Component Reel Estimator

Made by Ryan Lim & Ryan Ong

Introduction

Purpose

The main purpose of the application is to automate the component reel counting process which is very labor intensive & time consuming.

There are a few cases to be considered:

- Case 1: half used reel with unknown initial quantity
- Case 2: half used reel with known initial quantity
- Case 3: unused reel with unknown initial quantity

Another feature of the application is to read labels of plastic carrier tape component reel using Optical Character Recognition (OCR) technology.

Project Scope

The application is made in flutter (dart) and consists of three modules:

1) Component Reel Estimator Module
2) OCR Label Reader Module
3) Database and tools

Requirements

Development Environment

Flutter - https://docs.flutter.dev/get-started/install
VSCode - https://code.visualstudio.com/download
Python - https://www.python.org/downloads/

Flutter Libraries (pubspec.yaml)

```
cupertino_icons: ^1.0.2
google_mlkit_text_recognition: ^0.6.0
image_picker: ^0.8.7+4
google_ml_kit: ^0.14.0
image: ^4.0.17
sqflite: ^2.2.8+2
path_provider: ^2.0.15
intl: ^0.18.1
html: ^0.15.3
image_cropper: ^4.0.1
image_crop: ^0.4.1
camera: ^0.10.5+2
gallery_saver: ^2.3.2
matrix_gesture_detector: ^0.2.0-nullsafety.1
fuzzywuzzy: 0.1.6
url_launcher: ^6.1.11
excel: ^2.1.0
permission_handler: ^10.4.3
share: ^2.0.4
flutter_image_compress: ^2.0.4
```

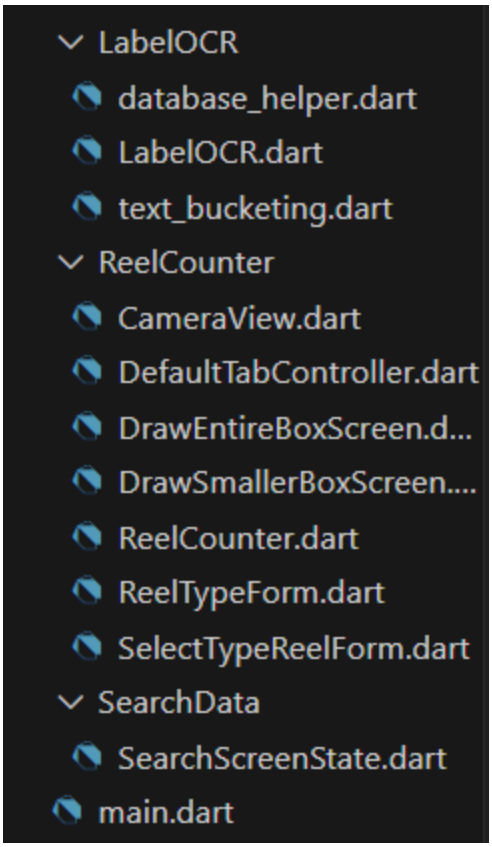
Running / Testing the Application

Users can choose to test the application via emulator or an external device via the command “flutter run”

To install libraries (after adding dependency in pubspec.yaml), use the command “flutter pub get”

To install libraries (from scratch), use the command “flutter pub add flutter_image_compress”, where flutter_image_compress is the name of the library. Library depo can be found on https://pub.dev/

Classes & Explanation



The code is located in the component_reel_estimator/lib path.

The code is split into modules where each file consists of all the code for that particular module.

Label OCR:
Consists of the database and code for extracting text from images, and sorting the text into specific categories.

Database_helper -> init database + helper functions

LabelOCR -> main image pre-processing + code for extracting text from image

Text_bucketing -> code for converting raw OCR output (string) into formatted string (categorized buckets) via fuzzy string matching algorithm

ReelCounter:
Consists of the code for the reel estimator. There are three methods defined in the application:

1. Math (uses raw measurements via https://www.compuphase.com/electronics/reelestimate.htm) to accurately measure the number of components on ANY given reel
2. Camera (uses a simple method of proportion). The user takes a picture of a component reel and manually draws boxes around them. The first box (outer box) signifies the 100% quantity while the second box (inner box) signifies the percentage of remaining reel. This method is used when the initial 100% quantity of the reel is known, and uses proportion to calculate the percentage remaining in the reel and thus the number of components remaining in the reel
3. Same as 1, but uses values from the EIA-481 standard for component reel plastic carrier tape packaging to simplify the process. It is not as accurate due to different manufacturer specifications.

