Tens
 The versions of TensorFlow you are currently using is 2.8.4 and is not supported.
Some things might work, some things might not.
If you were to encounter a bug, do not file an issue.
If you want to make sure you're using a tested and supported configuration, either change the TensorFlow version or the TensorFlow Addons's version.
You can find the compatibility matrix in TensorFlow Addon's readme:
https://github.com/tensorflow/addons
  warnings.warn(
```

Fig 3.2.17: import required package.

For the as np section of the code, which asks Python to give NumPy the alias of np, the (numpy as np) package is imported using the following pictures. It is a foundation library that can be utilized to directly build Deep Learning models or indirectly utilizing wrapper libraries made on top of TensorFlow to facilitate the process.

Setup drive path

We have set up the dataset path so that we can access the data from the precise spot.

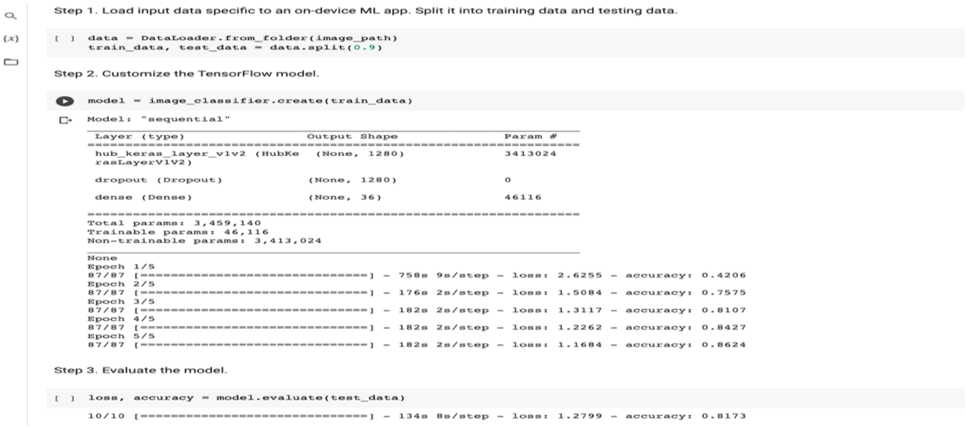


```
[ ] from google.colab import drive
    drive.mount('/content/drive')

    Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

image_path = '/content/drive/MyDrive/FruitsDataset/dataset/train'
```

/content/drive/MyDrive/FruitsDataset/dataset/train (cmd + click)

Fig 3.2.18: setup drive path.

The scene for setting up the path and declaring the methods, which are called image path, is shown in the accompanying figure. These functions are used in the following code to retrieve the data from the database.



Step 1. Load input data specific to an on-device ML app. Split it into training data and testing data.

```
[ ] data = DataLoader.from_folder(image_path)
    train_data, test_data = data.split(0.9)
```

Step 2. Customize the TensorFlow model.

