Joel Georges

Bo Sheng

Mobil Applications 443

Dec 20, 2019

FlowerFinder App

Note: The original name for the app was the WhatFlower app, but I wanted to change the name to FlowerFinder, but Android provided many difficulties in changing the package and project name and also had issues with syncing it Firebase console.

The purpose of my project was to develop a user-friendly app that would allow consumers to classify different plant species unknown to the user. In addition, the app would also teach the user about the different plants by providing background information on each plant image the user captured while using the FlowerFinder app. I also wanted to develop an Android app that would help me utilize some of the technology and skills learned throughout the semester. Prior to the course, JSON parsing and HTTP networking in Android were concepts that I struggled with along with ASYNC task and background threading. This project helped me master those important and industry needed skills.

To use the app can be done in three simple steps: first open the app and press the classify button to take a picture of the unknown plant you want to classify, second take the picture of the species, third click done button on external camera and the results appear on the screen. The app is currently only able to classify five different plant species which are tulips, roses, dandelions, daisies, and sunflowers. Once the app classifies the one of the five plants, it utilizes the Wikipedia API and provides the user with biography information on the plant from the Wikipedia website. The app also utilizes the Wikipedia API to obtain an image of the plant to provide to the user.

The people that will benefit most from this app are gardeners, novice planter, people who love to travel or outdoors, or people who are just curious about their environment and have a love for learning different knowledge. The similar apps in the market are: PlantSnapp, LikeThat Garden, FlowerChecker, Plantifier, and NatureGate. To use the FlowerFinder app, that user must have Wi-Fi or LTE internet in order to get the Wikipedia information and Android device with API level of 21 or higher.

The FlowerFinder app features one layout that consists of one image view, text view, and button. The image view is used to hold the image of plant obtained from the Wikipedia API. The text view is used to provide either Wikipedia bio about the planet or to notify the user that the plant image captured could not be classified. The button is connected with a click listener and is used to send an intent to the user’s camera application on their phone. Below are images of the different app states.

First image is initial screen when opening the app. Second image is the state of the app when an image cannot be classified. Third image is the image of a user’s surrounding inside their camera application. Fourth is image of a successful classification of a plant.

 State of app when plant cannot be classified. 
 A picture containing indoor, ceiling, wall, floor

Description automatically generated A screenshot of a cell phone

Description automatically generated

The images below are the metrics used to create the app.

A screenshot of a cell phone

Description automatically generated

A screenshot of a social media post

Description automatically generated

The app currently only targets smartphones.

To implement the app, first I gathered the flower image dataset from the Kaggle website. The author of the dataset is [Alexander Mamaev](https://www.kaggle.com/alxmamaev). The Firebase free account only allows thousand images to be trained and tested per month. Then I took one hundred and ninety-nine images of each flower: tulips, sunflowers, roses, daisies, and dandelions. I initially meant to do two hundred of each flower, but I accidently did one less. Then I sent the data to the Firebase console where 80% of the data was used for training and 20% was used for testing. The Firebase console created the flower model. There was only was one class made for the app which is called MainActivity. Inside the MainActivity class, I utilize the FireBaseAutoML, FirebaseVisionImageLabeler & FirebaseVisionImage API to create and convert the user’s image to a FirebaseVision image. I also used the same API’s to load the model locally and create a labeler to label the images. To load an image of the flower captured from the camera, I used the Wikipedia API and two ASYNC task to populate the biography data from flower and obtain the image in the background. This allows the user to interact with the app while the processes running without crashing. I also used JSON and HTTP API to get the web data from Wikipedia and parse the data. In addition, I set up the options for the classifier so that it only returns labels with at least a 70% confidence.

To test the app, I bought an android phone and took live pictures of flowers in grocery store. I also took pictures of pictures on google images to test the app. Then I was able to find some of the flowers in nature and took pictures of them.

There many errors and challenges faced while creating the app. Learning how to use the Firebase framework and finding the dataset was a challenge. I had errors trying to get the Firebase classifier to work. This issue was solved by converting the global Firebase object variables into local variables. The app would crash when the classifier would return an empty list of classifications because the first prediction was used to generate the URL for Wikipedia.

This was easily solved by using a simple if/else clause to make sure the async tasks would not run with an empty list and invalid URL. Parsing the Wikipedia API and trying to get the page ID number to obtain the Wikipedia bio was a challenge. To solve this, I did a lot of searching on stackoverflow and Wikipedia API docs. In addition, I struggle with converting a caffemodel into a tensor flow lite model and was not able to complete this in time to turn in the project.

In continuation, there are many updates and additions I would have liked to add to the app. First, I would utilize more UI libraries to have a better interface and user experience. Instead of having a button, I would update the app so that it would detect and classify multiple flowers in real time with a confidence score label on the side. Then the user would be able to tap on the label to obtain more information on the flower. I could have also integrated google authentication and backend database to keep track of users as well as allow users to save the different photos they capture along with the location. I could have also added VR, to allow the users to watch a video about the plant they are viewing. I would have also liked the model to be able to classify more different types of flowers instead of just five. This project gave me a great understanding of Android programming and enough skills to obtain a junior or entry level position.

References:

<https://www.kaggle.com/alxmamaev/flowers-recognition>

<https://www.youtube.com/watch?v=J18dQAmrO6M> (Demonstration video)