California State University

of Fullerton

Artificial Intelligence

CPSC 481

Project: AI is your eyes

Professor

Wenlin Han

Student

Huy Ho

Github project link

Table of Contents

1. Introduction

1.1. Purposes and ideas

I am currently developing some mobile application project, however, I have never tried to imply any AI into my mobile project; therefore, I think it is a good time to take a look at it while I am taking AI course.

After doing small research over the weekend, I have decided to develop a mobile application for the Blind and Visually Impaired people. The reason I chose this idea because nowadays smartphones have become so powerful and convenient. Creating a useful tool for someone could be more meaningful to the community and society.

There are actually some similar mobile apps out there, such as TapTapSee, Be My Eyes. However, when I tested TapTapSee on my android device, it does not work somehow. Similarly, Be My Eyes also have disadvantages because it needs a volunteer who helps a blind person through a video call and internet connection. What if there is an important time but no one volunteers, what if there is no internet at the user's location. Therefore, AI will be the most helpful in these situations.

1.2. Application Scope

The user will take a picture and AI will say what is inside the picture, how many people in it. The application will be simple as much as possible for visually impaired users.

Future possible functions for the project:

- Live camera AI
- AI reading

2. Requirements

2.1. User Interface

The application will be available both android and ios; however, I will focus on the Android version first, the ios version may be possible but it is not ensured.

The application will require users to give access to the gallery, camera, and speaker for fully running.

2.2. Integrated Development Environment

The application will be developed with the Flutter framework, and the main programming language will be Dart. I will use the Visual Studio Code for developing this application.

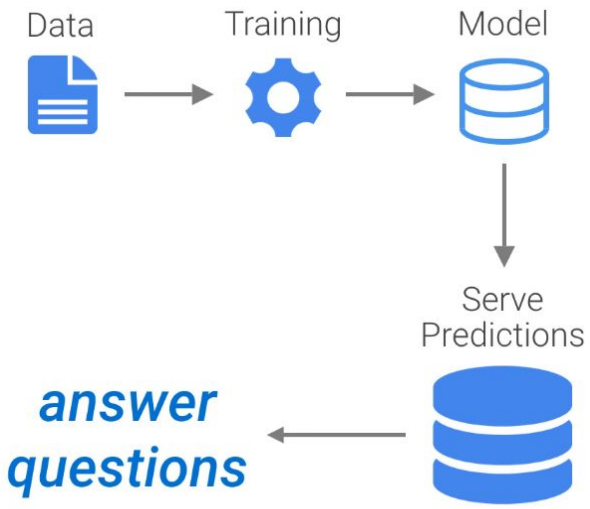
To run this application or developing, it requires to have Android SDK, Flutter SDK, and an android emulator or ADB USB connect the android device in debugging mode.

To make better and faster AI, I will use the pre-trained ML Vision API for Flutter from Google. However, I will make deep research writing in this report about how to train and build a Machine Learning Vision.

3. Machine Learning Vision

3.1. Collecting Data

To build a custom Machine Learning Vision (ML Vision) with , we need to follow several steps: Collecting Data -> Structuring Data -> Training Model, after these steps, we can either evaluate our trained model or use it in AI project.



To make it easier to capture data for ML Vision, we can collect training data by capturing videos of the objects instead of taking hundreds of pictures, and then use ffmpeg to extract the frames.

Here is an example of recording a chair.

Once the frames are extracted, you'll have one folder per label, filled with images of that label. This is simpler than having one massive folder of all the images.

```
ffmpeg -i chair.mp4 chair%03d.jpg
```

(The %03d in the filename will give us 3-digit padded numbering, such as chair003.jpg, chair073.jpg, etc. If you have more than 999 images, you should use %04d or some other value as appropriate)

Next we can upload the images to Google Cloud Storage using gsutil, replicating the folder structure of one folder per label:

```
gsutil -m cp -r all_data gs://cloudml-demo-vcm/dataset
```

(This command recursively copies/uploads the entire folder structure in all_data and does so with multiple streams using -m)

3.2. Structuring Data

To train an AI model, we need a CSV file that has 1 row per image, and two columns, the first for the location of the image in Google Cloud Storage, and the second for the label, like ‘chair’, or ‘table’.

Quick way to accomplish this, we can use Jupyter notebook with Pandas dataframe to export a CSV file.

Here code sample from Yufeng G (developer at Google Cloud), he create custom dataset for AutoML Vision.

3.3. Training Model

Once your dataset has been created and processed, click the **Train** tab to initiate model training.