```
model = image_classifier.create(train_data)
```

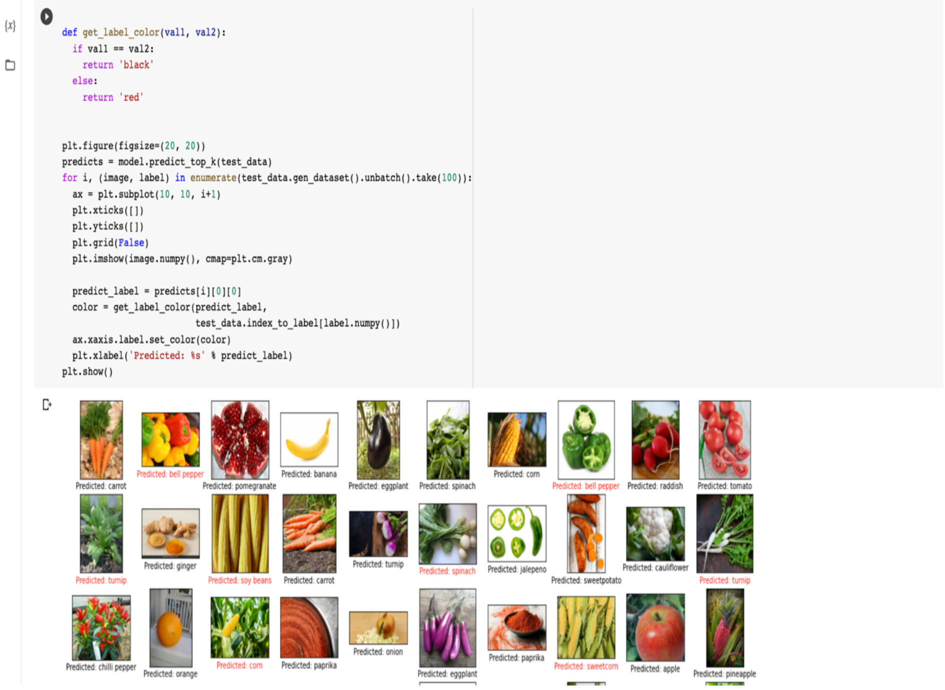
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 hub_keras_layer_v1v2 (HubKe  (None, 1280)             3413024
 rasLayerV1V2)

 dropout (Dropout)           (None, 1280)              0

 dense (Dense)               (None, 36)                46116

=================================================================
Total params: 3,459,140
Trainable params: 46,116
Non-trainable params: 3,413,024
_________________________________________________________________
None
Epoch 1/5
87/87 [==============================] - 758s 9s/step - loss: 2.6255 - accuracy: 0.4206
Epoch 2/5
87/87 [==============================] - 176s 2s/step - loss: 1.5084 - accuracy: 0.7575
Epoch 3/5
87/87 [==============================] - 182s 2s/step - loss: 1.3117 - accuracy: 0.8107
Epoch 4/5
87/87 [==============================] - 182s 2s/step - loss: 1.2262 - accuracy: 0.8427
Epoch 5/5
87/87 [==============================] - 182s 2s/step - loss: 1.1684 - accuracy: 0.8624
```

Step 3. Evaluate the model.

```
[ ] loss, accuracy = model.evaluate(test_data)

    10/10 [==============================] - 134s 8s/step - loss: 1.2799 - accuracy: 0.8173
```

Fig 3.2.19: Load input data and Customize the TensorFlow Model.

The image categorization scene is shown in the next figure. Here are a few of the steps for loading input data into a mobile machine learning app. Separate the information into training and testing data. Make the TensorFlow model your own. Review the model.



```
def get_label_color(val1, val2):
  if val1 == val2:
    return 'black'
  else:
    return 'red'


plt.figure(figsize=(20, 20))
predicts = model.predict_top_k(test_data)
for i, (image, label) in enumerate(test_data.gen_dataset().unbatch().take(100)):
  ax = plt.subplot(10, 10, i+1)
  plt.xticks([])
  plt.yticks([])
  plt.grid(False)
  plt.imshow(image.numpy(), cmap=plt.cm.gray)

  predict_label = predicts[i][0][0]
  color = get_label_color(predict_label,
                          test_data.index_to_label[label.numpy()])
  ax.xaxis.label.set_color(color)
  plt.xlabel('Predicted: %s' % predict_label)
plt.show()
```

Fig 3.2.20: check the loss and accuracy.

A categorization model is evaluated using three main metrics: precision, accuracy, and recall. Accuracy is defined as the proportion of correctly predicted test data. It is easy to calculate by simply dividing the total number of forecasts by the number of accurate predictions.

We face some Challenges:

In this area of our research, we encountered many difficulties. Not only did we run into issues with the study, but also with the coding. We had a lot of issues, including:

- It takes a while to uncover previous research on the subject of our project.
- Since picture categorization is a pretty recent development in technology, there aren't enough tutorials available for us to learn from.
- While writing the code, we were unable to identify any errors. The lack of inline error detection systems means that. After finishing the code entirely, we had to run it.
- Because we work on a large platform, implementation is particularly challenging.
- Not all types of gadgets can use it. A device with greater specifications is required.