CameraView -> camera function for taking a picture in flutter
DefaultTabController -> controls the tabbing feature in this module
DrawEntireBoxScreen -> draws the outer box and passes the values to the next function
DrawSmallerBoxScreen -> draws the inner box and passes data to reeltypeform
ReelCounter -> aforementioned method 1
ReelTypeForm -> does the math for method 2 using data from draw screens
SelectReelTypeForm -> method 3

SearchData
This class contains the helper functions for the database:

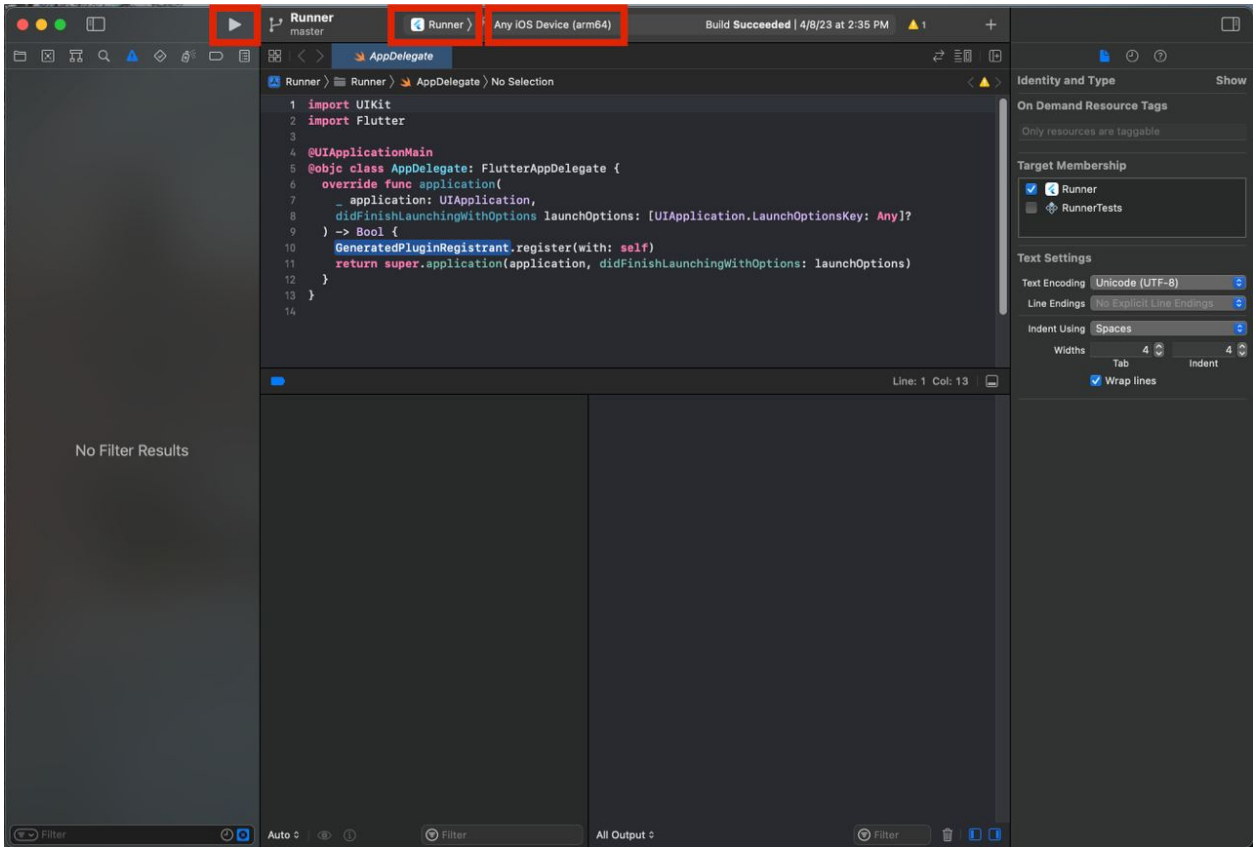
1. Search filter feature
2. Pagination
3. Copy selected items
4. Join Telegram
5. Search product via product/customer ID on web
6. Export DB to excel
7. Share items to telegram

Maintaining the application

Apple

Apple has a policy whereby apps need to be rebuilt on apple devices every 7 days (unless u pay $100 to be an apple developer)

1. Run flutter pub get
2. Cd into ios
3. Run pod install
4. Open Xcode
5. Open file in ios directory that with .worskpace
6. Click on Runner in the center top bar (The one with the flutter logo)
7. Click on edit scheme
8. Click on Run
9. Change build configuration to Release
10. Close the window
11. Ensure that the device stated beside the Runner in step 6 is the correct device that you want to deploy to
12. Press play button on the top left of Xcode



Additional Features

Text Bucketing

Currently, the targets that are being looked for in the code is defined manually under buckets (Vendor Product Number/ Customer Product Number/ Quantity/ Description).

Each company has their own version of a labeling system, I tried to add in as many variations of each label as I could think of but there could be more. The maintainer of the application can add more target texts and more buckets as necessary by editing the Map/List.

OpenCV / ML

The initial idea of the application was to use OpenCV to detect the edges of the reel via Hough Circle Transform, but poor picture quality.

A possibility would be to use some sort of transfer learning or machine learning model to fix it but lack of data (reel images with boxes already drawn) makes it challenging.