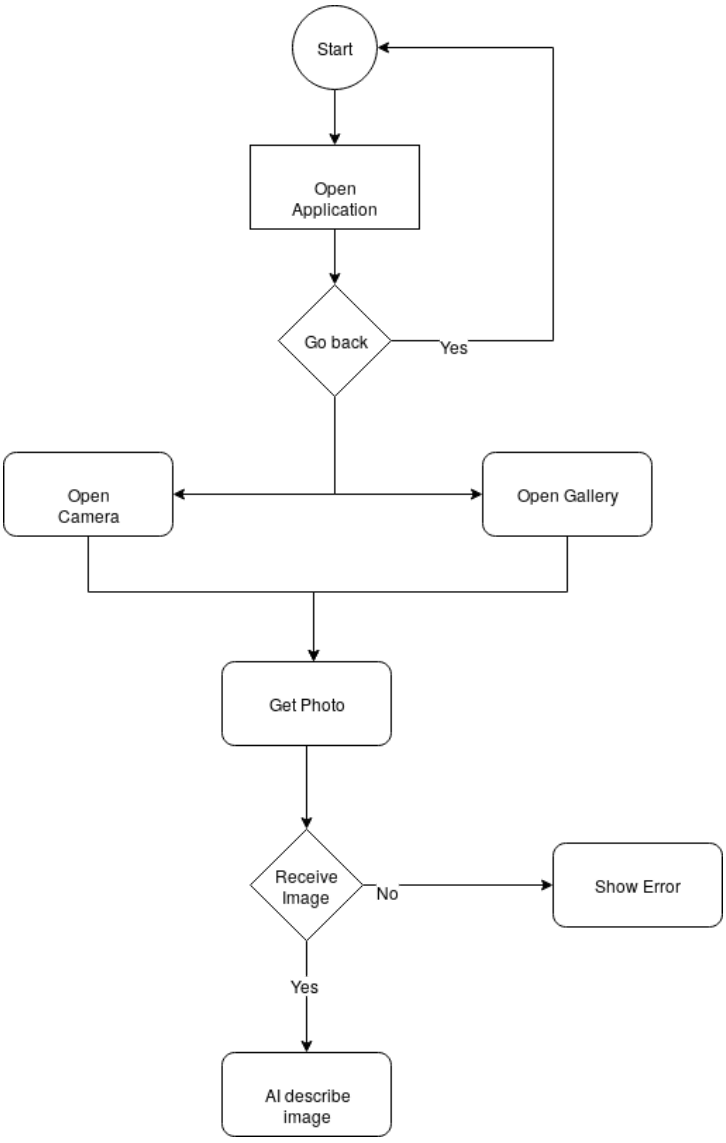
Click **Train New Model** to continue.

Once training is complete, your model is automatically deployed, and it is ready to be used.

4. Design and Implementation Concepts

4.1. Design

Activity Diagram:

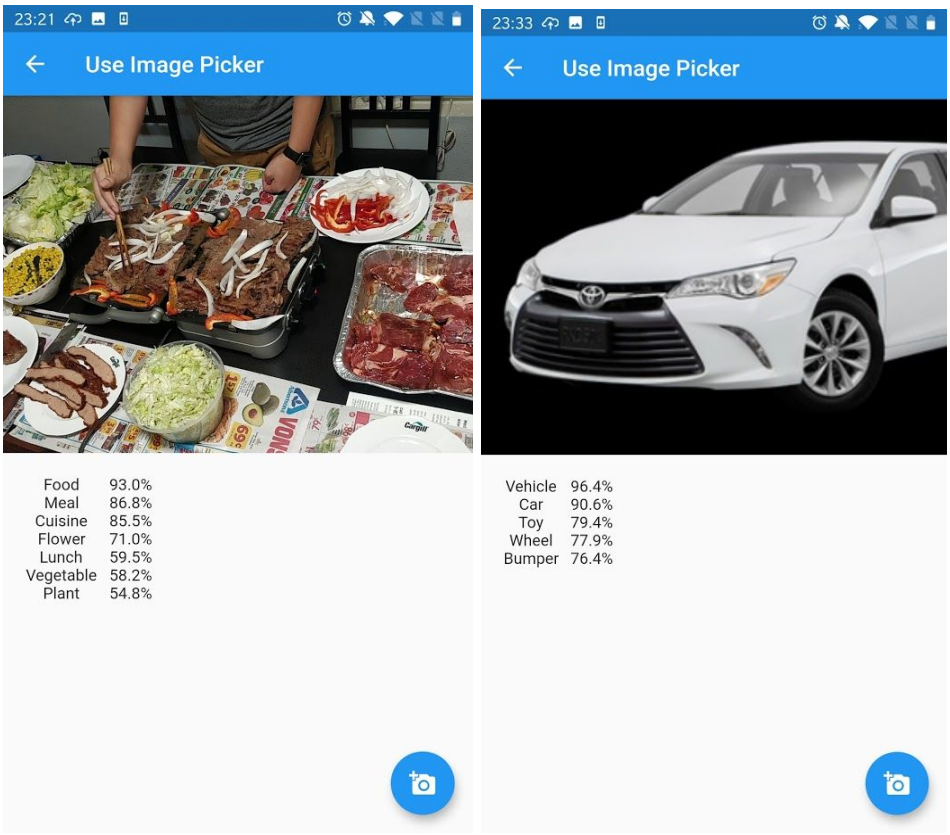


4.2. Implementation

In Flutter, there is a package name firebase_ml_vision which is a pre-trained ML Kit Vision for Firebase API. Even though it is still under development, it is recommended by Flutter team development. Furthermore, I also use this text to speech flutter package to have AI speak out loud. All the user interfaces will use material flutter with work for both android and ios.

5. Progress

- Currently project have finish function take picture and detect object label.
- Next plan will apply face detector and text to speech capacity for AI.



*Note: the percentage is the correctness of prediction.

Github project link