4.1 Experimental Results & Analysis

We will go over our project's descriptive analysis in chapter four. We will provide our experimental findings in this chapter, and we will conclude with a summary of the findings. After putting the project's components together, we have a home screen interface with a straightforward GridView Button that can be used to navigate to another page.

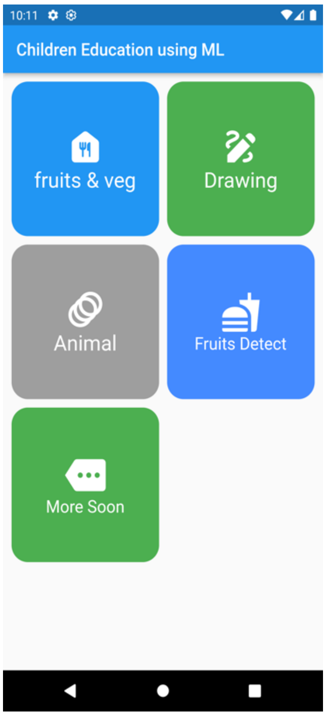


Fig 4.2.1: Interface of the application.

Frame UI Section

This part allows us to take a picture or import one from the gallery after clicking the grid view option.

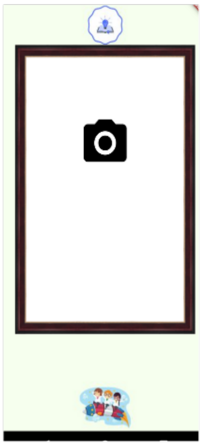


Fig 4.2.2: Location of the trial.

We took a picture of the fruit because we found it but don't know its name. Our software is now running and working to identify the Fruit.

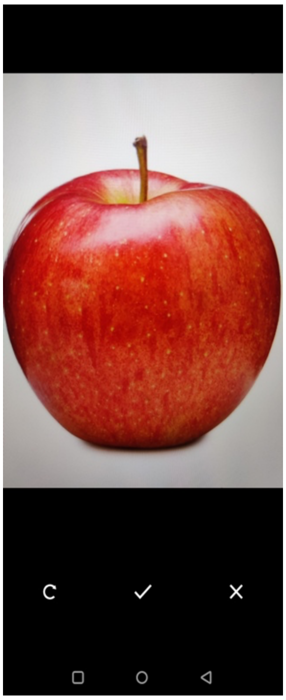


Fig 4.2.3: Camera view of the application.

When an image has been successfully captured, we click the done button to send it to the frame page via the image route.

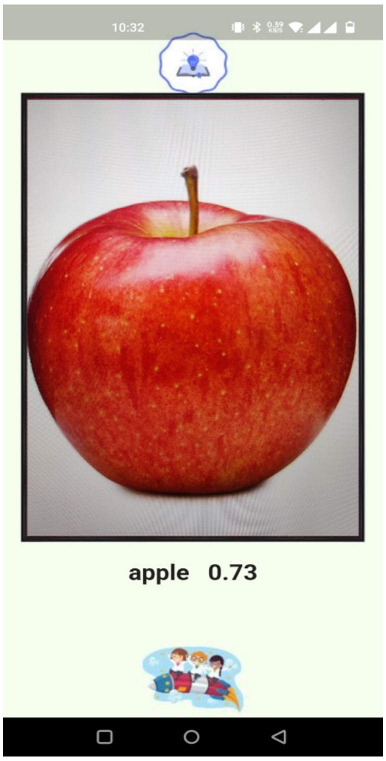


Fig 4.2.4: Showing the Result UI

This will provide us with a clear perspective of our applications because we can see the fruit image as well as the exact fruit name and label.

Import images from gallery

The installation of the gallery image classification was covered in chapter three. The outcome was likewise pleasing.

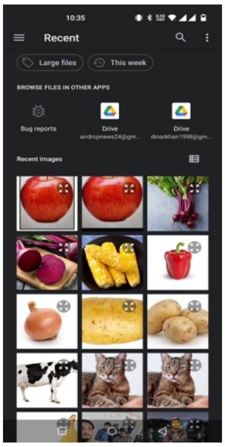
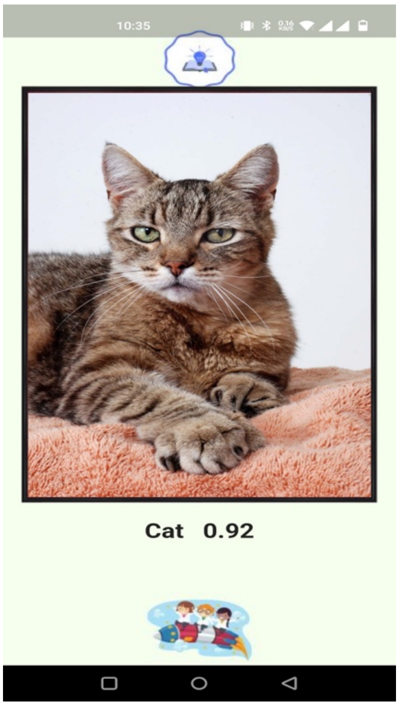


Fig 4.2.5: Select image from gallery.

The image selection process is illustrated in the following figure. To see the outcome, we are announcing the functions for gathering the photograph from the gallery.



Cat 0.92

Fig 4.2.6: Showing the Result UI after import image.

The result displayed after retrieving the image from the gallery is depicted in the following figure. We get more precise results when we choose the gallery image.

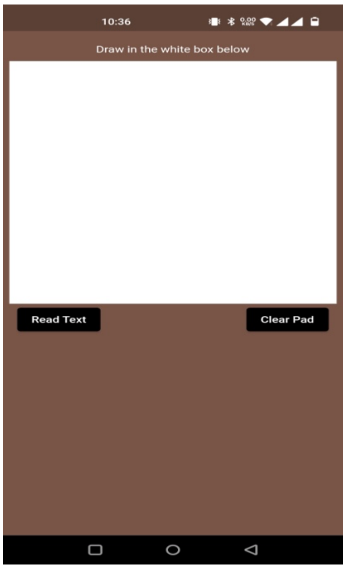


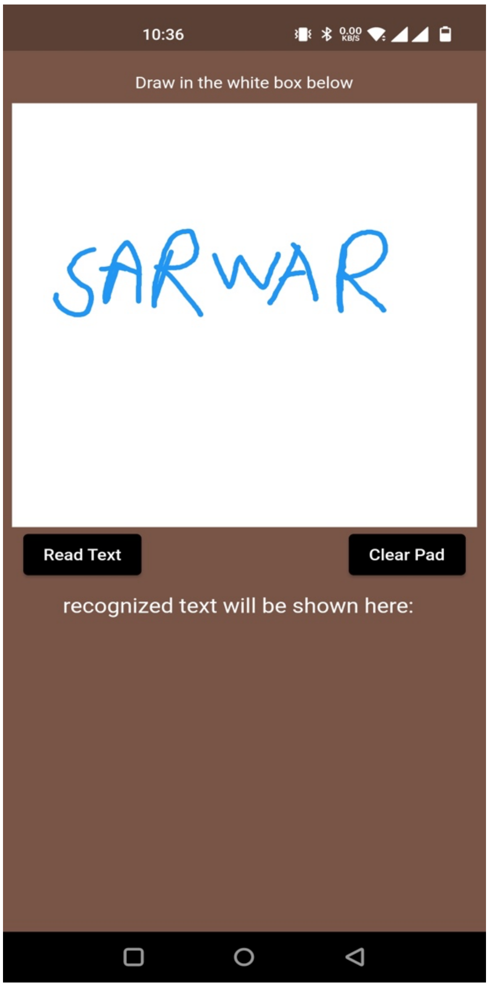
Fig 4.2.7: Drawing pad UI.

A clear view of the drawing pad page is shown in the accompanying illustration. Additionally, wipe the pad after drawing anything on this page.

Drawing pad

To represent a photo frame in 3D, we have a code and certain elements. The user can test out that picture frame.

We demonstrate in this video how a picture frame may be represented in three dimensions using a 3D effect.


10:36

Draw in the white box below

recognized text will be shown here:

Fig 4.2.8: Drawing on Pad Screen.

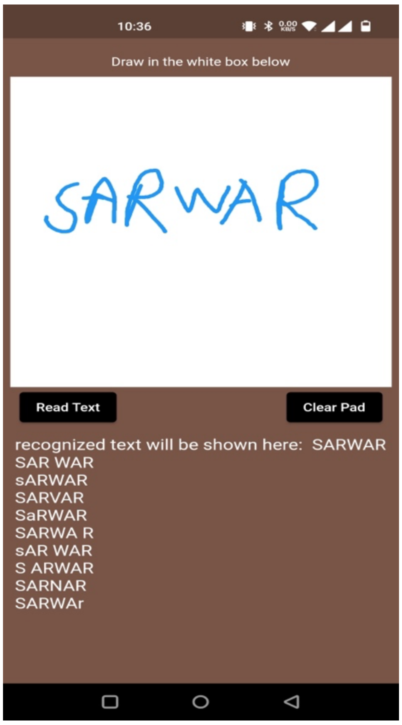
The sketching of the name (SARWAR) on the drawing pad is shown in the following figure, and the outcome is shown in the next step. Any type of text or word in Bangla and English is recognized in the future.

Step 1: Write the name of a new individual or group of individuals on the drawing pad (SARWAR), and the result is shown in the next step.

Step 2: In the future, any form of writing or word will be understood in both Bangla and English.

Drawing pad result

For kids, we have developed a function that can recognize any type of text, which is useful for learning.



```
recognized text will be shown here:  SARWAR
SAR WAR
sARWAR
SARVAR
SaRWAR
SARWA R
sAR WAR
S ARWAR
SARNAR
SARWAr
```

Fig 4.2.9: Drawing recognize result.

Give you eight suggested worlds when the text has been correctly detected and displayed in the text box.

4.3 Discussion

After the implementation went smoothly, we consolidated everything onto one platform and created a user interface (UI) to create an APK. We've developed and gathered several datasets for categorizing images, and we're using those datasets to carry out this research using the TensorFlow lite model. It was challenging to produce the accompanying figures, which display the results of the experiment. However, the final result was partially expected to happen.

CHAPTER 5
IMPACT on SOCIETY, ENVIRONMENT and SUSTAINABILITY

5.1 Impact on Society

This picture classification approaches enhanced and increased each child's understanding of daily living. This system gathers images from cameras that are mounted on belt conveyors. The necessary fruit attributes, such as color and size, are then obtained by image processing. Based on picture pixels, infected fruit is identified. Sorting is done according to size and color. The fundamental objective of object detection is to find instances of each item in digital pictures or real-world situations, segregate them, and assess their essential properties for in-the-moment predictions. A company's entire data architecture includes object detection. in the dining room. Perishable fruits and vegetables should be refrigerated right away. To preserve their freshness and security, keep them in the crisper drawer of the refrigerator. This technology improves safety when driving out of a parking place by detecting moving things nearby and warning the driver, enabling drivers be more aware of their surroundings and feel more at ease.

5.2 Impact on Environment

There are various forms of life in this earth, including 8.7 million living animals, 200 hundred fruits, and 1037 thousand animals. which are common elements of their fruit, vegetable, and animal identification and detection. Additionally, we occasionally forget the names of different kinds of items. Therefore, we can use this approach to remember a variety of items on our app, which is beneficial in various ways in our everyday lives. In terms of quantitative and accuracy analyses, it has been determined that the suggested approach performs better than the already-used picture classification algorithms. With a modest training dataset, we show how to recognize six additional unusual fruits, including rock melons, strawberries, apples, avocados, mangos, and oranges. Our team gathered the datasets for sweet pepper and rock melon, and a Google Image search was used to find the remainder. We select orchard-themed fruit photos. Upon publication, the training and testing dataset will be made accessible.

5.3 Ethical Aspects

It can be challenging to consistently anticipate for users the right fruit, vegetable, and animal. But I have a good chance of succeeding. We also observe that certain nations are developing systems to support them. However, every platform only recognizes things from one category, therefore numerous categories are not supported. Children need to learn a variety of things. Children must be aware of some exam names and which fruit varieties they have never seen previously. Children are highly knowledgeable about common items, but occasionally they forget. For this reason, we are intending to produce a multi-category detection system. Therefore, in our opinion, it must share the experience with other users. We currently have plans to create a platform that will benefit them and advance their learning. It will assist all kids in learning new information.

5.4 Sustainability Plan

Fruits can be recognized by their color, shape, and texture. They highlight how difficult it is to correctly classify fruits from other species that are similar. They recommend combining current methods with the texture, shape, and color of fruits in order to pinpoint areas of interest.

images. To increase recognition accuracy, the texture of the fruits is also used in a technique that combines shape, size, and color with a k-nearest neighbor algorithm. Peppers, which have intricately formed, variegated fruits that complement the plant canopy, are the plants being targeted. Finding and counting green and red pepper fruits on large, dense greenhouse pepper plants is the aim of the application. 28000 pictures that serve as both the training and validation data for this work represent more than 1000 plants and their fruits. The method used to locate and count peppers requires two steps: the first involves identifying the fruits in a single image, and the second involves merging several images to increase the fruit detection rate. The approach for locating the pepper fruits in a single picture includes finding points of interest, applying a complex high-dimensional feature descriptor to a patch surrounding the point of interest, and using a so-called "bag-of-words" to categorize the patch. They draw attention to how challenging it is to categorize fruits from related species. The texture, shape, and color of fruits should be used in conjunction with existing techniques to help identify relevant sections.

Chapter 6

Summary, Conclusion, Recommendation, and implication for Future Research

6.1 Summary of the Study

Image categorization is a relatively recent concept in education. Using IC in the educational setting is mostly done to aid students in developing a solid grasp of cutting-edge technology. Now, in addition to being able to display any fruit, we can also display the result's name and accuracy. In this paper, we provide a fresh classification system and method for locating plant diseases. Six separate classes will be able to be recognized and categorized using the proposed approach. The majority of works in the body of literature that is now available fit into one of three to five types. The bulk of publications employs little information, which can result in overfitting, despite the possible issues with overfitting.

6.2 Conclusion

We occasionally travel outside of my own nation while still viewing fruits, plants, and animals. Our goal is to develop a system that will enable people to recognize fruit and animals. So, if kids are concerned or uncertain about any sort of thing, this technique can be useful. Our system storage will house all of the data photos. After taking the image and displaying the accurate result with an accuracy number, we are able to represent a result in our system. Children and other users will benefit from a better user experience as a result. The accuracy of the suggested approach, which is based on Python, is around 90% higher. The developed technology is able to recognize plant ailments and provide a cure for them. By properly recognizing the disease and its treatment, it is possible to enhance the health of the plant. Recent advances in ML provide solutions with incredibly accurate results.

6.3 Implication for Further Study

Any e-learning institution is welcome to collaborate on the implementation and advancement of this study. The fruit, vegetable, and animal identification and segmentation model may eventually be further integrated with the edge intelligent platform with powerful computing capabilities. Additionally, if we adopt this technique in fruit orchards as well as in every zoo to identify any kind of animal, people would benefit much from it. Processing speed and precision may be increased by using Google's GPU. Drones may be equipped with the technology to provide aerial surveillance of detecting fields. Future investigations should focus on locating all fruit and vegetable stages and places on the plant. The developed model could be a part of a system that aids in decision-making and, as a result, presents a conducive environment for selecting the optimal course of action. give an economical way to recognize things labeled in a smartphone application. As was indicated in the preceding sections, The backdrop is simple and clear, despite the fact that this collection contains several photos of various fruits, vegetables, and animals. But the actual surroundings must be taken into account for a scenario to be authentic. In many different scientific fields, a novel technology known as multispectral and multispectral imaging has been used. The best way to recognize any picture is thus to combine it with efficient ML architectures. A detailed investigation is required to fully understand the factors influencing picture label recognition, including the classes and number of datasets, learning rate, illumination, and similar elements. The created model could be a component of a system that facilitates decision-making and, as a consequence, provides a favorable setting for deciding on the best course of action. provide a cost-effective solution to identify items designated in a smartphone application. Locating all fruit and vegetable phases and locations on the plant should be the main goal of further research.

REFERENCES

[1] Anderson Rocha, Daniel C. Hauagge, Jacques Wainer, Siome Goldenstein, “Automatic fruit and vegetable classification from images”, Computers and Electronics in Agriculture, Vol. 70, pp. 96–104, 2010.

[2] Arivazhagan S., Newlin Shebiah R., Nidhyanandhan Selva S., Ganesan L., “Fruit Recognition using Color and Texture Features”, Journal of Emerging Trends in Computing and Information Sciences, vol. 1, no. 2, Pages: 90-94, Oct 2010.

[3] Ismail Wan Ishak Wan, Razali Mohd. Hudzari, “Outdoor color recognition system for oil palm fresh fruit bunches (FFB)”, International Journal of Machine Intelligence, Vol-2, Issue 1, Pages:01-10, 2010.

[4] Woo Chaw Seng and Seyed Hadi Mirisaee, “A New Method for Fruits Recognition System”, MNCC Transactions on ICT, Vol. 1, No. 1, June 2009.

[5] Lopez Jose J., Cobos Maximo, and Aguilera Emanuel, “Computer-based detection and classification of flaws in citrus fruits”, Internation conference on natural image processing (ICONIP-2009).

[6] Fernández, L., Castillero, C. and Aguilera, J. M., “An application of image analysis to dehydration of apple discs” Journal of Food Engineering, vol.67, pp.185-193, 2005.

[7] Zhao, J.T., J. Katupitiya, J., “On-tree fruit recognition using texture properties and color data”, International Conference on Intelligent Robots and Systems, pp. 263- 268, 2005.

[8] Bulanon D.M., Burks T.F., Alchanatis V., “Image Fusion of visible and thermal images for fruit detection”, Biosystems Engineering, Vol-103, Issue-1, May 2009, pages:12-22.

[9] Bulanon D.M., Burks T.F., and Alchanatis V., “Study of temporal variation in citrus canopy using thermal imaging for citrus fruit detection”, Biosystems Engineering, Vol-101, Issue 2, Pages 161-171, October 2008.

[10] Qiaorong Zhang, Haibo Liu, Jing Shen, Guochang Gu, Huimin Xiao, “An Improved Computational Approach for Salient Region Detection.” , Journal of Computers, vol.5,no-7, pp.1011-1018, July 2010.

[11] Itti L, Kouch C, “Feature combination strategies for saliency-based visual attention systems,” Journal of Electronic Imaging, vol. 10, pp.161-169, January 2003.

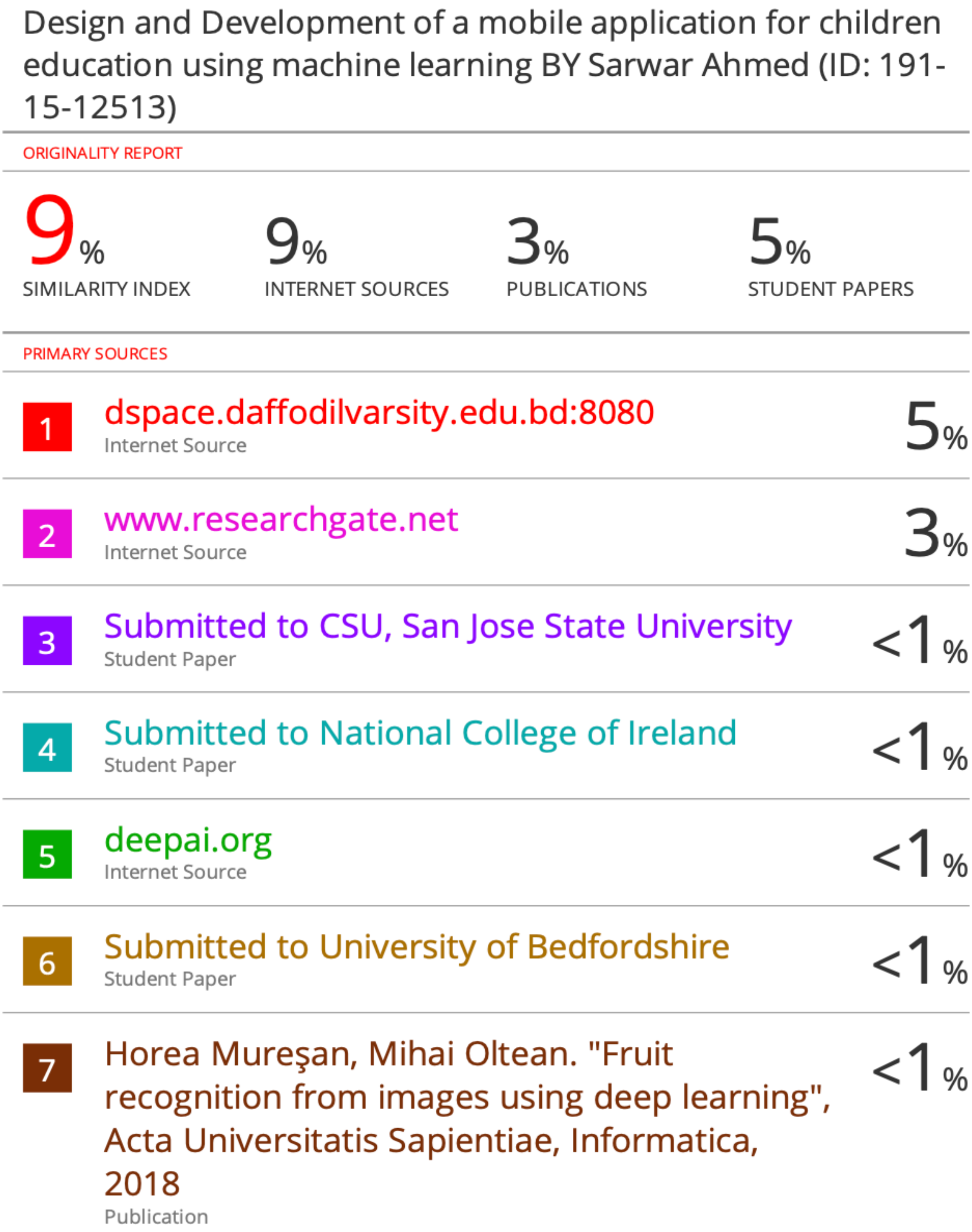
[12] Zheshen Wang, Baoxin Li, “A Two-Stage Approach to Saliency Detection in Images,” IEEE ICASSP 2008,
April 2008, pp. 965-96.

[13] J. Yang, J. Zhao, L. Lu, T. Pan, and S. Jubair, “A new improved learning algorithm for convolutional neural networks,” Processes, vol. 8, no. 3, p. 295, 2020.

[14] Q. Yin, R. Zhang, and X. Shao, “Cnn and rnn mixed model for image classification,” in MATEC Web of Conferences, vol. 277, EDP Sciences, 2019.

[15] J. Zhang, Q. Wu, C. Shen, J. Zhang, and J. Lu, “Multilabel image classification with regional latent semantic dependencies,” IEEE Transactions on Multimedia, vol. 20, no. 10, pp. 2801–2813, 2018.

PLAGIARISM REPORT



Design and Development of a mobile application for children education using machine learning BY Sarwar Ahmed (ID: 191-15-12513)

ORIGINALITY REPORT

9%	9%	3%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	5%
2	www.researchgate.net Internet Source	3%
3	Submitted to CSU, San Jose State University Student Paper	<1 %
4	Submitted to National College of Ireland Student Paper	<1 %
5	deepai.org Internet Source	<1 %
6	Submitted to University of Bedfordshire Student Paper	<1 %
7	Horea Mureşan, Mihai Oltean. "Fruit recognition from images using deep learning", Acta Universitatis Sapientiae, Informatica, 2018 Publication	<1 %